Achievable CCA2 Relaxation for Homomorphic Encryption*

Adi Akavia[1][0000−0003−0853−3576], Craig Gentry[2], Shai Halevi[3][0000−0003−3432−7899], and Margarita Vald[4][0000−0003−1149−7182]

[1] University of Haifa, Israel adi.akavia@gmail.com
[2] TripleBlind, USA craigbgentry@gmail.com
[3] Algorand Foundation, USA shaih@alum.mit.edu
[4] Intuit Inc., Israel margarita.vald@cs.tau.ac.il

Abstract. Homomorphic encryption (HE) protects data in-use, but can be computationally expensive. To avoid the costly bootstrapping procedure that refreshes ciphertexts, some works have explored client-aided outsourcing protocols, where the client intermittently refreshes ciphertexts for a server that is performing homomorphic computations. But is this approach secure against malicious servers?

We present a CPA-secure encryption scheme that is completely insecure in this setting. We define a new notion of security, called *funcCPA*, that we prove is sufficient. Additionally, we show:

- Homomorphic encryption schemes that have a certain type of circuit privacy – for example, schemes in which ciphertexts can be "sanitized" – are funcCPA-secure.
- In particular, assuming certain existing HE schemes are CPA-secure, they are also funcCPA-secure.
- For certain encryption schemes, like Brakerski-Vaikuntanathan, that have a property that we call oblivious secret key extraction, funcCPA-security implies circular security – i.e., that it is secure to provide an encryption of the secret key in a form usable for bootstrapping (to construct fully homomorphic encryption).

Namely, funcCPA-security lies strictly between CPA-security and CCA2-security (under reasonable assumptions), and has an interesting relationship with circular security, though it is not known to be equivalent.

1 Introduction

Homomorphic encryption (HE) supports computing over encrypted data without access to the secret key. HE is a prominent approach to safeguarding data and minimizing the impact of potential breaches, especially useful for outsourcing of computations over sensitive data, as required by the industry cloud-based architecture.

* The first author thanks the Israel Science Foundation (grant 3380/19) and Israel National Cyber Directorate via the Haifa, BIU and Tel-Aviv cyber centers for their support. The fourth author thanks Yaron Sheffer for helpful discussions. Pre-prints for preliminary versions of this works appeared in [3,8,2].

The security notion achievable for HE schemes is security against chosen-plaintext attack (CPA-security), whereas it is well known that security against chosen-ciphertext attack (CCA2-security) is not achievable due to the inherent malleability of HE schemes. However, CPA-security is not always sufficient for securing protocols, as it considers only honestly generated ciphertexts and has no guarantees in settings where an adversary is allowed to inject its own maliciously crafted ciphertexts into an honest system (see e.g. [41], Chapter 10). Therefore, relying on CPA-security typically secures protocols only against semi-honest adversaries e.g. in [43,9,1,26,7,30,5] (unless further cryptographic tools are employed to enhance security).

In practice however security against malicious adversaries is desired to combat real-life attacks. A natural question therefore is the following:

> *Is there a relaxation of CCA2-security that is achievable for HE schemes and secures protocols against malicious attackers?*

Our Contribution. In this work we answer affirmatively the above question by providing a new security notion, showing it is achievable for HE schemes and that it guarantees privacy against malicious adversaries for a wide and natural family of protocols.

The new security notion, named *function-chosen-plaintext-attack* (funcCPA-security), is a relaxation of CCA2 security for public key encryption schemes. Concretely, while CCA2 security captures resiliency against adversaries that receive decryptions of ciphertexts of their choice, funcCPA guarantees resiliency only against adversaries that receive re-encryptions of the underlying cleartext values of ciphertexts of their choice (or, more generally, encryptions of the result of a computation on those values); See Definition 6. That is, in funcCPA the adversary sees only ciphertexts, no cleartext values; nonetheless, the adversary has full control on the computation performed on the underlying values, even without knowing them, and can inject maliciously crafted ciphertexts.

We note that funcCPA-security is clearly implied by CCA2, moreover, we show it is a strict weakening of CCA2 by showing it is achievable for HE schemes (where CCA2-security is not). Furthermore, funcCPA-security implies CPA-security, but not vice-versa. To prove the latter, we provide: (1) a security proof showing, for a wide and natural family of outsourcing protocols (named, *client-aided outsourcing protocols*), that they preserve privacy when instantiated with any funcCPA-secure encryption scheme; and (2) an attack that breaks privacy in these protocols when instantiated with a (carefully crafted) CPA-secure encryption scheme. This shows that funcCPA-security lies strictly between CPA and CCA2 security.

To prove that funcCPA is achievable for HE schemes we show how to construct funcCPA-secure HE schemes from any CPA-secure HE scheme equipped with a sanitization algorithm, including the HE schemes of Gentry [24], Brakerski-Vaikuntanathan [13] and Ducas and Micciancio [20] (where sanitization is as defined in [21], see Definition 3).

Theorem 1 (funcCPA-secure HE scheme achievability, informal). *Every CPA-secure HE scheme with a sanitization algorithm can be transformed into a funcCPA-secure HE scheme.*

To further motivate the definition of funcCPA-security we note that many secure outsourcing protocols in the literature provide the server with the capability of seeing re-encryptions of ciphertexts of its choice, and even encrypted results of computations performed on the underlying values of such ciphertexts. For example, in [43] the client provides the server with re-encryptions for ciphertexts of the server's choice, with the goal of avoiding costly bootstrapping at the server's side. Likewise, in [9,1,5,26,7,30] the server obtains, via interaction with the client, the encrypted results of applying various computations on the underlying cleartext values of ciphertexts of its choice, including computing comparisons [9], minima [1,5], linear equations solutions [26,7], ReLU [30].

To capture and generalize secure outsourcing protocols such as discussed above [9,1,5,26,7,30], we define a natural family of protocols named: *client-aided outsourcing protocols*. This family consists of all protocols where a client generates keys and uploads encrypted data to a server; the server executes computations over the encrypted data and sends encrypted results to the client; moreover, the server may send the client (typically few and lightweight) queries of the form $(\mathbf{e}, G)$, for $\mathbf{e}$ a vector of ciphertexts and G a function, so that the client computes G on the underlying cleartext values and sends the server the encrypted result $\mathbf{e}' \leftarrow \mathsf{Enc}_{pk}(G(\mathsf{Dec}_{sk}(\mathbf{e})))$.

We prove that client-aided outsourcing protocols instantiated with funcCPA-secure schemes preserve privacy against malicious servers.

Theorem 2 (privacy against malicious servers, informal). *Client-aided outsourcing protocols instantiated with any funcCPA-secure scheme preserve privacy against malicious servers.*

Conversely, the attack we exhibit exemplifies that CPA-security does not provide privacy against malicious servers for this class of protocols.

Theorem 3 (attack, informal). *There exist CPA-secure HE schemes so that for client-aided outsourcing protocols instantiated with these schemes, there is an attack by the server that recovers the client's input.*

Achievability by existing schemes of funcCPA-security. To avoid the performance overhead incurred due to using sanitization we examine the achievability of funcCPA-security for popular HE schemes. We prove that the leveled HE schemes of BV [13], BGV [12] and B/FV[11,22] are leveled-funcCPA-secure (based on their CPA-security). That is, they satisfy a natural adaptation of funcCPA to leveled settings, where the funcCPA oracle answers queries with ciphertexts for the next level.[5] Our security proof requires essentially no modifications to the schemes

[5] This leveled-funcCPA oracle is useful, for example, in applications where the oracle is employed to replace deep homomorphic computations that will consume many levels of the scheme by a query to the oracle that consumes only a single level.

(other than a slight change in their evaluation keys generation that has little influence on performance) and without any extra security assumptions.

Theorem 4 (leveled HE are leveled-funcCPA-secure, informal). *The leveled HE schemes of BV, BGV, B/FV are leveled-funcCPA-secure.*

More generally, the above holds for every leveled HE scheme with keys generated independently for each level (as specified in Definition 9).

In contrast, for the homomorphic schemes of BV and BGV we show that funcCPA-security implies (weak) circular security. Concretely, we show that the funcCPA oracle enables generating from the public key an encryption of the secret key (in the encoding required for bootstrapping), and thus funcCPA-security eliminates the need for the weak circular security assumption. This can be interpreted as a barrier on proving funcCPA-security for these schemes, as it would resolve the long standing open problem on the necessity of circular security assumption (see e.g. Question 11 in Peikert's survey [38]).

Theorem 5 (funcCPA vs. circular security, informal). *If the homomorphic encryption scheme of BV or BGV is funcCPA-secure, then it is weakly circular secure.*

On the necessity of funcCPA against semi-honest adversaries. To further study the funcCPA-security notion, we examine its necessity against semi-honest adversaries. We prove that for client-aided outsourcing protocols satisfying a natural property, CPA-security suffices against semi-honest adversaries. The property we require is that the protocol is *cleartext computable* in the sense that the client's input determines the underlying cleartext values of the ciphertexts transmitted throughout the protocol. This captures the fact that the encryption in the protocol is an external wrapping of the cleartext values, used merely for achieving privacy against the server, and does not affect the underlying cleartext computation. This property is natural in outsourcing protocols, where the server does not contribute any input to the computation but rather it is only a vessel for storing and processing encrypted data on behalf of the client.

Theorem 6 (privacy against semi-honest servers, informal).
Cleartext-computable client-aided outsourcing protocols using a CPA-secure encryption scheme preserve privacy against semi-honest servers.

Our Techniques. Our definition of funcCPA (Definition 6) extends CPA by granting the adversary in the CPA experiment access to an $\mathsf{Enc}_{pk}(\mathcal{G}(\mathsf{Dec}_{sk}(\cdot)))$ oracle for a family of functions $\mathcal{G}$. Namely, the adversary can submit (possibly, adaptive) queries $(\mathbf{e}, G)$, for ciphertexts $\mathbf{e}$ and a function $G \in \mathcal{G}$ of its choice, and receive an encrypted result $\mathbf{e}' \leftarrow \mathsf{Enc}_{pk}(G(\mathsf{Dec}_{sk}(\mathbf{e})))$.

To prove achievability of funcCPA for sanitized HE schemes (Theorem 1), we first define the notion of circuit-privacy$^+$ that lies between the semi-honest and malicious definitions of circuit privacy in allowing maliciously formed ciphertexts

but requiring honestly generated keys. We then show how to transform CPA-secure schemes with a sanitization algorithm into CPA-secure circuit-private$^+$ schemes (Lemma 2). Finally, we prove that CPA-secure circuit-private$^+$ schemes are funcCPA-secure.

For our attack proving the insufficiency of CPA-security (Theorem 3) we first show that every CPA-secure scheme can be slightly modified to yield a punctured CPA-secure scheme with which our attack is applicable. The attack uses a single query $\mathbf{e}' \leftarrow \mathsf{Enc}_{pk}(G(\mathsf{Dec}_{sk}(\mathbf{e})))$, where $\mathbf{e}$ is a concatenation of the client's encrypted input with a special "trapdoor" ciphertexts planted in the public-key. The query $\mathbf{e}$ hits the puncturing of the scheme so that the result $\mathbf{e}'$ reveals the client's input. The encryption scheme remains CPA secure, despite the puncturing, because the trapdoor ciphertext is infeasible to generate honestly i.e. by encrypting an efficiently samplable message.

Related Work. *Several CCA relaxations* were previously considered. Relaxing CCA2 by forbidding querying the decryption oracle on any ciphertext that decrypt to the same message as the challenge ciphertext (or extensions of this notion) was proposed in [42,14,39]. However, for HE this is unachievable (because the adversary can produce ciphertexts that decrypt to related messages, query the decryption oracle on those, and consequently recover the message in the challenge ciphertext).

CCA1-secure HE schemes were constructed in a line of work including [34,32]. This however seems insufficient for privacy against malicious servers in client-aided outsourcing protocols, because CCA1 does not guarantee security if non-trivial queries are submitted after the challenge. Moreover, CCA1 is known to be unachievable for *fully* homomorphic encryption schemes that follow Gentry's blueprint (because they provide an encryption of the secret key for the purpose of bootstrapping, and querying the CCA1 oracle on this ciphertext would recover the secret key and break security); and even when deviating from Gentry's blueprint, CCA1 is only known to be achievable from non-standard assumptions [15]: indistinguishability obfuscation (iO) or succinct non-interactive arguments of knowledge (SNARKs).

In contrast, we show that funcCPA-security is: (1) sufficient for guaranteeing privacy against malicious servers in client-aided outsourcing protocols; (2) achievable for HE schemes, even fully homomorphic ones that follow Gentry's blueprint; and (3) achievable from standard assumptions.

Insufficiency of CPA-security for protocols utilizing homomorphic encryption was considered by Li and Micciancio [33]. They show that protocols instantiated with the CPA-secure approximate HE schemes of CKKS [17] are insecure when the protocol exposes decryptions to the attacker, even for semi-honest adversaries. In contrast, our attack applies both to exact and approximate schemes and even when no decryptions are provided (albeit with a malicious adversary).

Prior versions of this work. Preliminary versions of this work appeared in [3,8,2]: The notion of funcCPA-security and its implication to privacy against malicious

servers (Theorem 2) was introduced in [3], in the context of presenting new privacy preserving machine learning protocols. We remark that these protocols were published in [4,6], albeit with security only against semi-honest servers. The study of funcCPA was extended in [8] by introducing the generic construction of funcCPA-secure encryption from sanitization (Theorem 1); proving the insufficiency of CPA-security for privacy against malicious servers (Theorem 3); and proving the sufficiency of CPA-security for privacy against semi-honest servers in cleartext computable client-aided outsourcing protocols (Theorem 6). Open problems presented in [8] were addressed in [2], where we proved that leveled HE schemes are (leveled) funcCPA-secure (Theorem 4), and introduced connections between funcCPA and circular-security (Theorem 5). In addition, [2] introduced the observation that funcCPA w.r.t the identity function (i.e., with an oracle that can only refresh ciphertexts) implies funcCPA w.r.t an oracle that can compute all the circuits for which the scheme is homomorphic (Lemma 1).

