Achievable CCA2 Relaxation for Homomorphic Encryption*

Adi Akavia[1] **, Craig Gentry[2], Shai Halevi[2], and Margarita Vald[3]

[1] University of Haifa, Israel
adi.akavia@gmail.com
[2] Algorand Foundation, USA
craigbgentry@gmail.com,shaih@alum.mit.edu
[3] 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).

In summary, 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

Background. Homomorphic encryption (HE) supports computing over encrypted data without access to the secret key. HE is a prominent approach

* A preliminary version of this work appeared in IACR Cryptology ePrint Archive Report 2021/803 [4].

** The first author was supported in part by the Israel Science Foundation grant 3380/19, and by the Israel National Cyber Directorate via the Haifa and BIU Cyber Centers.

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 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. [36], Chapter 10). Therefore, relying on CPA-security typically secures protocols only against semi-honest adversaries e.g. in [38,5,1,22,3,26,2] (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?

1.1 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 (assuming one-way functions exist). 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 [20], Brakerski [8] and Ducas and Micciancio [16] (where sanitization is as defined in [17], 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 [38] 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 [5,1,2,22,3,26] 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 [5], minima [1,2], linear equations solutions [22,3], ReLU and Max-Pooling [26].

To capture and generalize secure outsourcing protocols such as discussed above [5,1,2,22,3,26], 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, to lessen some of the computational burden, 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. Namely, funcCPA suffices in these settings for ensuring privacy, even against malicious attackers.

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 [9], BGV [8] and B/FV[7,18] 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.[4] Our security proof requires essentially no modifications to the schemes (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 12).

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

[4] 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.

open problem on the necessity of circular security assumption (see e.g. Question 11 in Peikert's survey [33]).

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 for security 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 instantiated with a CPA-secure encryption scheme preserve privacy against semi-honest servers.*

In summary, in this work we introduce the notion of funcCPA that lies strictly between CPA and CCA2 (under standard assumptions), and show that funcCPA-security is achievable for HE schemes (unlike CCA2) and sufficient for ensuring privacy against malicious servers for the wide and natural family of client-aided outsourcing protocols (unlike CPA). To the best of our knowledge funcCPA-security is the first relaxation of CCA2 that is both achievable for HE schemes and ensures privacy (for client-aided outsourcing protocols) against malicious adversaries.

1.2 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. 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.

1.3 Related Work

CCA2 relaxations. Several relaxations of CCA2-security have been previously considered [37,10,34,30,28], albeit relaxing other aspects of CCA2 than addressed by our funcCPA notion.

Concretely, Shoup [37], Canetti et al [10] and Prabhakaran and Rosulek [34] proposed a relaxation of CCA2 where forbidden decryption oracle queries include, not only the challenge ciphertext, but any ciphertext that decrypts to the same message as the challenge ciphertext (or extensions of this notion). This captures encryption schemes where ciphertexts are malleable but only in ways that preserve their underlying plaintext (or its coset). These relaxations are motivated by capturing encryption schemes (some of which come up naturally in practice) that are not CCA2 secure but seem sufficiently secure "for most practical purposes"; the intuition is that the ability to generate different ciphertexts that decrypt to the same value as a given ciphertext should not help the attacker. In the context of HE however their relaxation is unachievable, because if homomorphism is supported, then an adversary receiving a challenge ciphertext encrypting a message x can homomorphically produce a ciphertext for a related message (e.g. $x+1$ or $2x$) and by calling the decrypting oracle on this ciphertext the adversary can recover x.

Another line of work, including e.g. [30,28] shows that CCA1 is achievable for HE (unlike CCA2). This 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 unachievable for fully homomorphic schemes (because fully homomorphic schemes provide an encryption of the secret key as part of the public key, for the purpose of bootstrapping, and querying the CCA1 oracle on this ciphertext would recover the secret key and break security). In contrast, our results show that funcCPA-security is achievable for fully homomorphic schemes (e.g., see Theorem 1).

Insufficiency of CPA-security. The insufficiency of CPA-security for protocols utilizing homomorphic encryption was considered by Li and Micciancio [29]. They show that protocols instantiated with the CPA-secure approximate HE schemes of CKKS [12] 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). Moreover, we show that CPA-security does suffice to guarantee privacy against semi-honest adversaries for cleartext computable client-aided outsourcing protocols.