Follow-up work. A follow-up work by Nuida [36] proposed a different definition of funcCPA (albeit, using the same name funcCPA). It was shown in [36] that their definition does not guarantee privacy in client-aided outsourcing protocols, and the thrust of that work was to study several possible treatments of invalid ciphertexts.

We stress that the results from [36] have no bearing on our funcCPA definition, in particular we show in Theorem 2 that our definition does imply privacy for client-aided protocols. We note that our results hold regardless of how invalid ciphertexts are treated (as long as funcCPA holds wrt an oracle that uses the same treatment as the client in the protocol). See Remark 2. We also note that [36, Theorems 3 and 5] are special cases of [8, Theorem 7].

Paper organization. Preliminary definitions are given in Section 2. Our results on funcCPA definition, sufficiency and achievability in Section 3. Our result on the insufficiency of CPA against malicious adversaries in Sections 4, and on its sufficiency against semi-honest ones for natural protocols in Section 5. We conclude in Section 6.

2 Preliminaries

We briefly specify standard definitions. See details in our full version [2].

Terminology and notations. For $n \in \mathbb{N}$, we denote by $[n]$ the set $\{1, \ldots, n\}$. We use standard definitions (see e.g. Goldreich [27]) for *negligible* and *polynomial* functions with respect to the security parameter λ, denoted $\mathsf{neg}(\lambda)$ and $\mathsf{poly}(\lambda)$; *probabilistic polynomial time* algorithms, denoted ppt; *random variables*; *probability ensembles*; *computationally indistinguishability*; *statistical distance* denoted by $\Delta(\cdot,\cdot)$; and *(strong) one-way function*.

CPA-secure public key encryption. We use the standard definition for public key encryption (PKE) scheme $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ and its properties of *correctness*, *CPA-indistinguishability experiment* against an adversary $\mathcal{A}$ denoted $\mathsf{EXP}^{cpa}_{\mathcal{A},\mathcal{E}}(\lambda)$, and *CPA-security* for single and multiple messages. A scheme is *fully decryptable* if applying the decryption algorithm on any ciphertext in the ciphertext space returns an element from the message space (and requiring, in addition, that the ciphertext space is efficiently recognizable). See the formal definitions in [31].

Homomorphic encryption. A homomorphic public-key encryption scheme (HE) is a public-key encryption scheme equipped with an additional ppt algorithm called Eval that supports "homomorphic evaluations" on ciphertexts. The correctness requirement is extended to hold with respect to any sequence of homomorphic evaluations performed on ciphertexts. A fully homomorphic encryption scheme must satisfy an additional property called *compactness* requiring that the size of the ciphertext does not grow with the complexity of the sequence of homomorphic operations.

Definition 1 (Homomorphic encryption (HE)). *A homomorphic public-key encryption (HE) scheme* $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *with message space* $\mathcal{M}$ *is a tuple of ppt algorithms as follows:* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is a correct PKE.* Eval *(homomorphic evaluation) takes as input the public key* pk*, a circuit* $C\colon \mathcal{M}^\ell \to \mathcal{M}$*, and ciphertexts* $c_1, \ldots, c_\ell$*, and outputs a ciphertext* $\widehat{c} \leftarrow \mathsf{Eval}_{pk}(C, c_1, \ldots, c_\ell)$*.*

The scheme is secure *if it is a CPA-secure PKE;* compact *if its decryption circuit is of polynomial size (in the security parameter);* $\mathcal{C}$-homomorphic *for a circuit family* $\mathcal{C}$ *if for all* $C \in \mathcal{C}$ *and all inputs* $x_1, \ldots, x_\ell$ *to* C*, letting* $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$ *and* $c_i \leftarrow \mathsf{Enc}(pk, x_i)$ *it holds that:*

$$\Pr[\mathsf{Dec}_{sk}(\mathsf{Eval}_{pk}(C, c_1, \ldots, c_\ell)) \neq C(x_1, \ldots, x_\ell)] \leq \mathsf{neg}(\lambda)$$

where the probability is taken over all the randomness in the experiment; and fully homomorphic *if it is compact and* $\mathcal{C}$*-homomorphic for* $\mathcal{C}$ *the class of all circuits.*

A $\mathcal{C}$-homomorphic encryption scheme is *bootstrappable* if it supports homomorphic evaluation of all circuits composed from copies of its decryption circuit connected by a single gate from the set of gates (see [23, Definitions 4.1.2-4.1.3]).

A HE scheme is *leveled* (leveled HE) if for each $L \in \mathbb{Z}^+$ given as an extra parameter to Gen, denoted $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda, 1^L)$, the scheme compactly evaluates all circuits of depth at most L. The complexity of its algorithms is polynomial in L on top of λ. CPA-security for leveled HE is defined similarly to the standard CPA definition except for the capability of the adversary to choose the level to which the challenge ciphertext is encrypted (to guarantee security of the scheme for all the levels). More formally,

The CPA indistinguishability experiment $\mathsf{EXP}^{cpa}_{\mathcal{A},\mathcal{E}}(\lambda, L)$ for leveled HE is parameterized by the security parameter λ and number of levels L, and executed between a challenger Chal and an adversary $\mathcal{A}$ as follows:

1. $\mathsf{Gen}(1^\lambda, 1^L)$ is run by Chal to obtain keys $(pk_\ell, sk_\ell)_{\ell\in\{0,\ldots,L\}}$ (we consider the public key pk_ℓ to include the evaluation key evk_ℓ if exists).
2. Chal provides the adversary $\mathcal{A}$ with $(pk_\ell)_{\ell\in\{0,\ldots,L\}}$. $\mathcal{A}$ sends to Chal two messages $x_0, x_1 \in \mathcal{M}$ s.t. $|x_0| = |x_1|$ and $\ell \in \{0,\ldots,L\}$.
3. Chal chooses a random bit $b \in \{0,1\}$, computes a ciphertext $c \leftarrow \mathsf{Enc}_{pk_\ell}(x_b)$ and sends c to $\mathcal{A}$. We call c the challenge ciphertext.
4. $\mathcal{A}$ outputs a bit b'.
5. The output of the experiment is defined to be 1 if $b' = b$ (0 otherwise).

Definition 2 (CPA security for leveled HE). *A leveled HE scheme $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ is* CPA-secure *if for every ppt adversary $\mathcal{A}$, there exists a negligible function neg such that for all sufficiently large λ and every L polynomial in λ,*

$$\Pr[\mathsf{EXP}^{cpa}_{\mathcal{A},\mathcal{E}}(\lambda, L) = 1] < \frac{1}{2} + \mathsf{neg}(\lambda)$$

where the probability is over all randomness in the experiment.

Sanitization. A ciphertext *sanitization* algorithm for a homomorphic encryption re-randomizes ciphertexts to make them statistically close to other (sanitized) ciphertexts decrypting to the same plaintext. Sanitization algorithms exists for most contemporary HE schemes [21].

Definition 3 (Sanitization algorithm [21]). *A Sanitize algorithm for a homomorphic public-key encryption scheme $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ is a ppt algorithm that takes a public key pk and a ciphertext c and returns a ciphertext, so that with probability $\geq 1 - \mathsf{neg}(\lambda)$ over the choice of $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$ the following holds:*

- (Message-preservation) $\forall c$ *in the ciphertext space:* $\mathsf{Dec}_{sk}(\mathsf{Sanitize}_{pk}(c)) = \mathsf{Dec}_{sk}(c)$.
- (Sanitization) $\forall c, c'$ *in the ciphertext space s.t.* $\mathsf{Dec}_{sk}(c) = \mathsf{Dec}_{sk}(c')$*:*

$$\Delta\left((\mathsf{Sanitize}_{pk}(c), (pk, sk)), (\mathsf{Sanitize}_{pk}(c'), (pk, sk))\right) \leq \mathsf{neg}(\lambda).$$

Interactive client-server protocols. The protocols considered in this work involve two-parties, client and server, denoted by Clnt and Srv respectively, where the client has input and output, the server has no input and no output, and both receive the security parameter λ. The client and server interact in an interactive protocol denoted by $\pi = \langle \mathsf{Clnt}, \mathsf{Srv} \rangle$. The server's view in an execution of π, on client's input x, no server's input (denoted by $\perp$), and security parameter λ, is a random variable $\mathsf{view}^\pi_{\mathsf{Srv}}(x, \perp, \lambda)$ capturing what the server has learned, and defined by $\mathsf{view}^\pi_{\mathsf{Srv}}(x, \perp, \lambda) = (r, m_1, \ldots, m_t)$where r is the random coins of Srv, and $m_1, \ldots, m_t$ are the messages Srv received during the protocol's execution. The client's output in the execution is denoted by $\mathsf{out}^\pi_{\mathsf{Clnt}}(x, \perp, \lambda)$. The protocol preserves privacy if the views of any server on (same length) inputs are computationally indistinguishable [28, Definition 2.6.2 Part 2]:[6]

[6] The server has no input or output, so we do not require security against the client.

Definition 4 (Correctness and privacy). *An interactive client-server protocol $\pi = \langle \mathsf{Clnt}, \mathsf{Srv} \rangle$ for computing $F : \mathsf{A} \to \mathsf{B}$, where the server has no input or output is said to be:*

Correct: *if* Srv *and* Clnt *are ppt and for all* $x \in \mathsf{A}$,
$\Pr[\mathit{out}^{\pi}_{\mathsf{Clnt}}(x, \bot, \lambda) = F(x)] > 1 - \mathsf{neg}(\lambda)$.

Private: *if for every ppt server* Srv^* *and every ppt distinguisher* $\mathcal{D}$ *that chooses* $x_0, x_1 \in \mathsf{A}$ *s.t.* $|x_0| = |x_1|$, *there exists a negligible function* $\mathsf{neg}(\cdot)$ *such that for every* $\lambda \in \mathbb{N}$, *it holds that:*

$$|\Pr[\mathcal{D}(\mathit{view}^{\pi}_{\mathsf{Srv}^*}(x_0, \bot, \lambda)) = 1] - \Pr[\mathcal{D}(\mathit{view}^{\pi}_{\mathsf{Srv}^*}(x_1, \bot, \lambda)) = 1]| \leq \mathsf{neg}(\lambda)$$

where the probability is taken over the random coins of Clnt *and* Srv^*.

Definition 4 captures malicious adversaries, but can be relaxed to semi-honest ones by quantifying only over the prescribed Srv rather than every ppt Srv^*. We call the former *privacy against malicious servers* and the latter *privacy against semi-honest servers*.

Client-aided outsourcing protocols. We formally define the family of *client-aided outsourcing protocols*, or $(\mathcal{E}, \mathcal{G})$*-aided outsourcing protocols*, parameterized by a PKE scheme $\mathcal{E}$ with message space $\mathcal{M}$ and a family of functions $\mathcal{G} = \{G_n : \mathcal{M} \to \mathcal{M}\}_{n \in \mathbb{N}}$. We note that $\mathcal{E}$ can be any PKE scheme (i.e., not necessarily an HE scheme).

Definition 5 ($(\mathcal{E}, \mathcal{G})$-aided outsourcing protocol). *Let* $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a public-key encryption scheme with message space* $\mathcal{M}$, *and* $\mathcal{G} = \{G_n : \mathcal{M} \to \mathcal{M}\}_{n \in \mathbb{N}}$ *a family of functions. An interactive client-server protocol* $\pi = \langle \mathsf{Clnt}, \mathsf{Srv} \rangle$ *for computing a function* $F : \mathsf{A} \to \mathsf{B}$ *is called an* $(\mathcal{E}, \mathcal{G})$-aided outsourcing protocol *if it has the following three stage structure:*

1. ***Client's input outsourcing phase (on input*** $x \in \mathsf{A}$***):*** Clnt *runs* $(pk, sk) \leftarrow \mathsf{Gen}(1^{\lambda})$, *encrypts its input* $\mathbf{c} \leftarrow \mathsf{Enc}_{pk}(x)$, *and sends* $\mathbf{c}$ *and* pk *to* Srv.
2. ***Server's computation phase:*** Srv *performs some computation and in addition may interact (multiple times) with* Clnt *by sending it pairs* $(\mathbf{e}, n)$, *for* $\mathbf{e}$ *a vector of ciphertexts and* $n \in \mathbb{N}$, *receiving in response* $\mathsf{Enc}_{pk}(G_n(\mathsf{Dec}_{sk}(\mathbf{e})))$.
3. ***Client's output phase:*** Srv *sends to* Clnt *the last message of the protocol; upon receiving this message,* Clnt *produces an output.*

Remark 1 (multiple inputs and outputs). The query $\mathbf{e}$ and response $\mathbf{e}'$ can be vectors of ciphertexts, with decryption and encryption in $\mathsf{Enc}_{pk}(G_n(\mathsf{Dec}_{sk}(\mathbf{e})))$ computed entry-by-entry. Throughout the paper we slightly abuse notations and denote by $\mathcal{M}$, Dec, Enc, $\mathbf{e}$ and $\mathbf{e}'$ also their extension to vectors.

3 A Sufficient and Achievable Relaxation of CCA2

In this section we formally define funcCPA-security and prove that client-aided protocols instantiated with a funcCPA-secure scheme preserve privacy against

malicious adversaries (Section 3.1); Show that funcCPA-secure HE is achievable from any HE equipped with a sanitization algorithm (Section 3.2); Prove that funcCPA-security is satisfied by all leveled schemes satisfying a natural property, e.g., the leveled HE schemes of BV [13], BGV [12] and B/FV [11,22] (Section 3.3); Conversely, show that funcCPA-security for homomorphic schemes with (another) natural property, e.g., the schemes of BV [13] and BGV [12], implies weak circular security (Section 3.4).

3.1 funcCPA-Security: A Sufficient Relaxation of CCA2

We define the *function-chosen-plaintext attack* (funcCPA-security) security notion of public-key encryption, and show that $(\mathcal{E}, \mathcal{G})$-aided outsourcing protocols preserve privacy against malicious servers if $\mathcal{E}$ is funcCPA-secure. We remark that $\mathcal{E}$ may be a PKE that is not necessarily a HE.

The definition captures a weaker adversary than the standard CCA2 adversary in the sense that the adversary has access to a "decrypt-function-encrypt" oracle, specified with respect to a family of functions, where the adversary may submit a ciphertext together with a function identifier and receive in response a ciphertext that is produced as follows. The submitted ciphertext is first decrypted, then the requested function is calculated on the plaintext and the result is encrypted and returned to the adversary.

More formally, we define funcCPA-security via a funcCPA-experiment specified for a public-key encryption scheme $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$, a family of functions $\mathcal{G} = \{G_n \colon \mathcal{M} \to \mathcal{M}\}_{n\in\mathbb{N}}$, and an adversary $\mathcal{A}$, as follows:

The funcCPA indistinguishability experiment $\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda)$*:*

1. $\mathsf{Gen}(1^\lambda)$ is run to obtain a key-pair (pk, sk)
2. The adversary $\mathcal{A}$ is given pk and access to a decrypt-function-encrypt oracle, denoted $\mathsf{Enc}_{pk}(\mathcal{G}(\mathsf{Dec}_{sk}(\cdot)))$, defined as follows: queries to $\mathsf{Enc}_{pk}(\mathcal{G}(\mathsf{Dec}_{sk}(\cdot)))$ are pairs consisting of a ciphertext $\mathbf{e}$ and a function index n, and the response is $\mathbf{e}' \leftarrow \mathsf{Enc}_{pk}(G_n(\mathsf{Dec}_{sk}(\mathbf{e})))$.
3. $\mathcal{A}$ outputs a pair of messages $x_0, x_1 \in \mathcal{M}$ with $|x_0| = |x_1|$.
4. A random bit $b \in \{0,1\}$ is chosen, and the ciphertext $c \leftarrow \mathsf{Enc}_{pk}(x_b)$ is computed and given to $\mathcal{A}$. We call c the challenge ciphertext. $\mathcal{A}$ continues to have access to the $\mathsf{Enc}_{pk}(\mathcal{G}(\mathsf{Dec}_{sk}(\cdot)))$ oracle.
5. The adversary $\mathcal{A}$ outputs a bit b'. The experiment's output is defined to be 1 if $b' = b$, and 0 otherwise.

Definition 6 (funcCPA). *A PKE scheme* $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *with message space* $\mathcal{M}$ *is* funcCPA-secure *w.r.t. a family of functions* $\mathcal{G} = \{G_n \colon \mathcal{M} \to \mathcal{M}\}_{n\in\mathbb{N}}$ *(*funcCPA-secure w.r.t. $\mathcal{G}$*) if for all ppt adversaries* $\mathcal{A}$*, there exists a negligible function* $\mathsf{neg}(\cdot)$ *such that for all sufficiently large* λ*,*

$$\Pr[\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda) = 1] \leq \frac{1}{2} + \mathsf{neg}(\lambda)$$

where the probability is taken over the random coins used by $\mathcal{A}$, as well as the random coins used to generate (pk, sk), choose b, and encrypt.

Remark 2 (Handling decryption errors). In Definitions 5 and 6 we do not include an explicit discussion of how decryption errors are treated. This is because our theorem showing that funcCPA implies privacy (Theorem 7) holds with any treatment of errors, as long as errors are treated identically by both the client in the client-aided outsourcing protocol and the oracle in the funcCPA-experiment. An example of a possible treatment of errors follows: if decryption fails on a query $(\mathbf{e}, n)$ submitted to the client or oracle, they return $\mathsf{Enc}_{pk}(G_n(m))$ for an arbitrary message $m \in \mathcal{M}$. Another example is provided in our preprint [3].

Theorem 7 (funcCPA implies privacy). *Let $\mathcal{E}$ be a PKE with message space $\mathcal{M}$ and $\mathcal{G} = \{G_n \colon \mathcal{M} \to \mathcal{M}\}_{n\in\mathbb{N}}$ a family of functions. If $\mathcal{E}$ is funcCPA-secure w.r.t. $\mathcal{G}$, then every $(\mathcal{E}, \mathcal{G})$-aided outsourcing protocol preserves privacy against malicious servers.*

Proof. Let π be a $(\mathcal{E}, \mathcal{G})$-aided outsourcing protocol for a function $F : \mathsf{A} \to \mathsf{B}$. Assume by contradiction that privacy does not hold for π. That is, there exists a ppt distinguisher $\mathcal{D}$ that chooses $x_0, x_1 \in \mathsf{A}$ with $|x_0| = |x_1|$, a malicious ppt server Srv^*, and a polynomial $p(\cdot)$ such that for infinitely many $\lambda \in \mathbb{N}$:

$$\Pr[\mathcal{D}(\mathsf{view}^{\pi}_{\mathsf{Srv}^*}(x_1, \perp, \lambda)) = 1] - \Pr[\mathcal{D}(\mathsf{view}^{\pi}_{\mathsf{Srv}^*}(x_0, \perp, \lambda)) = 1] \geq \frac{1}{p(\lambda)} \quad (1)$$

We show that given $\mathcal{D}$ and Srv^* we can construct an adversary $\mathcal{A}$ that violates the funcCPA security of $\mathcal{E}$ with respect to the family $\mathcal{G}$.

The adversary $\mathcal{A}$ participates in $\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{G}}$ as follows:

1. Upon receiving pk, $\mathcal{A}$ outputs x_0, x_1 (as computed by $\mathcal{D}$).
2. Upon receiving $\mathbf{c}_x \leftarrow \mathsf{Enc}_{pk}(x_b)$ from the challenger, $\mathcal{A}$ internally executes Srv^* and behaves as the Clnt in the execution of the protocol π: in the client's input outsourcing phase of π, $\mathcal{A}$ sends $(\mathbf{c}_x, pk)$ to Srv^*; in the server's computation phase of π, every incoming message $(\mathbf{e}, n)$ to Clnt is redirected to the oracle $\mathsf{Enc}_{pk}(\mathcal{G}(\mathsf{Dec}_{sk}(\cdot)))$ and the response is sent to Srv^* as if it were coming from Clnt.
3. $\mathcal{A}$ runs the distinguisher $\mathcal{D}$ on $\mathsf{view}_{\mathsf{Srv}^*}$ (Srv^*'s view in $\mathcal{A}$ during Step 2) and outputs whatever $\mathcal{D}$ outputs.

The adversary $\mathcal{A}$ is ppt due to Srv^* and $\mathcal{D}$ being ppt. Note that π is perfectly simulated.

We denote by $\mathsf{view}^{\mathsf{EXP}^{Fcpa}}_{\mathsf{Srv}^*}(x_b, \perp, \lambda)$ the view of Srv^*, simulated by $\mathcal{A}$, in the execution of $\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{G}}$ with bit b being selected by the challenger. Since $\mathcal{A}$ behaves exactly as Srv^* in π, it holds that for every $b \in \{0, 1\}$,

$$\Pr[\mathcal{D}(\mathsf{view}^{\pi}_{\mathsf{Srv}^*}(x_b, \perp, \lambda)) = 1] = \Pr[\mathcal{D}(\mathsf{view}^{\mathsf{EXP}^{Fcpa}}_{\mathsf{Srv}^*}(x_b, \perp, \lambda)) = 1] \quad (2)$$

From Equations 1 and 2 it follows that:

$$\Pr[\mathcal{D}(\mathsf{view}_{\mathsf{Srv}^*}^{\mathsf{EXP}^{Fcpa}}(x_1,\perp,\lambda))=1]-\Pr[\mathcal{D}(\mathsf{view}_{\mathsf{Srv}^*}^{\mathsf{EXP}^{Fcpa}}(x_0,\perp,\lambda))=1]\geq\frac{1}{p(\lambda)} \tag{3}$$

Therefore, we obtain that:

$$\Pr[\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda)=1]$$

$$=\frac{1}{2}\cdot\big(\Pr[\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda)=1|b=1]+\Pr[\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda)=1|b=0]\big)$$

$$=\frac{1}{2}\cdot\Big(\Pr[\mathcal{D}(\mathsf{view}_{\mathsf{Srv}^*}^{\mathsf{EXP}^{Fcpa}}(x_1,\perp,\lambda))=1]+\Pr[\mathcal{D}(\mathsf{view}_{\mathsf{Srv}^*}^{\mathsf{EXP}^{Fcpa}}(x_0,\perp,\lambda))=0]\Big)$$

$$=\frac{1}{2}+\frac{1}{2}\cdot\Big(\Pr[\mathcal{D}(\mathsf{view}_{\mathsf{Srv}^*}^{\mathsf{EXP}^{Fcpa}}(x_1,\perp,\lambda))=1]-\Pr[\mathcal{D}(\mathsf{view}_{\mathsf{Srv}^*}^{\mathsf{EXP}^{Fcpa}}(x_0,\perp,\lambda))=1]\Big)$$

$$\geq\frac{1}{2}+\frac{1}{2}\cdot\frac{1}{p(\lambda)}$$

where the last inequality follows from Equation 3. Combining this with $\mathcal{A}$ being ppt we derive a contradiction to $\mathcal{E}$ being funcCPA secure. This concludes the proof. □

We observe that for fully decryptable $\mathcal{C}$-homomorphic schemes, it suffices to prove funcCPA-security w.r.t the identity function $\mathcal{I}$ to obtain funcCPA-security w.r.t $\mathcal{C}$. We note that full decryption holds for well-known schemes including [40,12,11,25,20,18].

Lemma 1. *Let $\mathcal{E}=(\mathsf{Gen},\mathsf{Enc},\mathsf{Dec},\mathsf{Eval})$ be a fully decryptable [7] $\mathcal{C}$-homomorphic PKE scheme. If $\mathcal{E}$ is funcCPA-secure w.r.t the identity function $\mathcal{I}$ then it is funcCPA-secure w.r.t $\mathcal{C}$.*

Proof. Let $\mathcal{E}=(\mathsf{Gen},\mathsf{Enc},\mathsf{Dec},\mathsf{Eval})$ be a fully decryptable $\mathcal{C}$-homomorphic encryption scheme with message space $\mathcal{M}$ and ciphertext $\mathcal{T}$ that is funcCPA-secure w.r.t the identity function $\mathcal{I}:\mathcal{M}\to\mathcal{M}$. For any ppt adversary $\mathcal{A}$ that participates in $\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{C}}$ we construct an adversary $\mathcal{B}$ for $\mathsf{EXP}^{Fcpa}_{\mathcal{B},\mathcal{E},\mathcal{I}}$ that behaves as follows: The adversary $\mathcal{B}$ runs $\mathcal{A}$ internally while relaying messages between the challenger and $\mathcal{A}$, with the exception that $\mathsf{Enc}_{pk}(\mathcal{C}(\mathsf{Dec}_{sk}(\cdot)))$ queries are treated as follows: first the queried ciphertext is forwarded to the challenger that returns a fresh ciphertext of the encrypted value, then Eval is executed over this fresh ciphertext and the result ciphertext is forwarded again to the challenger that returns a fresh ciphertext for its underlying value. That is, $\mathcal{B}$ does the following:

[7] We note that the fully decryptable requirement addresses decryption errors. This requirement can be replaced by including in Definition 6 the following treatment of errors: in case of a decryption error, the funcCPA oracle returns an encryption of the queried function on an arbitrary message in the message space.

- Upon receiving pk from challenger, forward it to $\mathcal{A}$.
- Answer queries $(\mathbf{e}, n)$ to $\mathsf{Enc}_{pk}(\mathcal{C}(\mathsf{Dec}_{sk}(\cdot)))$ by sending $(\mathbf{e}, \mathcal{I})$ to the challenger and obtaining a fresh ciphertext $\mathbf{e}'$ (and $\perp$ if $\mathbf{e} \notin \mathcal{T}$), computing $\mathbf{e}'' \leftarrow \mathsf{Eval}_{pk}(C_n, \mathbf{e}')$ and sending $(\mathbf{e}'', \mathcal{I})$ to the challenger. The response to the second query is given to $\mathcal{A}$.
- Once $\mathcal{A}$ generates x_0, x_1 forward them to the challenger and return the response $c \leftarrow \mathsf{Enc}_{pk}(x_b)$ to $\mathcal{A}$.
- Output the b' that $\mathcal{A}$ outputs.