Paper organization. Preliminary definitions are given in Section 2; our results for malicious adversaries –including the funcCPA definition, achievability from sanitization and sufficiency, as well as the insufficiency of CPA– appear in Section 3; our results on the achievability of funcCPA-security by existing HE scheme are given in Section 4; and our result for semi-honest adversaries in Section 5; we conclude in Section 6.

2 Preliminaries

In this section we briefly specify standard terminology, notations and definitions used throughout this paper, including CPA-security, homomorphic encryption, sanitization algorithm and privacy-preserving protocols. See further details on standard definitions in Appendix A.

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 [23]) 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*. See the details in Appendix A.1.

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. See the details in Appendix A.2.

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. The formal definition follows (adapted from [8]).

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 quadruple 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* $\widehat{c}$*; denoted:* $\widehat{c} \leftarrow \mathsf{Eval}_{pk}(C, c_1, \ldots, c_\ell)$*.*

The scheme $\mathcal{E}$ *is called* secure *if it is a CPA-secure PKE;* compact *if its decryption circuit is of polynomial size;* $\mathcal{C}$-homomorphic *for a circuit family* $\mathcal{C}$ *if for all* $C \in \mathcal{C}$ *and for 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 polynomially computable 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 Definitions 4.1.2-4.1.3 in [19].

Definition 2 (leveled HE [20]). *A family of homomorphic encryption schemes* $\{\mathcal{E}^{(L)}: L \in \mathbb{Z}^+\}$ *is* leveled fully homomorphic (leveled HE) *if, for all* $L \in \mathbb{Z}^+$*, they all use the same decryption circuit,* $\mathcal{E}^{(L)}$ *compactly evaluates all circuits of depth at most* L *(that use some specified set of gates), and the computational complexity of* $\mathcal{E}^{(L)}$*'s algorithms is polynomial in* λ*,* L*, and (in the case of* Eval*) the size of the circuit* C*. In this case* L *can be given as an extra parameter to* Gen*, denoted* $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda, 1^L)$*.*

Remark 1. In Definition 1 the syntax denotes by pk the key used both in Enc and Eval. When it is desired to explicitly specify what information is needed by each of these two procedures, it is customary to slightly change this syntax so that key generation outputs three keys: $(pk, evk, sk) \leftarrow \mathsf{Gen}(1^\lambda, 1^L)$, where Enc takes the public key pk and Eval takes the evaluation key evk (Dec takes the secret key sk).

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, as shown by Ducas and Stehlé [17], essentially for all the major schemes known at the time their paper was published, including Gentry's original scheme [20], BGV [8], and FHEW [16].[5]

Definition 3 (Sanitization algorithm [17]). *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.* $\mathsf{Dec}_{sk}(c) = \mathsf{Dec}_{sk}(c')$*:*

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

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

[5] We conjecture that [17] can be extended to newer schemes, published following their paper, including TFHE [14] and CKKS [13]; this is beyond the scope of this work.

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 (see [24] Definition 2.6.2 Part 2):[6]

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* $x \in \mathsf{A}$,

$$\Pr[\mathit{out}^{\pi}_{\mathsf{Clnt}}(x, \perp, \lambda) = F(x)] > 1 - \mathsf{neg}(\lambda).$$

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:*

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

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 $\mathcal{G} = \{G_n \colon \mathcal{M} \to \mathcal{M}\}_{n \in \mathbb{N}}$. We note that $\mathcal{E}$ can be any PKE scheme (i.e., not necessarily an HE scheme).

[6] We note that the server has no input and no output, and hence we do not require security against the client.

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* $\mathcal{G} = \{G_n \colon \mathcal{M} \to \mathcal{M}\}_{n\in\mathbb{N}}$ *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 ciphertexts and* $n \in \mathbb{N}$*, and 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 2 (multiple inputs and outputs). The family $\mathcal{G}$ may include functions with multiple inputs and outputs. In this case the query $\mathbf{e}$ and response $\mathbf{e}'$ are vectors of ciphertexts, and the decryption and encryption in $\mathsf{Enc}_{pk}(G_n(\mathsf{Dec}_{sk}(\mathbf{e})))$ are 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 (Section 3.1, Definition 6), show that funcCPA-secure HE is achievable from any HE equipped with a sanitization algorithm (Section 3.2, Theorem 7) and prove that every client-aided protocols (see Definition 5 in Section 2) instantiated with a funcCPA-secure scheme preserves privacy against malicious adversaries (Section 3.3, Theorem 8). Furthermore, we prove that CPA-security does not suffice to guarantee privacy in client-aided outsourcing protocols by presenting an input-recovery attack on such protocols when instantiated with a (carefully crafted) CPA-secure scheme (Section 3.4, Theorem 10). The results presented in this section together with the definition of the funcCPA-experiment imply that funcCPA-security is strictly between CCA2 and CPA (under standard assumptions).

3.1 funcCPA-Security: A Relaxation of CCA2

We define the *function-chosen-plaintext attack* (funcCPA-security) security notion of public-key encryption. 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: the 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 *with respect to 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.*

We observe that for "fully decryptable" $\mathcal{C}$-homomorphic schemes it suffices to prove $\mathsf{funcCPA}$-security w.r.t the identity function $\mathcal{I}$ to obtain $\mathsf{funcCPA}$-security w.r.t $\mathcal{C}$. 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). We note that full decryption holds for well-known schemes including [35,8,7,21,16,14].

Definition 7 (fully decryptable). *A PKE scheme $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ and ciphertext space $\mathcal{T}$ is* fully decryptable *if $\mathcal{T}$ is efficiently recognizable and for all $\lambda \in \mathbb{N}$, $c \in \mathcal{T}$, and any (pk, sk) in the range of $\mathsf{Gen}(1^\lambda)$ it holds that $\mathsf{Dec}_{sk}(c) \in \mathcal{M}$.*

Lemma 1. *Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be a fully decryptable $\mathcal{C}$-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 fully decryptable $\mathcal{C}$-homomorphic encryption scheme with message space $\mathcal{M}$ and ciphertext $\mathcal{T}$ that is $\mathsf{funcCPA}$-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 first evaluated using Eval and then forwarded to the challenger that computes $\mathsf{Enc}_{pk}(\mathcal{I}(\mathsf{Dec}_{sk}(\cdot)))$. 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 computing $\mathbf{e}' \leftarrow \mathsf{Eval}_{pk}(C_n, \mathbf{e})$ and sending $(\mathbf{e}', \mathcal{I})$ to the challenger. The response is given to $\mathcal{A}$.
- 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 $\mathcal{B}$ due to the full decryption property of $\mathcal{E}$ together with the $\mathcal{C}$-homomorphism. 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:

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

and 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 CPA-secure HE scheme $\mathcal{E}$ that has a sanitization algorithm (e.g. [20,8,16]) into a sanitized HE scheme $\mathcal{E}^{\mathsf{santz}}$ that is funcCPA-secure. See the construction of $\mathcal{E}^{\mathsf{santz}}$ in Definition 8, and the proof it is funcCPA-secure in Theorem 7.

Definition 8 (Sanitized scheme $\mathcal{E}^{\mathsf{santz}}$). *Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be a $\mathcal{C}$-homomorphic PKE scheme with message space $\mathcal{M}$ and a sanitization algorithm Sanitize. We define the sanitized scheme, denoted $\mathcal{E}^{\mathsf{santz}} = (\mathsf{Gen}, \mathsf{Enc}^{\mathsf{santz}}, \mathsf{Dec}, \mathsf{Eval}^{\mathsf{santz}})$, 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 a public key* pk*, a circuit* $C \in \mathcal{C}$*, 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 all the properties of $\mathcal{E}$: $\mathcal{C}$-homomorphism, compactness, security, and correctness. In particular, correctness holds due to correctness of $\mathcal{E}$ and the message-preservation property of Sanitize. We show that if $\mathcal{E}$ is CPA-secure, then $\mathcal{E}^{\mathsf{santz}}$ in funcCPA-secure.

Theorem 7 ($\mathcal{E}^{\mathsf{santz}}$ is funcCPA-secure). *If $\mathcal{E}$ is a $\mathcal{C}$-homomorphic CPA-secure PKE scheme with a sanitization algorithm, then the sanitized scheme $\mathcal{E}^{\mathsf{santz}}$ is funcCPA-secure w.r.t. $\mathcal{C}$.*[7]

Proof. To prove the theorem we first enhance the definition of circuit privacy to *circuit-privacy*$^+$ (cf. Definition 9 below); then show that if $\mathcal{E}$ is $\mathcal{C}$-homomorphic and has a sanitization algorithm then the sanitized scheme $\mathcal{E}^{\mathsf{santz}}$ is circuit-privacy$^+$ for $\mathcal{C}$ (cf. Lemma 2 below); and show that if a $\mathcal{C}$-homomorphic CPA-secure encryption scheme is 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. $\mathcal{C}$. □

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

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 [25,20,6], or considered settings where both keys and ciphertexts may be maliciously formed [25,32,15,31]. In contrast, in our settings the keys are properly formed whereas the ciphertexts may be maliciously formed.