The adversary $\mathcal{B}$ is ppt (due to $\mathcal{A}$ and Eval being ppt), and all the interaction of $\mathcal{A}$ is perfectly simulated by $\mathcal{B}$ due to $\mathcal{E}$ being fully decryptable together with $\mathcal{C}$-homomorphic. More formally, letting $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$, for all $C \in \mathcal{C}$ and $c_1, \ldots c_\ell \in \mathcal{T}$ it holds that:

$$\Pr\left[\begin{matrix}\mathsf{Dec}_{sk}(\mathsf{Eval}_{pk}(C,\mathsf{Enc}_{pk}(\mathsf{Dec}_{sk}(c_1)),\ldots,\mathsf{Enc}_{pk}(\mathsf{Dec}_{sk}(c_\ell)))) \\ \neq \\ C(\mathsf{Dec}_{sk}(c_1),\ldots,\mathsf{Dec}_{sk}(c_\ell))\end{matrix}\right] \leq \mathsf{neg}(\lambda)$$

(if $\mathcal{A}$ submits a ciphertext not in $\mathcal{T}$ then the challenger's response is $\perp$ in both executions). Since the number of queries of $\mathcal{A}$ is polynomial in λ the indistinguishability of $\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{C}}(\lambda)$ and $\mathsf{EXP}^{Fcpa}_{\mathcal{B},\mathcal{E},\mathcal{I}}(\lambda)$ follows. Finally, from the funcCPA-security of $\mathcal{E}$ w.r.t $\mathcal{I}$ we conclude that

$$\Pr[\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{C}}(\lambda) = 1] \leq \frac{1}{2} + \mathsf{neg}(\lambda)$$

as required. □

3.2 Sanitized HE Schemes are funcCPA-Secure

We show how to transform any HE scheme $\mathcal{E}$ that has a sanitization algorithm into a sanitized HE scheme, denoted $\mathcal{E}^{\mathsf{santz}}$, so that if $\mathcal{E}$ is CPA-secure, then $\mathcal{E}^{\mathsf{santz}}$ is funcCPA-secure.

Definition 7 (Sanitized scheme $\mathcal{E}^{\mathsf{santz}}$). *Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be HE scheme with message space $\mathcal{M}$ and a sanitization algorithm Sanitize. We define its sanitized scheme $\mathcal{E}^{\mathsf{santz}} = (\mathsf{Gen}, \mathsf{Enc}^{\mathsf{santz}}, \mathsf{Dec}, \mathsf{Eval}^{\mathsf{santz}})$ as follows:* Gen *and* Dec *are as in $\mathcal{E}$;* $\mathsf{Enc}^{\mathsf{santz}}$ *takes a public key pk and a message $m \in \mathcal{M}$ and outputs:*

$$\mathsf{Enc}^{\mathsf{santz}}_{pk}(m) = \mathsf{Sanitize}_{pk}\left(\mathsf{Enc}_{pk}(m)\right);$$

$\mathsf{Eval}^{\mathsf{santz}}$ *takes pk, a circuit C, and ciphertexts $c_1, \ldots, c_\ell$ and outputs:*

$$\mathsf{Eval}^{\mathsf{santz}}_{pk}(C, c_1, \ldots, c_\ell) = \mathsf{Sanitize}_{pk}\left(\mathsf{Eval}_{pk}(C, \mathsf{Sanitize}_{pk}(c_1), \ldots, \mathsf{Sanitize}_{pk}(c_\ell))\right).$$

We note that $\mathcal{E}^{\mathsf{santz}}$ inherits the compactness, security and correctness properties of $\mathcal{E}$ (in particular, correctness holds due to correctness of $\mathcal{E}$ and the message-preservation property of Sanitize). The homomorphism of $\mathcal{E}^{\mathsf{santz}}$ may, in general, hold with respect to a subset of the circuits for which $\mathcal{E}$ is homomorphic. Nonetheless, when employing the sanitization algorithm of Ducas and Stehlé [21] both $\mathcal{E}$ and $\mathcal{E}^{\mathsf{santz}}$ are fully homomorphic.

Theorem 8 ($\mathcal{E}^{\mathsf{santz}}$ is funcCPA-secure). *Let $\mathcal{E}$ be a fully decryptable CPA-secure HE scheme with a sanitization algorithm; $\mathcal{E}^{\mathsf{santz}}$ its sanitized scheme. If $\mathcal{E}^{\mathsf{santz}}$ is $\mathcal{C}$-homomorphic, then it is funcCPA-secure w.r.t. $\mathcal{C}$.*[8]

Proof. To prove the theorem we first enhance the definition of circuit privacy to *circuit-privacy*$^+$ (cf. Definition 8 below); then show that the sanitized scheme $\mathcal{E}^{\mathsf{santz}}$ satisfies circuit-privacy$^+$ for $\mathcal{C}$ (cf. Lemma 2 below); and show that if a $\mathcal{C}$-homomorphic CPA-secure encryption scheme satisfies circuit-privacy$^+$ for $\mathcal{C}$, then it is funcCPA-secure w.r.t. $\mathcal{C}$ (cf. Lemma 3 below). We conclude that $\mathcal{E}^{\mathsf{santz}}$ is funcCPA-secure w.r.t. $\mathcal{C}$. □

Circuit-privacy$^+$*.* Our definition of circuit-privacy$^+$ addresses maliciously generated ciphertexts by quantifying over all ciphertexts in the ciphertext space, rather than only over ciphertexts that were properly formed by applying the encryption algorithm on a message. Prior definitions of circuit privacy either considered the semi-honest settings where both the keys and the ciphertext are properly formed [29,24,10], or considered settings where both keys and ciphertexts may be maliciously formed [29,37,19,35]. In contrast, in our settings the keys are properly formed whereas the ciphertexts may be maliciously formed.

Definition 8 (Circuit-privacy$^+$). *A $\mathcal{C}$-homomorphic PKE scheme $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ is* circuit-private$^+$ *for $\mathcal{C}$ if the following holds with probability $\geq 1 - \mathsf{neg}(\lambda)$ over the choice of $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$: For every circuit $C \in \mathcal{C}$ over ℓ inputs and ciphertexts $c_1, \ldots, c_\ell$ in the ciphertext space of $\mathcal{E}$ the following distributions are statistically close:*

$$\Delta\left(\mathsf{Enc}_{pk}\left(C\left(\mathsf{Dec}_{sk}(c_1), \ldots, \mathsf{Dec}_{sk}(c_\ell)\right)\right), \mathsf{Eval}_{pk}\left(C, c_1, \ldots, c_\ell\right)\right) \leq \mathsf{neg}(\lambda)$$

where the distributions are over the random coins of Enc *and* Eval*.*

We prove that the sanitized scheme $\mathcal{E}^{\mathsf{santz}}$ is circuit-private$^+$.

Lemma 2 ($\mathcal{E}^{\mathsf{santz}}$ is circuit-private$^+$). *Let $\mathcal{E}$ be a fully decryptable HE scheme with a sanitization algorithm, and $\mathcal{E}^{\mathsf{santz}}$ its sanitized scheme. If $\mathcal{E}^{\mathsf{santz}}$ is $\mathcal{C}$-homomorphic, then it is circuit-private$^+$ for $\mathcal{C}$.*

Proof. We highlight the key steps; the formal details appear in Appendix A.

To prove that $\mathcal{E}^{\mathsf{santz}}$ is circuit-private$^+$ we show that ciphertexts resulting from homomorphic evaluation *over maliciously crafted ciphertexts* are statistically close to those resulting from first decrypting then computing in cleartext and then encrypting the output. Sanitizing these ciphertexts (as done in $\mathcal{E}^{\mathsf{santz}}$) is aimed for guaranteeing this statistical closeness. However, the sanitization guarantee holds only if these ciphertexts *decrypt to the same message*; proving the latter is the heart of our proof.

We cannot rely on homomorphism to argue the latter, because correct evaluation is guaranteed only on "fresh" encryptions (cf. *maliciously* crafted ciphertexts

[8] We slightly abuse notations and allow funcCPA with respect to a circuit family.

as in our scenario). To address this issue we introduce a "hybrid" experiment, where we *decrypt-and-then-encrypt* the ciphertexts given as input to Eval, which guarantees that they are fresh encryptions. (We rely on full decryption to ensure that decryption yields some element in the message space.) In this hybrid experiment correct evaluation indeed holds.

To guarantee that correct evaluation holds even without re-encryption, we rely on the fact that in $\mathcal{E}^{\mathsf{santz}}$ we *sanitize also the input* to Eval and not just its output. This "inner" sanitization guarantees that the sanitized input ciphertexts are *statistically close* to those in the hybrid experiment (since they decrypt to the same message); from this (together with their statistical independent due to injecting fresh randomness in each sanitization) we derive that the ciphertext produced by the homomorphic evaluation is statistically close to the one produced in the hybrid experiment. This in turn implies that they decrypt to the same message. □

Circuit-privacy$^+$ implies funcCPA. We prove that a HE scheme is $\mathsf{funcCPA}$-secure if it is CPA-secure and circuit-private$^+$.

Lemma 3 (circuit-privacy$^+$ implies funcCPA). *Let $\mathcal{E}$ be a CPA-secure PKE. If $\mathcal{E}$ is $\mathcal{C}$-homomorphic and circuit-private$^+$ for $\mathcal{C}$, then $\mathcal{E}$ is funcCPA-secure w.r.t. $\mathcal{C}$.*

Proof. The main proof idea is to carefully replace $\mathsf{Enc}_{pk}(\mathcal{G}(\mathsf{Dec}_{sk}(\cdot)))$ oracle calls with Eval operations; details follow.

Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be a CPA-secure $\mathcal{C}$-homomorphic encryption scheme with message space $\mathcal{M}$ that is circuit-private$^+$ for $\mathcal{C}$. For any ppt adversary $\mathcal{A}$ that participates in $\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{C}}$ we construct an adversary $\mathcal{B}$ for $\mathsf{EXP}^{cpa}_{\mathcal{B},\mathcal{E}}$ that behaves as follows: The adversary $\mathcal{B}$ runs $\mathcal{A}$ internally while relaying messages between the challenger and $\mathcal{A}$, with the exception that $\mathsf{Enc}_{pk}(\mathcal{C}(\mathsf{Dec}_{sk}(\cdot)))$ queries are answered using Eval. That is, $\mathcal{B}$ does the following:

- Upon receiving pk from challenger, forward it to $\mathcal{A}$.
- Answer queries $(\mathbf{e}, n)$ to $\mathsf{Enc}_{pk}(\mathcal{C}(\mathsf{Dec}_{sk}(\cdot)))$ by $\mathbf{e}' \leftarrow \mathsf{Eval}_{pk}(C_n, \mathbf{e})$.
- Once $\mathcal{A}$ generates x_0, x_1 forward them to the challenger and return the response $\mathbf{c} \leftarrow \mathsf{Enc}_{pk}(x_b)$ to $\mathcal{A}$.
- Output the b' that $\mathcal{A}$ outputs.

The adversary $\mathcal{B}$ is ppt (due to $\mathcal{A}$ and Eval being ppt), and all the interaction of $\mathcal{A}$ is perfectly simulated by $\mathcal{B}$ except for the responses to queries to $\mathsf{Enc}_{pk}(\mathcal{C}(\mathsf{Dec}_{sk}(\cdot)))$ that are simulated using Eval. Circuit privacy$^+$ of $\mathcal{E}$ guarantees that these responses are indistinguishable from decrypting, applying C_n and encrypting the result.

More formally, we define a series of hybrid executions that gradually move between $\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{C}}$ experiment (where $\mathsf{Enc}_{pk}(\mathcal{C}(\mathsf{Dec}_{sk}(\cdot)))$ oracle is used) to $\mathsf{EXP}^{cpa}_{\mathcal{B},\mathcal{E}}$ experiment (where Eval is used). Let q denote an upper bound on the number of queries done by $\mathcal{A}$, we define $q+1$ hybrids as follows:

Hybrid H_0 is defined as the execution of $\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{C}}$.

Hybrid H_i is defined for $i \in [q]$. The hybrid H_i is defined as $\mathsf{EXP}^{Fcpa}_{\mathcal{A}_i,\mathcal{E},\mathcal{C}}$, where $\mathcal{A}_i$'s last i queries are answered using Eval instead of oracle $\mathsf{Enc}_{pk}(\mathcal{C}(\mathsf{Dec}_{sk}(\cdot)))$.

Note that H_q is equivalent to the CPA-experiment $\mathsf{EXP}^{cpa}_{\mathcal{B},\mathcal{E}}$, and hence,

$$\Pr[\mathsf{EXP}^{cpa}_{\mathcal{B},\mathcal{E}}(\lambda) = 1] = \Pr[\mathsf{EXP}^{Fcpa}_{\mathcal{A}_q,\mathcal{E},\mathcal{C}}(\lambda) = 1] \tag{4}$$

In each pair of adjacent hybrids H_{i-1} and H_i the difference is that in H_i the $(q-i+1)$'th query is done using Eval instead $\mathsf{Enc}_{pk}(\mathcal{C}(\mathsf{Dec}_{sk}(\cdot)))$ oracle. In this case the indistinguishability follows from $\mathcal{E}$ being circuit private$^+$ for $\mathcal{C}$. Namely,

$$|\Pr[\mathsf{EXP}^{Fcpa}_{\mathcal{A}_i,\mathcal{E},\mathcal{C}}(\lambda) = 1] - \Pr[\mathsf{EXP}^{Fcpa}_{\mathcal{A}_{i-1},\mathcal{E},\mathcal{C}}(\lambda) = 1]| \leq \mathsf{neg}(\lambda).$$

Since q is polynomial in λ, by the hybrid argument the indistinguishability of $\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{C}}$ and $\mathsf{EXP}^{cpa}_{\mathcal{B},\mathcal{E}}$ follows. Finally, from the CPA-security of $\mathcal{E}$ and Equation 4 we conclude that

$$\Pr[\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{C}}(\lambda) = 1] \leq \frac{1}{2} + \mathsf{neg}(\lambda)$$

As required. □

3.3 funcCPA Security of leveled HE Schemes

We show that CPA implies funcCPA for leveled HE schemes satisfying a natural property. This property is satisfied, e.g., by BV [13], BGV [12] and B/FV [11,22] (with a slight modification of their evaluation key), see Corollary 1.

Concretely, we address leveled HE schemes where each level is associated with a set of keys (usually, public, secret and evaluation keys), each ciphertext is associated with a (efficiently recognizable) level corresponding to the keys used for this ciphertext, and the scheme has independent level keys in the sense that the public and secret key pair can be sampled independently for each level, and the evaluation key for each level can be efficiently generated from the secret key for the current level and the public key for the next level.