Definition 9 (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 $\mathcal{C}$-homomorphic PKE with a sanitization algorithm, then $\mathcal{E}^{\mathsf{santz}}$ is circuit-private$^+$ for $\mathcal{C}$.*

Proof. Informally, the proof follows from the definition of $\mathcal{E}^{\mathsf{santz}}$ and the properties of $\mathcal{C}$-homomorphism and Sanitize; See Appendix B.1. □

circuit-privacy$^+$ *implies funcCPA.* We prove that a sufficient condition for a HE scheme to be funcCPA-secure is that it is CPA-secure and circuit-private$^+$. We remark that Lemma 3 holds even if the schemes satisfies only a weaker notion of circuit-privacy$^+$ where we require only computational indistinguishability rather than statistical.

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. See the formal proof in Appendix B.2. □

3.3 funcCPA Implies Privacy against Malicious Adversaries

We show that $(\mathcal{E}, \mathcal{G})$-aided outsourcing protocols preserve privacy against malicious servers, if $\mathcal{E}$ is funcCPA-secure. This implication holds for any funcCPA-secure PKE, not only HE schemes.

Theorem 8 (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. The proof relies on the fact that any communication with the client, specified by the protocol, can be replaced by communication with the $\mathsf{Enc}_{pk}(\mathcal{G}(\mathsf{Dec}_{sk}(\cdot)))$ oracle. See the formal proof in Appendix B.3. □

3.4 CPA does not Imply Privacy 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 $\mathsf{Enc}_{pk}(m^*)$.[8] See our construction in Figure 1 and Theorem 9. Our attack breaks security in the strong sense that the server is able to completely recover the client's input; See Theorem 10.

Theorem 9 (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*

[8] 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.

$\underline{\mathsf{Gen}^f(1^\lambda)}$: 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.

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.[9]

Proof. Correctness, compactness and homomorphism of $\mathcal{E}^f$ follow directly from the properties of $\mathcal{E}$. The CPA-security of $\mathcal{E}^f$ essentially follows from the fact that 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 $f(m^*)$. The latter however 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 Lemma 5-6, Appendix B.4. □

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 10 (CPA-security does not imply privacy). *For every PKE scheme $\mathcal{E}$ with message-space $\mathcal{M}$ and every one-way function f over*

[9] 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 $(C, C) \in \mathcal{C} \times \mathcal{C}$.

$\mathcal{M}$, 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.

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 $\mathbf{c}'_1 = x$. □

4 On the Achievability of funcCPA for Existing Schemes

In this section we present our results on the achievability of funcCPA-security for existing HE schemes. On the positive side, we prove that funcCPA-security is satisfied by all leveled schemes with independent level keys (see Definition 12), e.g., the leveled HE schemes of BV [9], BGV [8] and B/FV [7,18]; See Section 4.1. Conversely, we show that funcCPA-security for homomorphic schemes supporting oblivious secret key extraction (see Definition 14), e.g., the schemes of BV [9] and BGV [8], implies weak circular security; See Section 4.2.

4.1 funcCPA Security of leveled HE Schemes

In this section we prove that funcCPA-security holds for natural leveled HE schemes (leveled HE) such as BV [9], BGV [8] and B/FV [7,18] (provided

they are CPA-secure). To prove this, we first reformulate the definitions of CPA and funcCPA to capture security for leveled HE schemes (leveled-funcCPA). Next, we show that CPA implies funcCPA for leveled HE schemes satisfying a natural property we call *independent level keys*, and conclude that BV, BGV and B/FV (with a slight modification of their evaluation key generation) are leveled-funcCPA-secure.

Security Definitions for leveled HE. We address leveled HE schemes with the common structure of having each level associated with a set of keys (usually, public, secret and evaluation keys), and each ciphertext associated with a (efficiently recognizable) level corresponding to the keys used for this ciphertext. In these schemes, it is suffices to use the appropriate level keys to encrypt, decrypt, and evaluate. This holds, for example, in BV [9], BGV [8], B/FV [7,18] and CKKS [13].