Definition 9 (independent level keys). *We say that a leveled HE scheme* $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *has* independent level keys *if* Gen *(level key generation) takes as input the security parameter* 1^λ *and a number of levels* 1^L*, uses ppt algorithms* GenKey *and* $\mathsf{GenEvKey}$*, and outputs for each level* $\ell \in \{0,\ldots,L\}$ *a public key, secret key, and an evaluation key defined by:* $(pk_\ell, sk_\ell) \leftarrow \mathsf{GenKey}(1^\lambda)$ *and* $evk_\ell \leftarrow \mathsf{GenEvKey}(sk_\ell, pk_{\ell-1})$*denoted:* $(pk_\ell, evk_\ell, sk_\ell,)_{\ell\in[L]} \leftarrow \mathsf{Gen}(1^\lambda, 1^L)$

We reformulate the definition of funcCPA to capture security for leveled HE schemes (leveled-funcCPA) as follows: the adversary can choose the level to which the challenge ciphertext is encrypted, and the "decrypt-function-encrypt" oracle is modified to return a ciphertext for the next level. That is, to answer a query on a ciphertext of level ℓ, the ciphertext is first decrypted using sk_ℓ, then the requested function is calculated on the plaintext and the result is encrypted under the public-key for the next level $pk_{\ell-1}$ and returned to the adversary, see Definition 10.

The leveled-funcCPA indistinguishability experiment $\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda, L)$ for leveled HE is parameterized by the security parameter λ and number of levels L, and executed between a challenger Chal and an adversary $\mathcal{A}$:

1. $\mathsf{Gen}(1^\lambda, 1^L)$ is run to obtain keys $(pk_\ell, sk_\ell)_{\ell\in\{0,\ldots,L\}}$ (we consider the public key pk_ℓ to include the evaluation key evk_ℓ if it exists).
2. The adversary $\mathcal{A}$ is given $(pk_\ell)_{\ell\in\{0,\ldots,L\}}$ and access to a decrypt-function-encrypt oracle, denoted $\{\mathsf{Enc}_{pk_{\ell-1}}(\mathcal{G}(\mathsf{Dec}_{sk_\ell}(\cdot)))\}_{\ell\in[L]}$, defined as follows: the queries to this oracle are pairs $(\mathbf{e}_\ell, n)$ consisting of a ciphertext $\mathbf{e}_\ell$ of some level $\ell \in [L]$ (where the level is efficiently identifiable given the ciphertext) and a function index n, and the response is $\mathbf{e}' \leftarrow \mathsf{Enc}_{pk_{\ell-1}}(G_n(\mathsf{Dec}_{sk_\ell}(\mathbf{e}_\ell)))$.[9]
3. $\mathcal{A}$ outputs a pair of messages $x_0, x_1 \in \mathcal{M}$ s.t. $|x_0| = |x_1|$ and $\ell \in \{0, \ldots, L\}$.
4. A random bit $b \in \{0, 1\}$ is chosen, and the ciphertext $c \leftarrow \mathsf{Enc}_{pk_\ell}(x_b)$ is computed and given to $\mathcal{A}$. We call c the challenge ciphertext. $\mathcal{A}$ continues to have access to the oracle.
5. The adversary $\mathcal{A}$ outputs a bit b'. The experiment's output is defined to be 1 if $b' = b$ (0 otherwise).

Definition 10 (funcCPA for leveled HE). *A leveled HE scheme* $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *with message space* $\mathcal{M}$ *is* leveled-funcCPA-secure *with respect to a family of functions* $\mathcal{G} = \{G_n \colon \mathcal{M} \to \mathcal{M}\}_{n\in\mathbb{N}}$ *(*leveled-funcCPA-secure w.r.t. $\mathcal{G}$*) if for all ppt adversaries* $\mathcal{A}$*, there exists a negligible function* $\mathsf{neg}(\cdot)$ *such that for all sufficiently large* λ *and every* L *polynomial in* λ*,*

$$\Pr[\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda, L) = 1] < \frac{1}{2} + \mathsf{neg}(\lambda)$$

where the probability is taken over all random coins of the experiment.

We prove that CPA-secure leveled HE schemes with independent level keys are funcCPA-secure w.r.t any admissible family $\mathcal{G}$. *Admissible* here says that all $G_n \in \mathcal{G}$ are polynomial-time computable and have fixed output length $|G_n(x_0)| = |G_n(x_1)|$ for all $x_0, x_1 \in \mathcal{M}$. (We note that the latter trivially holds when $\mathcal{G}$ is a family of circuits.)

Theorem 9 (leveled HE is funcCPA). *Let* $\mathcal{E}$ *be a leveled HE scheme with independent level keys. If* $\mathcal{E}$ *is CPA-secure, then* $\mathcal{E}$ *is leveled-funcCPA-secure w.r.t. any admissible family* $\mathcal{G}$*.*

Proof. Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be a CPA-secure public-key leveled HE scheme with message space $\mathcal{M}$. Assume by contradiction that there exists an admissible family of functions $\mathcal{G} = \{G_n \colon \mathcal{M} \to \mathcal{M}\}_{n\in\mathbb{N}}$ over $\mathcal{M}$ such that $\mathcal{E}$ is not funcCPA-secure w.r.t $\mathcal{G}$. That is, there exists a ppt adversary $\mathcal{A}$ and a polynomial $p(\cdot)$ such that for infinity many λ and L it holds that:

$$\Pr[\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda, L) = 1] > \frac{1}{2} + \frac{1}{p(\lambda)} \tag{5}$$

[9] In case of an error, compute $\mathbf{e}' \leftarrow \mathsf{Enc}_{pk_{\ell-1}}(G_n(m))$ for an arbitrary $m \in \mathcal{M}$.

We show below that given $\mathcal{A}$ we can construct an adversary $\mathcal{B}$ that wins in $\mathsf{EXP}^{cpa}_{\mathcal{B},\mathcal{E}}(\lambda, L)$ with non-negligible advantage, violating the CPA security of $\mathcal{E}$.

The adversary $\mathcal{B}$ executes $\mathcal{A}$, relaying messages between the challenger and $\mathcal{A}$, while responding to any query $(\mathbf{e}_\ell, n)$ from $\mathcal{A}$ with an encryption using $pk_{\ell-1}$ of G_n on an arbitrary message $m \in \mathcal{M}$. That is $\mathcal{B}$ does the following,

- Upon receiving $(pk_\ell)_{\ell\in\{0,\ldots,L\}}$ from challenger, forward it to $\mathcal{A}$.
- Answer queries $(\mathbf{e}_\ell, n)$ for a ciphertext $\mathbf{e}_\ell$ of level ℓ by $\mathbf{e}' \leftarrow \mathsf{Enc}_{pk_{\ell-1}}(G_n(m))$ for an arbitrary $m \in \mathcal{M}$.
- Once $\mathcal{A}$ generates x_0, x_1 and ℓ forward them to the challenger and return the response $c \leftarrow \mathsf{Enc}_{pk_\ell}(x_b)$ to $\mathcal{A}$.
- Output the b' that $\mathcal{A}$ outputs.

The adversary $\mathcal{B}$ is ppt due to adversary $\mathcal{A}$ being ppt and admissibility of $\mathcal{G}$. Moreover all the interaction of $\mathcal{A}$ is perfectly simulated by $\mathcal{B}$ except for the responses to queries to $\{\mathsf{Enc}_{pk_{\ell-1}}(\mathcal{G}(\mathsf{Dec}_{sk_\ell}(\cdot)))\}_{\ell\in[L]}$ that are simulated using encryption of the image of G_n on an arbitrary message.

Let $\mathsf{EXP}^{Fcpa^\#}$ experiment denote this variant of EXP^{Fcpa} that is simulated by $\mathcal{A}$, namely $\mathsf{EXP}^{Fcpa^\#}$ is an experiment identical to EXP^{Fcpa} except that each query $(\mathbf{e}_\ell, n)$ to Chal is answered by the encryption of $G_n(m)$ under $pk_{\ell-1}$ for arbitrary $m \in \mathcal{M}$.

By definition of $\mathsf{EXP}^{Fcpa^\#}$ it holds that,

$$\Pr[\mathsf{EXP}^{Fcpa^\#}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda, L) = 1] = \Pr[\mathsf{EXP}^{cpa}_{\mathcal{B},\mathcal{E}}(\lambda, L) = 1] \tag{6}$$

Furthermore, the CPA security and independent level keys of $\mathcal{E}$ guarantees (as shown in Lemma 4 below) that $\mathcal{A}$'s winning probability in $\mathsf{EXP}^{Fcpa^\#}$ and EXP^{Fcpa} is computationally indistinguishable. In particular,

$$\begin{aligned} &|\Pr[\mathsf{EXP}^{Fcpa^\#}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda, L) = 1] \\ &-\Pr[\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda, L) = 1]| \leq \mathsf{neg}(\lambda)\ . \end{aligned} \tag{7}$$

Putting Equation 7 together with Equations 5-6 it follows that

$$\Pr[\mathsf{EXP}^{cpa}_{\mathcal{B},\mathcal{E}}(\lambda, L) = 1] \geq \frac{1}{2} + \frac{1}{p(\lambda)} - \mathsf{neg}(\lambda). \tag{8}$$

Combining this with $\mathcal{A}$ being ppt we derive a contradiction to $\mathcal{E}$ being CPA secure. This concludes the proof. □

Let $\mathsf{EXP}^{Fcpa^\#}$ be as defined in the proof of Theorem 9, i.e., it is identical to EXP^{Fcpa} except that Chal, upon receiving queries $(\mathbf{e}_\ell, n)$, instead of responding as in step 2 in Definition 10, responds by sending the encryption under $pk_{\ell-1}$ of $G_n(m)$ for an arbitrary message $m \in \mathcal{M}$ (rather then $m = \mathsf{Dec}_{sk_\ell}(\mathbf{e}_\ell)$). We show that the adversary is indifferent to the correctness of answers it receives from the Chal in the sense that its output distribution in EXP^{Fcpa} and $\mathsf{EXP}^{Fcpa^\#}$ is indistinguishable.

Lemma 4. *Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be a CPA-secure leveled HE scheme with a message space $\mathcal{M}$. Let $\mathcal{G} = \{G_n : \mathcal{M} \to \mathcal{M}\}_{n\in\mathbb{N}}$ be a family of admissible functions. If $\mathcal{E}$ has independent level keys then for any ppt adversary $\mathcal{A}$, there exists a negligible function $\mathsf{neg}(\cdot)$ such that for all sufficiently large λ and every L polynomial in λ the following holds:*

$$|\Pr[\mathsf{EXP}^{Fcpa^{\#}}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda, L) = 1] - \Pr[\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda, L) = 1]| \leq \mathsf{neg}(\lambda)$$

Proof. The proof relies on keys independence; details appear in the full version [2]. □

Schemes with independent level keys. In BV, BGV and B/FV, for example, indeed each ciphertext is associated with a level and there are independent encryption and decryption keys (pk_ℓ, sk_ℓ) for each level ℓ. Moreover, the evaluation key evk_ℓ (called key switching in BV, BGV and B and re-linearization keys in FV) is essentially the encryption of an efficiently computable function of the secret key sk_ℓ of the current level (concretely, the encryption of $sk'_\ell = \mathsf{Powersof2}(sk_\ell \otimes sk_\ell)$) under the public key $pk_{\ell-1}$ for the next level.

More accurately, to generate evk_ℓ they use a *fresh* public key $pk'_{\ell-1}$ with which they mask sk'_ℓ. This is important when instantiating their scheme as a fully homomorphic encryption, i.e., when there's a single key tuple (pk, evk, sk) used for all levels, in which case using pk (rather than pk') to encryt a function of sk would require a circular security assumption. In contrast, when using these schemes as a leveled HE, as we do, then anyhow the keys (pk_ℓ, sk_ℓ) are sampled independently from $(pk_{\ell-1}, sk_{\ell-1})$, and so encrypting sk'_ℓ under $pk_{\ell-1}$ requires no circular security assumption. Therefore, their generation of the evaluation keys can be modified to output the encryption of sk'_ℓ under $pk_{\ell-1}$, without harming correctness or security.[10] With this slight modification indeed these scheme satisfy Definition 9.

Proposition 1. *The leveled HE schemes of BV, BGV and B/FV [13,12,11,22] (with the aforementioned evaluation key) have independent level keys.*

Corollary 1. *The leveled HE schemes of BV, BGV and B/FV [13,12,11,22] (with the aforementioned evaluation key) are leveled-funcCPA-secure.*

3.4 Barriers on Proving funcCPA for Existing HE Schemes

In this section we prove that if the homomorphic encryption scheme of BV [13] or BGV [12] is funcCPA-secure, then it is (weakly) circular secure. More generally, we show the above holds for all schemes satisfying a property we call *oblivious secret key extraction (ObvSK)*. In the following we first formally define weak circular security and ObvSK; then prove that for schemes supporting ObvSK, funcCPA-security w.r.t a proper family $\mathcal{F}$ implies weak circular security; and conclude by showing that the schemes of BV and BGV support ObvSK.

[10] We remark that the noise in the modified evaluation keys is slightly larger: the noise of a fresh ciphertext, rather than a sample from the error distribution; nonetheless, this makes essentially no difference when using the scheme.

Circular security extends CPA-security to capture security of public key encryption schemes against adversaries seeing an encryption of the secret key [16, Definition 2.5].

Circular security is required by all fully homomorphic encryption schemes following Gentry's [23] blueprint, as they publish an encryption of the secret key to be used during bootstrapping (where bootstrapping [23] is the process of homomorphically evaluating the scheme's decryption circuit with a hardwired ciphertext on an encrypted secret key as input). Specifically, they require security to hold against adversaries seeing an encryption of the secret key in the encoding by which it is specified as input to the decryption circuit (see [13, Definition 3.8]).

Weak circular security is formally stated, for a public key encryption scheme $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, using the following experiment between a challenger Chal and an adversary $\mathcal{A}$ (where sk denotes the secret key when specified in the encoding as required for the decryption circuit):

The weak circular indistinguishability experiment $\mathsf{EXP}^{wc}_{\mathcal{A},\mathcal{E}}(\lambda)$*:*

1. Chal computes $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$ and $\mathbf{c}_{sk} \leftarrow \mathsf{Enc}_{pk}(\mathsf{sk})$, and sends $(pk, \mathbf{c}_{sk})$ to $\mathcal{A}$.
2. $\mathcal{A}$ sends to Chal two messages x_0, x_1 s.t. $|x_0| = |x_1|$.
3. Chal chooses a random bit $b \in \{0, 1\}$, computes a ciphertext $c \leftarrow \mathsf{Enc}_{pk}(x_b)$ and sends c to $\mathcal{A}$. We call c the challenge ciphertext.
4. $\mathcal{A}$ outputs a bit b'.
5. The output of the experiment is defined to be 1 if $b' = b$ (0 otherwise).

Definition 11 (weak circular security). *A PKE scheme* $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *is* weakly circular secure *if for every ppt adversary* $\mathcal{A}$*, there exists a negligible function* $\mathsf{neg}(\cdot)$ *such that for all sufficiently large* λ*,*

$$\Pr[\mathsf{EXP}^{wc}_{\mathcal{A},\mathcal{E}}(\lambda) = 1] \leq \frac{1}{2} + \mathsf{neg}(\lambda)$$

where the probability is taken over the random coins of $\mathcal{A}$ *and* Chal*.*

Oblivious secret key extraction captures the ability to generate, from the public key, ciphertexts encrypting data related to the secret key, so that from their decryption one can efficiently compute the secret key in the encoding as required for the decryption circuit.

Definition 12 (oblivious secret key extraction (ObvSK)). *Let* $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a PKE scheme with message space* $\mathcal{M}$*, and* $\mathcal{F} = \{F_n : \mathcal{M} \to \mathcal{M}\}_{n\in\mathbb{N}}$ *be a family of functions. We say that* $\mathcal{E}$ *supports* oblivious secret key extraction (ObvSK) w.r.t $\mathcal{F}$ *if there exists a ppt algorithm* Alg *that takes a public key* pk *and outputs* $n = n(\lambda)$ *ciphertexts under* pk*, so that the following holds. There exists a negligible function* $\mathsf{neg}(\cdot)$ *such that for all* $\lambda \in \mathbb{N}$ *the following holds:*

$$\Pr\left[\begin{array}{c}(pk,sk)\leftarrow\mathsf{Gen}(1^\lambda)\\(c_1,\ldots,c_n)\leftarrow\mathsf{Alg}(pk)\\F_n(\mathsf{Dec}_{sk}(c_1),\ldots,\mathsf{Dec}_{sk}(c_n))=\mathsf{sk}\end{array}\right] \geq 1 - \mathsf{neg}(\lambda) \tag{9}$$

where the secret key sk *outputted by* F_n *is in the encoding required for the decryption circuit, and where the probability is taken over the randomness in* Gen *and* Alg.

funcCPA-security for schemes supporting ObvSK implies weak circular security. Next we show that if a public key encryption scheme $\mathcal{E}$ support ObvSK w.r.t $\mathcal{F}$ and is funcCPA-secure w.r.t $\mathcal{G}$ that contains $\mathcal{F}$, then $\mathcal{E}$ is weakly circular secure.

Theorem 10. *Let* $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a PKE scheme that is funcCPA-secure w.r.t a family of functions* $\mathcal{G}$*. If* $\mathcal{E}$ *is ObvSK w.r.t* $\mathcal{F}$ *and* $\mathcal{F} \subseteq \mathcal{G}$ *then* $\mathcal{E}$ *is weakly circular-secure.*

Proof. The proof idea is, given pk, to first use Alg (from the ObvSK property) to get encrypted data related to sk; then use $\mathsf{Enc}_{pk}(\mathcal{G}(\mathsf{Dec}_{sk}(\cdot)))$ (from the funcCPA property) to transform them to ciphertexts $\mathbf{c}_{sk}$ encrypting sk (in the encoding for the decryption circuit); finally show that –if the scheme is not circular secure– then using $\mathbf{c}_{sk}$ we can break funcCPA-security. The formal details follow.

Suppose by contradiction that $\mathcal{E}$ is not circular-secure, i.e., there exists a ppt adversary $\mathcal{A}$ that wins $\mathsf{EXP}^{wc}_{\mathcal{A},\mathcal{E}}$ with non-negligible advantage over a random guess. We construct an adversary $\mathcal{B}$ that runs $\mathcal{A}$ internally and breaks funcCPA-security of the scheme.

The adversary $\mathcal{B}$ participates in the funcCPA-security experiment as follows. First, given pk from Chal, $\mathcal{B}$ computes $(c_1, \ldots, c_n) \leftarrow \mathsf{Alg}(pk)$ (for Alg as guaranteed by the ObvSK property), sends a query $((c_1, \ldots, c_n), n)$ to the $\mathsf{Enc}_{pk}(\mathcal{G}(\mathsf{Dec}_{sk}(\cdot)))$ oracle (provided as part of the funcCPA experiment), and receives in response (the vector of ciphertexts)

$$\mathbf{c}_{sk} = \mathsf{Enc}_{pk}(F_n(\mathsf{Dec}_{sk}(c_1), \ldots, \mathsf{Dec}_{sk}(c_n))),$$

which is an encryption of the secret key sk in the encoding as needed for bootstrapping with $1-\mathsf{neg}(\lambda)$ probability (by the ObvSK property). Next $\mathcal{B}$, internally runs $\mathcal{A}$, while providing to it $\mathbf{c}_{sk}$ together with pk, relaying messages between $\mathcal{A}$ and Chal, and outputting the guess b' outputted by $\mathcal{A}$.

The view of $\mathcal{A}$ in $\mathsf{EXP}^{Fcpa}_{\mathcal{B},\mathcal{E}}$ is identical to its view in $\mathsf{EXP}^{wc}_{\mathcal{A},\mathcal{E}}$ (except with a $\mathsf{neg}(\lambda)$ probability, for the case of failure in the ObvSK). Implying (by the contradiction assumption)

$$\Pr[\mathsf{EXP}^{Fcpa}_{\mathcal{B},\mathcal{E}}(\lambda) = 1] > \frac{1}{2} + \frac{1}{p(\lambda)}$$

for some polynomial $p(\cdot)$, in contradiction to the funcCPA-security of $\mathcal{E}$. □

As a corollary from Theorem 10 we conclude that for bootstrappable ObvSK schemes, funcCPA-security implies full homomorphism without relying on any circular security assumption.

Corollary 2. *Let* $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ *be a bootstrappable HE scheme that supports ObvSK w.r.t* $\mathcal{F}$*. If* $\mathcal{E}$ *is funcCPA-secure w.r.t* $\mathcal{G}$ *and* $\mathcal{F} \subseteq \mathcal{G}$ *then* $\mathcal{E}$ *is fully homomorphic.*

Proof. The proof is derived by combining the following two facts. First, by Theorem 4.3.2 in [23], bootstrappable HE schemes that are weakly circular secure are fully homomorphic. Second, by Theorem 10, if $\mathcal{E}$ support ObvSK w.r.t $\mathcal{F}$ and it is funcCPA-secure w.r.t $\mathcal{G}$ that contains $\mathcal{F}$, then $\mathcal{E}$ is weakly circular secure. Combining the above, we conclude that $\mathcal{E}$ is fully homomorphic. □

Schemes supporting ObvSK. BV and BGV are examples of schemes supporting ObvSK. More generally, we show that ObvSK is supported by all public key encryption schemes $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ satisfying the following:

1. The secret key $sk = (1, s)$ and ciphertext c are from the ring:
   - LWE-based schemes: $\mathbb{Z}_q^{n+1}$
   - RLWE-based schemes: R_q^2 for $R_q = \mathbb{Z}_q[x]/F[X]$

   where q, n, d are positive integers, d a power of 2, $F[X] = X^d + 1$, and s has small coefficients in the sense that decryption correctness holds on ciphertexts encrypting each coefficient of s.
2. Decryption is via inner-product (with messages encoded in the least significant bits): $\mathsf{Dec}_{sk}(c) = \Big[\big[\langle c, sk\rangle\big]_q\Big]_p$ where $[z]_x$ is the remainder of z in division by x and p a positive integer.

In the following let $\mathcal{F}^{LWE} = \{F_n^{LWE} \colon \mathbb{Z}_q^n \to \{0,1\}^{n\cdot\lceil \log q\rceil}\}_{q,n}$ denote a family of functions that given $(s_1, \ldots, s_n) \in \mathbb{Z}_q^n$ outputs $sk = (1, s) \in \mathbb{Z}_q^{n+1}$ in the encoding as required by the decryption circuit in LWE-based schemes satisfying the above properties. Similarly, let $\mathcal{F}^{RLWE} = \{F_d^{RLWE} \colon R_q \to R_q^2\}_{q,d}$ denote a family of functions that given $(s'_{d-1}, \ldots, s'_0) \in R_q$ outputs $sk = (\mathbf{1}, (-s'_0, s'_{d-1}, \ldots, s'_1)) \in R_q^2$ in the encoding as required by the decryption circuit in the RLWE-based schemes satisfying the above properties. (Here $(s'_{d-1}, \ldots, s'_0)$ is a vector of coefficients specifying a polynomial $s'(X) \in R_q$, and $\mathbf{1}$ denotes the unit element in R_q.) Moreover, for a scheme $\mathcal{E}$ satisfying the above properties, either in the LWE-based or RLWE-based form, we use the short hand notation of denoting by $\mathcal{F}^{GLWE}$ the family $\mathcal{F}^{LWE}$ in case $\mathcal{E}$ is LWE-based, and $\mathcal{F}^{RLWE}$ otherwise.

Proposition 2. *Suppose $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ satisfies (1)-(2) above. Then $\mathcal{E}$ supports ObvSK w.r.t to $\mathcal{F}^{GLWE}$.*

Proof. The proof appears in the full version [2].

As an immediate corollary from Proposition 2 we obtain that the addressed schemes support ObvSK.

Corollary 3 (BV and BGV support ObvSK). *The HE schemes from BV [13] and BGV [12] support ObvSK w.r.t to $\mathcal{F}^{GLWE}$.*

Since these schemes are known to be bootstrappable, then combining Corollary 3 with Corollary 2 we derive that if they are funcCPA-secure then they are fully homomorphic.

Corollary 4. *If BV [13] or BGV [12] is funcCPA-secure w.r.t to $\mathcal{G}$ containing $\mathcal{F}^{GLWE}$, then it is fully homomorphic.*

4 CPA insufficiency against Malicious Adversaries

We show that CPA-security is insufficient for guaranteeing privacy in client-aided outsourcing protocols. For this purpose we construct a CPA-secure PKE scheme and exhibit an input-recovery attack that completely breaks privacy in client-aided outsourcing protocols instantiated with our scheme. In fact, we can transform any CPA-secure encryption scheme $\mathcal{E}$ with message space $\mathcal{M}$ of super polynomial size, using a one-way function f and any function G, into a CPA-secure encryption scheme $\mathcal{E}^f$ for which our attack works on any $(\mathcal{E}^f, \mathcal{G})$-aided outsourcing protocol for any $\mathcal{G}$ containing G . Moreover, if $\mathcal{E}$ was an HE scheme then so is $\mathcal{E}^f$. For simplicity of the presentation we concentrate on G being the identity function $\mathcal{I}$ for the construction of $\mathcal{E}^f$. The scheme $\mathcal{E}^f$ is similar to $\mathcal{E}$, except for the key difference that its encryption and decryption are "punctured" on a random point $m^* \in \mathcal{M}$, where its public key implicitly specifies m^* by augmenting it with $f(m^*)$ and $\mathsf{Enc}_{pk}(m^*)$.[11] See our construction in Figure 1 and Theorem 11. Our attack breaks security in the strong sense that the server is able to completely recover the client's input; See Theorem 12.

Theorem 11 (properties of $\mathcal{E}^f$). *For every PKE scheme $\mathcal{E}$ and one-way function f over the message-space of $\mathcal{E}$, the scheme $\mathcal{E}^f$ (cf. Figure 1) is a PKE scheme satisfying the following. If $\mathcal{E}$ is CPA-secure, compact, and $\mathcal{C}$-homomorphic, then $\mathcal{E}^f$ is CPA-secure, compact, and $\mathcal{C} \times \mathcal{C}$-homomorphic.*[12]

Proof. Correctness, compactness and homomorphism of $\mathcal{E}^f$ follow directly from the properties of $\mathcal{E}$. The CPA-security of $\mathcal{E}^f$ follows from the CPA-security of $\mathcal{E}$ and the one-wayness of f: the encryption in $\mathcal{E}^f$ is identical to encrypting pairs (m_1, m_2) of messages under $\mathcal{E}$, except if m_2 is a pre-image of $f(m^*)$, but the latter occurs with no more than a negligible probability due to f being a one-way function and m^* being a random message. See formal details in the full version [2]. □

We present our attack in which the server recovers the client's input in any $(\mathcal{E}^f, \mathcal{G})$-aided outsourcing protocol for $\mathcal{G}$ containing the identity function $\mathcal{I}$. We remark that our attack is applicable from every PKE $\mathcal{E}$, regardless of whether it is a HE scheme.