The definition of CPA-security for leveled HE is similar to the standard CPA definition and the only difference between the two is the capability of the adversary to choose the level to which the challenge ciphertext is encrypted, see Definition 10. This guarantees security of the scheme for all the levels and not only for a specific one. 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 10 (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.

Now we define funcCPA for leveled HE denoted *leveled-funcCPA*. Here, in addition to the level of challenge ciphertext, 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 11. More formally,

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 $\mathcal{A}$ as follows:

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 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 $\mathbf{e}' \leftarrow \mathsf{Enc}_{pk_{\ell-1}}(G_n(\mathsf{Dec}_{sk_\ell}(\mathbf{e}_\ell)))$.[10]
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 11 (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* $\mathcal{G} = \{G_n : \mathcal{M} \to \mathcal{M}\}_{n\in\mathbb{N}}$ *(*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.

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

CPA implies funcCPA for leveled HE. In this section we define a natural property of leveled HE scheme we call *independent level keys* and show that funcCPA-security holds for schemes satisfying this property (provided they are CPA-secure). Informally, we say that a leveled HE scheme has independent level keys if 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. See Definition 12.

Definition 12 (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* GenEvKey*, and outputs for each a public key, secret key, and an evaluation key as follows:*

$$(pk_\ell, sk_\ell) \leftarrow \mathsf{GenKey}(1^\lambda) \text{ for all } \ell \in \{0, \ldots, L\}$$

and

$$evk_\ell \leftarrow \mathsf{GenEvKey}(sk_\ell, pk_{\ell-1}) \text{ for all } \ell \in \{1, \ldots, L\}$$

denoted: $(pk_\ell, evk_\ell, sk_\ell,)_{\ell \in [L]} \leftarrow \mathsf{Gen}(1^\lambda, 1^L)$

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.[11] With this slight modification indeed these scheme satisfy Definition 12.

Proposition 1. *The leveled HE schemes of BV, BGV and B/FV [9,8,7,18] (with the aforementioned evaluation key) have independent level keys.*

We now prove that CPA-secure leveled HE schemes with independent level keys are funcCPA-secure w.r.t any *admissible* family $\mathcal{G}$. 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 $x_0, x_1 \in \mathcal{M}$. We note that the latter trivially holds when $\mathcal{G}$ is specified as a family of circuits.

Theorem 11 (leveled HE is funcCPA). *Let $\mathcal{E}$ be a leveled HE scheme with independent level keys. If $\mathcal{E}$ is CPA-secure (cf. Definition 10), then $\mathcal{E}$ is leveled-funcCPA-secure w.r.t. any admissible family $\mathcal{G}$ (cf. Definition 11).*

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:

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

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 $\mathcal{A}$.

[11] 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.

– 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,

$$\Pr[\mathsf{EXP}^{Fcpa^{\#}}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda, L) = 1] = \Pr[\mathsf{EXP}^{cpa}_{\mathcal{B},\mathcal{E}}(\lambda, L) = 1] \tag{2}$$

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,

$$\begin{aligned} &|\Pr[\mathsf{EXP}^{Fcpa^{\#}}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda, L) = 1] \\ &- \Pr[\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda, L) = 1]| \leq \mathsf{neg}(\lambda)\ . \end{aligned} \tag{3}$$

Putting Equation 3 together with Equations 1-2 it follows that

$$\Pr[\mathsf{EXP}^{cpa}_{\mathcal{B},\mathcal{E}}(\lambda, L) = 1] \geq \frac{1}{2} + \frac{1}{p(\lambda)} - \mathsf{neg}(\lambda). \tag{4}$$

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 11, 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 11, 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 $\mathcal{G} = \{G_n \colon \mathcal{M} \to \mathcal{M}\}_{n\in\mathbb{N}}$ 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 is given in Appendix C.

Corollary 1. *The leveled HE schemes of BV, BGV and B/FV [9,8,7,18] (with the aforementioned evaluation key) are leveled-funcCPA-secure.*

4.2 Barriers on Proving funcCPA for Existing HE Schemes

In this section we prove that if the homomorphic encryption scheme of BV [9] or BGV [8] 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.

Circular security extends CPA-security to capture security of public key encryption schemes against adversaries seeing an encryption of the secret key (see [11], Definition 2.5). This is required by all currently known fully homomorphic encryption schemes, as they publish an encryption of the secret key to be used during bootstrapping (where bootstrapping [19] 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 Definition 3.8 in [9]).

This 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 13 (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 14 (oblivious secret key extraction (ObvSK)). *Let* $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *be a PKE scheme with message space* $\mathcal{M}$*, and* $\mathcal{F} = \{F_n \colon \mathcal{M} \to \mathcal{M}\}_{n\in\mathbb{N}}$ *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* $\lambda \in \mathbb{N}$ *and* (pk, sk) *in the range of* $\mathsf{Gen}(1^\lambda)$*,*

$$\Pr\left[\begin{smallmatrix}(c_1,\ldots,c_n)\leftarrow \mathit{Alg}(pk)\\ F_n(\mathsf{Dec}_{sk}(c_1),\ldots,\mathsf{Dec}_{sk}(c_n))=\mathsf{sk}\end{smallmatrix}\right] \geq 1 - \mathsf{neg}(\lambda) \tag{5}$$

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 12. *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 appear in Appendix C.2.

As a corollary from Theorem 12 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 [19], bootstrappable HE schemes that are weakly circular secure are fully homomorphic. Second, by Theorem 12, 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 appear in Appendix C.3.

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 [9] and BGV [8] 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 [9] or BGV [8] is* funcCPA*-secure w.r.t to* $\mathcal{G}$ *containing* $\mathcal{F}^{GLWE}$*, then it is fully homomorphic.*

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 protocols 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 15 (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 11, 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 Appendix D.

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, 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.
3. 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.
4. 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`.
5. R. Bost, R. A. Popa, S. Tu, and S. Goldwasser. Machine learning classification over encrypted data. In *NDSS*, volume 4324, page 4325, 2015.
6. 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.
7. 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.
8. 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.
9. Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. *SIAM Journal on computing*, 43(2):831–871, 2014.
10. 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.
11. 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.
12. 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.
13. J. H. Cheon, A. Kim, M. Kim, and Y. S. Song. Homomorphic encryption for arithmetic of approximate numbers. In *Advances in Cryptology - ASIACRYPT 2017 - 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I*, pages 409–437, 2017.
14. I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. TFHE: Fast fully homomorphic encryption over the torus. *Journal of Cryptology*, 33:34–91, 2019.
15. 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.
16. 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.
17. L. Ducas and D. Stehlé. Sanitization of FHE ciphertexts. In *Advances in Cryptology – EUROCRYPT 2016*, pages 294–310. Springer Berlin Heidelberg, 2016.

18. J. Fan and F. Vercauteren. Somewhat practical fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2012:144, 2012.
19. C. Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. `crypto.stanford.edu/craig`.
20. 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.
21. 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.
22. 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.
23. O. Goldreich. *The Foundations of Cryptography - Volume 1, Basic Techniques*. Cambridge University Press, 2001.
24. C. Hazay and Y. Lindell. *Efficient Secure Two-Party Protocols: Techniques and Constructions*. Springer-Verlag, Berlin, Heidelberg, 1st edition, 2010.
25. 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.
26. 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.
27. J. Katz and Y. Lindell. *Introduction to Modern Cryptography (Chapman & Hall/Crc Cryptography and Network Security Series)*. Chapman & Hall/CRC, 2007.
28. 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.
29. B. Li and D. Micciancio. On the security of homomorphic encryption on approximate numbers. *IACR Cryptology ePrint Archive*, 2020:1533, 2020.
30. 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.
31. G. Malavolta. Circuit privacy for quantum fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2020:1454, 2020.
32. 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.
33. C. Peikert. A decade of lattice cryptography. *Found. Trends Theor. Comput. Sci.*, 10(4):283–424, mar 2016.
34. 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.
35. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), Sept. 2009.

36. M. Rosulek. The joy of cryptography. `https://joyofcryptography.com`.
37. V. Shoup. A proposal for an ISO standard for public key encryption. *IACR Cryptol. ePrint Arch.*, page 112, 2001.
38. 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 Preliminaries (details omitted from Section 2)

In this section we specify in depth standard terminology and notations and definitions used throughout this paper, including public key encryption and CPA-security.

A.1 Terminology and Notations

We use the following standard notations and terminology. For $n \in \mathbb{N}$, let $[n]$ denote the set $\{1, \ldots, n\}$.