Theorem 12 (CPA-security does not imply privacy). *For every PKE scheme $\mathcal{E}$ with message-space $\mathcal{M}$ and every one-way function f over $\mathcal{M}$, there exists a CPA-secure PKE scheme $\mathcal{E}^f$ so that for every family of functions $\mathcal{G} = \{G_n \colon \mathcal{M} \to \mathcal{M}\}_{n \in \mathbb{N}}$ containing the identity function $\mathcal{I}$ and every $(\mathcal{E}^f, \mathcal{G})$-aided outsourcing protocol there is a server's strategy that recovers the client's input.*

[11] In case our $\mathcal{G}$ of interest does not contain the identity function, we slightly modify $\mathcal{E}^f$ by replacing each occurrence of $\mathsf{Enc}_{pk}(m^*)$ and $f(m^*)$ in Figure 1 with $\mathsf{Enc}_{pk}(G(m^*))$ and $f(G(m^*))$ respectively for an efficiently computable $G \in \mathcal{G}$, and slightly modify the proof by replacing each occurrence of $\mathcal{I}$ by G.

[12] We note that a $\mathcal{C} \times \mathcal{C}$-homomorphic encryption scheme is also $\mathcal{C}$-homomorphic, as we can embed $\mathcal{C}$ in $\mathcal{C} \times \mathcal{C}$, e.g., by mapping every $C \in \mathcal{C}$ into $(C, C) \in \mathcal{C} \times \mathcal{C}$.

$\underline{\mathsf{Gen}^f(1^\lambda)}$: Given 1^λ, output (pk^f, sk^f) computed as follows. Let $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$ and sample a uniformly random $m^* \in \mathcal{M}$. Set

$$pk^f := (pk, \mathsf{Enc}_{pk}(m^*), f(m^*)) \text{ and } sk^f := (sk, f(m^*)).$$

$\underline{\mathsf{Enc}^f_{pk^f}(m)}$: Given $m = (m_1, m_2) \in \mathcal{M} \times \mathcal{M}$, if $f(m_2) = f(m^*)$ then output (m_1, m_2), else output

$$(\mathsf{Enc}_{pk}(m_1), \mathsf{Enc}_{pk}(m_2)).$$

$\underline{\mathsf{Dec}^f_{sk^f}(c)}$: Given $c = (c_1, c_2)$, if $f(c_2) = f(m^*)$ then output (c_1, c_2), else output

$$(\mathsf{Dec}_{sk}(c_1), \mathsf{Dec}_{sk}(c_2)).$$

$\underline{\mathsf{Eval}^f_{pk^f}(C, c_1, ..., c_\ell)}$: Given a circuit $C = C_1 \times C_2$ over ℓ inputs, and ℓ ciphertexts $c_i = (c_{i,1}, c_{i,2})$ for $i \in [\ell]$, do the following. For each $i \in [\ell]$, if $f(c_{i,2}) = f(m^*)$ then set $c'_i = (\mathsf{Enc}_{pk}(c_{i,1}), \mathsf{Enc}_{pk}(c_{i,2}))$, else set $c'_i = c_i$. Output

$$(\mathsf{Eval}_{pk}(C_1, c'_{1,1}, ..., c'_{\ell,1}), \mathsf{Eval}_{pk}(C_2, c'_{1,2}, ..., c'_{\ell,2})).$$

Fig. 1. The construction of the scheme $\mathcal{E}^f = (\mathsf{Gen}^f, \mathsf{Enc}^f, \mathsf{Dec}^f, \mathsf{Eval}^f)$ from a PKE scheme $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ with message space $\mathcal{M}$ and ciphertext space $\mathcal{T}$ and a one-way function f over $\mathcal{M}$. The message-space and ciphertext-space of $\mathcal{E}^f$ are $\mathcal{M} \times \mathcal{M}$ and $(\mathcal{T} \times \mathcal{T}) \cup (\mathcal{M} \times \mathcal{M})$ respectively.

Proof. Denote $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. Set $\mathcal{E}^f = (\mathsf{Gen}^f, \mathsf{Enc}^f, \mathsf{Dec}^f)$ to be the encryption scheme constructed from $\mathcal{E}$ and f in Figure 1.

Our active input-recovery attack is applicable on any $(\mathcal{E}^f, \mathcal{G})$-aided outsourcing protocol $\pi = \langle \mathsf{Clnt}, \mathsf{Srv} \rangle$ as follows.

1. Clnt executes phase 1 of π. That is, it runs $(pk^f, sk^f) \leftarrow \mathsf{Gen}^f(1^\lambda)$ to obtain a public key $pk^f = (pk, \mathsf{Enc}_{pk}(m^*), f(m^*))$, encrypts its input x by computing $\mathbf{c}_x \leftarrow \mathsf{Enc}^f_{pk^f}(x, x)$ and sends $\mathbf{c}_x$ and pk^f to Srv.
2. Upon receiving $\mathbf{c}_x = (\mathbf{c}_1, \mathbf{c}_2)$ and pk^f, Srv generates a new ciphertext $\mathbf{e} = (\mathbf{c}_1, \mathsf{Enc}_{pk}(m^*))$, where $\mathsf{Enc}_{pk}(m^*)$ is taken from pk^f, and sends $(\mathbf{e}, \mathcal{I})$ to Clnt.
3. Clnt sends $(\mathbf{c}'_1, \mathbf{c}'_2) \leftarrow \mathsf{Enc}^f_{pk^f}(\mathcal{I}(\mathsf{Dec}^f_{sk^f}(\mathbf{e})))$ to Srv.
4. Upon receiving the client's response $(\mathbf{c}'_1, \mathbf{c}'_2)$, Srv outputs $\mathbf{c}'_1$.

The attack recovers the client's input x because $\mathbf{c}'_1 = x$ as explained next. Observe that $\mathcal{I}(\mathsf{Dec}^f_{sk^f}(\mathbf{e})) = (x, m^*)$ is a message where the encryption algorithms $\mathsf{Enc}^f_{pk^f}$ is punctured, implying that

$$\mathsf{Enc}^f_{pk^f}(\mathcal{I}(\mathsf{Dec}^f_{sk^f}(\mathbf{e}))) = (x, m^*).$$

Namely, $(\mathbf{c}'_1, \mathbf{c}'_2) = (x, m^*)$ in Step 3, and so $\mathbf{c}'_1 = x$. □

5 CPA Implies Privacy against Semi-Honest Adversaries

We define a natural property for $(\mathcal{E}, \mathcal{G})$-aided outsourcing protocols (called *cleartext computable*), and show that for protocols satisfying this property, CPA-security guarantees privacy against semi-honest servers; See Theorem 13.

Cleartext computable protocols. A protocol is cleartext computable if the messages whose encryption constitutes the client's responses to the server's queries are efficiently computable given only the client's input. To formalize this we first define the client's cleartext response. Let $\pi = \langle \mathsf{Clnt}, \mathsf{Srv} \rangle$ be an $(\mathcal{E}, \mathcal{G})$-aided outsourcing protocol (cf. Definition 5). The client's *cleartext response* in an execution of π on client's input x and randomness r_{Clnt}, server's randomness r_{Srv}, and security parameter $\lambda \in \mathbb{N}$, is defined by:

$$\mathsf{clear\text{-}res}^{\pi}((x, r_{\mathsf{Clnt}}), r_{\mathsf{Srv}}, \lambda) = (G_{n_1}(\mathsf{Dec}_{sk}(\mathbf{e}_1)), \ldots, G_{n_q}(\mathsf{Dec}_{sk}(\mathbf{e}_q)))$$

where $(sk, pk) \leftarrow \mathsf{Gen}(1^{\lambda})$ is the key pair generated by the client in Phase 1 of π; q is the number of queries sent from server to client in Phase 2 of π; and for each $j \in [q]$, $(\mathbf{e_j}, n_j)$ and $\mathsf{Enc}_{pk}(G_{n_j}(\mathsf{Dec}_{sk}(\mathbf{e_j})))$ are the jth server's query and the corresponding client's response respectively with $G_{n_j}(\mathsf{Dec}_{sk}(\mathbf{e_j}))$ being the underlying cleartext response message.

Definition 13 (cleartext computable). *An $(\mathcal{E}, \mathcal{G})$-aided outsourcing protocol $\pi = \langle \mathsf{Clnt}, \mathsf{Srv} \rangle$ for computing a function $F : \mathsf{A} \to \mathsf{B}$ is* cleartext computable *if* Srv *is ppt and there exists a ppt function h such that for all inputs $x \in \mathsf{A}$, all client and server randomness r_{Clnt} and r_{Srv}, respectively, and all $\lambda \in \mathbb{N}$*

$$\mathit{clear\text{-}res}^{\pi}((x, r_{\mathsf{Clnt}}), r_{\mathsf{Srv}}, \lambda) = h(x)$$

CPA-security implies privacy for cleartext computable protocols. We show that for cleartext computable $(\mathcal{E}, \mathcal{G})$-aided outsourcing protocols, CPA-security of $\mathcal{E}$ implies that the protocol preserves privacy against semi-honest servers.

Similarly to Theorem 9, the family $\mathcal{G}$ should be admissible in the sense that all $G_n \in \mathcal{G}$ are polynomial-time computable (in the security parameter) and have fixed output length, i.e., $|G_n(x_0)| = |G_n(x_1)|$ for all $x_0, x_1 \in \mathcal{M}$.

Theorem 13 (privacy of cleartext computable protocols). *Every cleartext computable $(\mathcal{E}, \mathcal{G})$-aided outsourcing protocol preserves privacy against semi-honest servers, provided that $\mathcal{E}$ is CPA-secure and $\mathcal{G}$ is admissible.*

Proof. We show that for cleartext computable protocols, when instantiated with a CPA-secure encryption scheme, a semi-honest server cannot distinguish encrypted response of correct or random values, and hence privacy follows. The formal proof appears in the full version [2].

6 Conclusions

In this work we introduce the notion of funcCPA, which is a strict relaxation of CCA2-security, show it is achievable for HE schemes (unlike CCA2) and sufficient for ensuring privacy against malicious servers for the wide an natural family of client-aided outsourcing protocols (unlike CPA, as we prove). In contrast, against semi-honest adversaries, we prove that CPA-security suffices for ensuring privacy in all cleartext computable client-aided outsourcing protocols.

References

1. A. Akavia, D. Feldman, and H. Shaul. Secure search on encrypted data via multi-ring sketch. In D. Lie, M. Mannan, M. Backes, and X. Wang, editors, *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*, pages 985–1001. ACM, 2018.
2. A. Akavia, C. Gentry, S. Halevi, and M. Vald. Achievable cca2 relaxation for homomorphic encryption. Cryptology ePrint Archive, Paper 2022/282, 2022. `https://eprint.iacr.org/2022/282`.
3. A. Akavia, M. Leibovich, Y. S. Resheff, R. Ron, M. Shahar, and M. Vald. Privacy-preserving decision tree training and prediction against malicious server. Cryptology ePrint Archive, Paper 2019/1282, 2019. `https://eprint.iacr.org/2019/1282`.
4. A. Akavia, M. Leibovich, Y. S. Resheff, R. Ron, M. Shahar, and M. Vald. Privacy-preserving decision trees training and prediction. In F. Hutter, K. Kersting, J. Lijffijt, and I. Valera, editors, *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2020, Ghent, Belgium, September 14-18, 2020, Proceedings, Part I*, volume 12457 of *Lecture Notes in Computer Science*, pages 145–161. Springer, 2020.
5. A. Akavia, M. Leibovich, Y. S. Resheff, R. Ron, M. Shahar, and M. Vald. Privacy-preserving decision trees training and prediction. In *Machine Learning and Knowledge Discovery in Databases*, pages 145–161. Springer International Publishing, 2021.
6. A. Akavia, M. Leibovich, Y. S. Resheff, R. Ron, M. Shahar, and M. Vald. Privacy-preserving decision trees training and prediction. *ACM Trans. Priv. Secur.*, 25(3), may 2022.
7. A. Akavia, H. Shaul, M. Weiss, and Z. Yakhini. Linear-regression on packed encrypted data in the two-server model. In M. Brenner, T. Lepoint, and K. Rohloff, editors, *Proceedings of the 7th ACM Workshop on Encrypted Computing & Applied Homomorphic Cryptography, WAHC@CCS 2019, London, UK, November 11-15, 2019*, pages 21–32. ACM, 2019.
8. A. Akavia and M. Vald. On the privacy of protocols based on cpa-secure homomorphic encryption. Cryptology ePrint Archive, Report 2021/803, 2021. `https://ia.cr/2021/803`.
9. R. Bost, R. A. Popa, S. Tu, and S. Goldwasser. Machine learning classification over encrypted data. In *NDSS*, volume 4324, page 4325, 2015.
10. F. Bourse, R. Del Pino, M. Minelli, and H. Wee. FHE circuit privacy almost for free. In *Advances in Cryptology – CRYPTO 2016*, pages 62–89. Springer Berlin Heidelberg, 2016.

11. Z. Brakerski. Fully homomorphic encryption without modulus switching from classical gapSVP. In *Advances in Cryptology - CRYPTO 2012 - 32nd Annual Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2012. Proceedings*, pages 868–886, 2012.
12. Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*, pages 309–325, 2012.
13. Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. *SIAM Journal on computing*, 43(2):831–871, 2014.
14. R. Canetti, H. Krawczyk, and J. B. Nielsen. Relaxing chosen-ciphertext security. In D. Boneh, editor, *Advances in Cryptology - CRYPTO 2003*, pages 565–582, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.
15. R. Canetti, S. Raghuraman, S. Richelson, and V. Vaikuntanathan. Chosen-ciphertext secure fully homomorphic encryption. In S. Fehr, editor, *Public-Key Cryptography – PKC 2017*, pages 213–240, Berlin, Heidelberg, 2017. Springer Berlin Heidelberg.
16. D. Cash, M. Green, and S. Hohenberger. New definitions and separations for circular security. In *International Workshop on Public Key Cryptography*, pages 540–557. Springer, 2012.
17. J. H. Cheon, A. Kim, M. Kim, and Y. Song. Homomorphic encryption for arithmetic of approximate numbers. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 409–437. Springer, 2017.
18. I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. TFHE: Fast fully homomorphic encryption over the torus. *Journal of Cryptology*, 33:34–91, 2019.
19. W. Chongchitmate and R. Ostrovsky. Circuit-private multi-key FHE. In *20th IACR International Conference on Public-Key Cryptography – PKC 2017*, pages 24–270. Springer Berlin Heidelberg, 2017.
20. L. Ducas and D. Micciancio. FHEW: Bootstrapping homomorphic encryption in less than a second. In *Advances in Cryptology – EUROCRYPT 2015*, pages 617–640. Springer Berlin Heidelberg, 2015.
21. L. Ducas and D. Stehlé. Sanitization of FHE ciphertexts. In *Advances in Cryptology – EUROCRYPT 2016*, pages 294–310. Springer Berlin Heidelberg, 2016.
22. J. Fan and F. Vercauteren. Somewhat practical fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2012:144, 2012.
23. C. Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. `crypto.stanford.edu/craig`.
24. C. Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, STOC '09, pages 169–178. Association for Computing Machinery, 2009.
25. C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Annual Cryptology Conference*, pages 75–92. Springer, 2013.
26. I. Giacomelli, S. Jha, M. Joye, C. D. Page, and K. Yoon. Privacy-preserving ridge regression with only linearly-homomorphic encryption. In *Applied Cryptography and Network Security - 16th International Conference, ACNS 2018*, pages 243–261. Springer, 2018.
27. O. Goldreich. *The Foundations of Cryptography - Volume 1, Basic Techniques*. Cambridge University Press, 2001.
28. C. Hazay and Y. Lindell. *Efficient Secure Two-Party Protocols: Techniques and Constructions*. Springer-Verlag, Berlin, Heidelberg, 1st edition, 2010.

29. Y. Ishai and A. Paskin. Evaluating branching programs on encrypted data. In *Theory of Cryptography, 4th Theory of Cryptography Conference, TCC 2007*, pages 575–594. Springer, 2007.
30. C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan. Gazelle: A low latency framework for secure neural network inference. In *Proceedings of the 27th USENIX Conference on Security Symposium*, SEC'18, page 1651–1668. USENIX Association, 2018.
31. J. Katz and Y. Lindell. *Introduction to Modern Cryptography (Chapman & Hall/Crc Cryptography and Network Security Series)*. Chapman & Hall/CRC, 2007.
32. J. Lai, R. H. Deng, C. Ma, K. Sakurai, and J. Weng. Cca-secure keyed-fully homomorphic encryption. In C.-M. Cheng, K.-M. Chung, G. Persiano, and B.-Y. Yang, editors, *Public-Key Cryptography – PKC 2016*, pages 70–98, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.
33. B. Li and D. Micciancio. On the security of homomorphic encryption on approximate numbers. *IACR Cryptology ePrint Archive*, 2020:1533, 2020.
34. J. Loftus, A. May, N. P. Smart, and F. Vercauteren. On cca-secure somewhat homomorphic encryption. In A. Miri and S. Vaudenay, editors, *Selected Areas in Cryptography*, pages 55–72, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
35. G. Malavolta. Circuit privacy for quantum fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2020:1454, 2020.
36. K. Nuida. How to handle invalid queries for malicious-private protocols based on homomorphic encryption. In *Proceedings of the 9th ACM on ASIA Public-Key Cryptography Workshop*, APKC '22, page 15–25, New York, NY, USA, 2022. Association for Computing Machinery.
37. R. Ostrovsky, A. Paskin-Cherniavsky, and B. Paskin-Cherniavsky. Maliciously circuit-private FHE. In *Advances in Cryptology – CRYPTO 2014*, pages 536–553, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.
38. C. Peikert. A decade of lattice cryptography. *Found. Trends Theor. Comput. Sci.*, 10(4):283–424, mar 2016.
39. M. Prabhakaran and M. Rosulek. Homomorphic encryption with cca security. In L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, and I. Walukiewicz, editors, *Automata, Languages and Programming*, pages 667–678, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
40. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), Sept. 2009.
41. M. Rosulek. The joy of cryptography. `https://joyofcryptography.com`.
42. V. Shoup. A proposal for an ISO standard for public key encryption. *IACR Cryptol. ePrint Arch.*, page 112, 2001.
43. W. Wang, Y. Jiang, Q. Shen, W. Huang, H. Chen, S. Wang, X. Wang, H. Tang, K. Chen, K. E. Lauter, and D. Lin. Toward scalable fully homomorphic encryption through light trusted computing assistance. *CoRR*, abs/1905.07766, 2019.

A Proof of Lemma 2.

We prove Lemma 2 showing that for every fully decryptable HE scheme $\mathcal{E}$ that has a sanitization algorithm $\mathsf{Sanitize}$, if its sanitized version $\mathcal{E}^{\mathsf{santz}}$ is $\mathcal{C}$-homomorphic, then it is circuit-private$^+$ for $\mathcal{C}$.

Proof (of Lemma 2). Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be a fully decryptable HE scheme with a sanitization algorithm Sanitize. Denote by $\mathcal{E}^{\mathsf{santz}} = (\mathsf{Gen}, \mathsf{Enc}^{\mathsf{santz}}, \mathsf{Dec}, \mathsf{Eval}^{\mathsf{santz}})$ its sanitized version as specified in Definition 7. Let $\mathcal{C}$ be the set of circuits so that $\mathcal{E}^{\mathsf{santz}}$ is $\mathcal{C}$-homomorphic. We show that $\mathcal{E}^{\mathsf{santz}}$ is circuit-private$^+$ for $\mathcal{C}$.

Fix a circuit $C \in \mathcal{C}$ over ℓ inputs, ciphertexts $c_1, \ldots, c_\ell$, a security parameter λ. To prove circuit-privacy$^+$ holds we need to show the two ciphertexts $\mathsf{Enc}^{\mathsf{santz}}_{pk}\left(C\left(\mathsf{Dec}_{sk}(c_1), \cdots, \mathsf{Dec}_{sk}(c_\ell)\right)\right)$ and $\mathsf{Eval}^{\mathsf{santz}}_{pk}\left(C, c_1, \ldots, c_\ell\right)$ are statistically close, with overwhelming probability over the choice of $(pk, sk) \leftarrow \mathsf{Gen}(\lambda)$.

By definition of $\mathcal{E}^{\mathsf{santz}}$,

$$\begin{aligned} &\mathsf{Enc}^{\mathsf{santz}}_{pk}\left(C\left(\mathsf{Dec}_{sk}(c_1), \cdots, \mathsf{Dec}_{sk}(c_\ell)\right)\right) \\ &= \mathsf{Sanitize}_{pk}\left(\mathsf{Enc}_{pk}\left(C\left(\mathsf{Dec}_{sk}(c_1), \ldots, \mathsf{Dec}_{sk}(c_\ell)\right)\right)\right) \end{aligned} \tag{10}$$

and

$$\begin{aligned} &\mathsf{Eval}^{\mathsf{santz}}_{pk}\left(C, c_1, \ldots, c_\ell\right) \\ &= \mathsf{Sanitize}_{pk}\left(\mathsf{Eval}_{pk}\left(C, \mathsf{Sanitize}_{pk}(c_1), \ldots, \mathsf{Sanitize}_{pk}(c_\ell)\right)\right) \end{aligned} \tag{11}$$

By the sanitization property of Sanitize, if two ciphertexts decrypt to the same plaintext then their sanitized version is statistically close. Therefore it is sufficient to show that the corresponding ciphertexts in the above two equations (i.e., $\mathsf{Enc}_{pk}\left(C\left(\mathsf{Dec}_{sk}(c_1), \ldots, \mathsf{Dec}_{sk}(c_\ell)\right)\right)$ and $\mathsf{Eval}_{pk}(C, \mathsf{Sanitize}_{pk}(c_1), \ldots, \mathsf{Sanitize}_{pk}(c_\ell))$ decrypt to the same plaintext.

The correctness property of $\mathcal{E}$ together with it being fully decryptable ensures that for every $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$:

$$\forall i \in [\ell] : \Pr[\mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(\mathsf{Dec}_{sk}(c_i))) = \mathsf{Dec}_{sk}(c_i)] \geq 1 - \mathsf{neg}(\lambda) \tag{12}$$

and

$$\Pr\left[\begin{smallmatrix}\mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(C(\mathsf{Dec}_{sk}(c_1),\ldots,\mathsf{Dec}_{sk}(c_\ell))))\\ =C(\mathsf{Dec}_{sk}(c_1),\ldots,\mathsf{Dec}_{sk}(c_\ell))\end{smallmatrix}\right] \geq 1 - \mathsf{neg}(\lambda) \tag{13}$$

where the probabilities are taken over the random coins of the encryption algorithm.

From Equation 12 together with the sanitization property of Sanitize, we obtain that, for each $i \in [\ell]$, with probability $\geq 1 - \mathsf{neg}(\lambda)$ over the choice of $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$:

$$\Delta\left(\left(\mathsf{Sanitize}_{pk}(\mathsf{Enc}_{pk}(\mathsf{Dec}_{sk}(c_i))), (pk, sk)\right), \left(\mathsf{Sanitize}_{pk}(c_i), (pk, sk)\right)\right) \leq \mathsf{neg}(\lambda)$$

Moreover, with probability $\geq 1 - \mathsf{neg}(\lambda)$, the above holds for all $i \in [\ell]$ simultaneously (by union bound).

Since Sanitize uses independent randomness for each $i \in [\ell]$, its output on distinct i's is statistically independent. So the joint distribution over all $i \in [\ell]$

is likewise negligible (since the statistical distance of the joint distribution of independent random variables is the sum of their statistical distances, and the number of random variables is $\ell = \mathsf{poly}(\lambda)$). Namely,

$$\Delta\left(\begin{smallmatrix}(\mathsf{Sanitize}_{pk}(\mathsf{Enc}_{pk}(\mathsf{Dec}_{sk}(c_1))),\ldots,\mathsf{Sanitize}_{pk}(\mathsf{Enc}_{pk}(\mathsf{Dec}_{sk}(c_\ell))),(pk,sk)),\\ (\mathsf{Sanitize}_{pk}(c_1),\ldots,\mathsf{Sanitize}_{pk}(c_\ell),(pk,sk))\end{smallmatrix}\right) \leq \mathsf{neg}(\lambda) \quad (14)$$

The $\mathcal{C}$-homomorphism of $\mathcal{E}^{\mathsf{santz}}$ guarantees that $\mathcal{E}^* = (\mathsf{Gen}, \mathsf{Enc}^{\mathsf{santz}}, \mathsf{Dec}, \mathsf{Eval})$ is likewise $\mathcal{C}$-homomorphic (due to the message-preservation property of $\mathsf{Sanitize}$), and hence for every $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$ it holds that,

$$\Pr\left[\begin{smallmatrix}\mathsf{Dec}_{sk}(\mathsf{Eval}_{pk}(C,\mathsf{Sanitize}_{pk}(\mathsf{Enc}_{pk}(\mathsf{Dec}_{sk}(c_1))),\ldots,\mathsf{Sanitize}_{pk}(\mathsf{Enc}_{pk}(\mathsf{Dec}_{sk}(c_\ell))))))\\ =C(\mathsf{Dec}_{sk}(c_1),\ldots,\mathsf{Dec}_{sk}(c_\ell))\end{smallmatrix}\right] \geq 1 - \mathsf{neg}(\lambda) \quad (15)$$

Combining Equations 14-15 we guarantee correctness of Eval on the sanitized $c_1, \ldots, c_\ell$. That is, for every $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$ it holds that,

$$\Pr\left[\begin{smallmatrix}\mathsf{Dec}_{sk}(\mathsf{Eval}_{pk}(C,\mathsf{Sanitize}_{pk}(c_1),\ldots,\mathsf{Sanitize}_{pk}(c_\ell)))\\ =C(\mathsf{Dec}_{sk}(c_1),\ldots,\mathsf{Dec}_{sk}(c_\ell))\end{smallmatrix}\right] \geq 1 - \mathsf{neg}(\lambda)$$

Using the correctness property of $\mathcal{E}$ as stated in Equation 13 we obtain that for every $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$ it holds that with probability $\geq 1 - \mathsf{neg}(\lambda)$ over the random coins of the experiment,

$$\mathsf{Dec}_{sk}\left(\mathsf{Eval}_{pk}\left(C, \mathsf{Sanitize}_{pk}(c_1), \ldots, \mathsf{Sanitize}_{pk}(c_\ell)\right)\right)$$

$$= \mathsf{Dec}_{sk}\left(\mathsf{Enc}_{pk}\left(C\left(\mathsf{Dec}_{sk}(c_1), \ldots, \mathsf{Dec}_{sk}(c_\ell)\right)\right)\right)$$

This concludes the proof as by the sanitization property of $\mathsf{Sanitize}$, we obtain that with probability $\geq 1 - \mathsf{neg}(\lambda)$ over the choice of $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$ and the random coins in Enc and Eval the following distributions are statistically close,

$$\mathsf{Sanitize}_{pk}\left(\mathsf{Enc}_{pk}\left(C\left(\mathsf{Dec}_{sk}(c_1), \ldots, \mathsf{Dec}_{sk}(c_\ell)\right)\right)\right)$$

and

$$\mathsf{Sanitize}_{pk}\left(\mathsf{Eval}_{pk}\left(C, \mathsf{Sanitize}_{pk}(c_1), \ldots, \mathsf{Sanitize}_{pk}(c_\ell)\right)\right)$$

as desired. □