A function $\mu\colon \mathbb{N} \to \mathbb{R}^+$ is *negligible* in n if for every positive polynomial $p(\cdot)$ and all sufficiently large n it holds that $\mu(n) < 1/p(n)$. We use $\mathsf{neg}(\cdot)$ to denote a negligible function if we do not need to specify its name. Unless otherwise indicated, "polynomial" and "negligible" are measured with respect to a system parameter λ called the *security parameter*. We use the shorthand notation ppt for *probabilistic polynomial time* in λ.

A *probability ensemble* $X = \{X(a,n)\}_{a\in\{0,1\}^*, n\in\mathbb{N}}$ is an infinite sequence of random variables indexed by $a \in \{0,1\}^*$ and $n \in \mathbb{N}$. Let $X = \{X(a,n)\}_{a\in\{0,1\}^*, n\in\mathbb{N}}$ and $Y = \{Y(a,n)\}_{a\in\{0,1\}^*, n\in\mathbb{N}}$ be two probability ensembles. X and Y are said to be *computationally indistinguishable*, denoted by $X \approx_c Y$, if for every non-uniform polynomial-time algorithm $\mathcal{D}$ there exists a negligible function neg such that for every $a \in \{0,1\}^*$ and every $n \in \mathbb{N}$,

$$|\Pr[\mathcal{D}(X(a,n)) = 1] - \Pr[\mathcal{D}(Y(a,n)) = 1]| \leq \mathsf{neg}(n).$$

A *(strong) one-way function* is a polynomial time computable function $f\colon \{0,1\}^* \to \{0,1\}^*$ so that any ppt algorithm can invert f with at most negligible probability; See a formal Definition in Goldreich [23], Definition 2.2.1.

A.2 CPA-Secure Public Key Encryption

Public key encryption. A public key encryption scheme has the following syntax and correctness requirement.

Definition 16 (Public-Key Encryption (PKE)). *A* public-key encryption (PKE) scheme *with message space $\mathcal{M}$ is a triple* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *of ppt algorithms satisfying the following conditions:*

- Gen *(key generation) takes as input the security parameter 1^λ, and outputs a pair (pk, sk) consisting of a public key pk and a secret key sk; denoted: $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$.*

– Enc *(encryption) takes as input a public key* pk *and a message* $m \in \mathcal{M}$*, and outputs a ciphertext* c*; denoted:* $\mathbf{c} \leftarrow \mathsf{Enc}_{pk}(m)$.
– Dec *(decryption) takes as input a secret key* sk *and a ciphertext* c*, and outputs a decrypted message* m'*; denoted:* $m' \leftarrow \mathsf{Dec}_{sk}(c)$.

Correctness. The scheme is correct *if for every* (pk, sk) *in the range of* $\mathsf{Gen}(1^\lambda)$ *and every message* $m \in \mathcal{M}$*,*

$$\Pr[\mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(m)) = m] > 1 - \mathsf{neg}(\lambda)$$

where the probability is taken over the random coins of the encryption algorithm.

Security against chosen plaintext attack. A PKE $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ is CPA-secure if no ppt adversary $\mathcal{A}$ can distinguish between the encryption of two equal length messages x_0, x_1 of his choice. This is formally stated using the following experiment between a challenger Chal and the adversary $\mathcal{A}$.

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

1. $\mathsf{Gen}(1^\lambda)$ is run by Chal to obtain keys (pk, sk).
2. Chal provides the adversary $\mathcal{A}$ with pk. $\mathcal{A}$ sends to Chal two messages $x_0, x_1 \in \mathcal{M}$ 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 17 (CPA-security). *A public key encryption scheme* $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *has indistinguishable encryptions under chosen-plaintext attacks (or is* CPA-secure*) if for all* ppt *adversaries* $\mathcal{A}$ *there exists a negligible function* neg *such that:*

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

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

Security for multiple messages. A PKE $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ has $\mathsf{indistinguishable\ multiple\ encryptions}$ if no ppt adversary $\mathcal{A}$ can distinguish between the encryption of two vectors of equal length messages $X_0 = (x_0^1, \ldots, x_0^t)$ and $X_1 = (x_1^1, \ldots, x_1^t)$ of his choice. See formal definition in [27].

Theorem 14 (from [27], thm. 10.10). *If a public-key encryption scheme is CPA-secure, then it has indistinguishable multiple encryptions security.*

B Omitted Proofs from Section 3

We bring here formal proof details omitted from Section 3.

B.1 Proof of Lemma 2.

We prove Lemma 2 showing that for every $\mathcal{C}$-homomorphic public-key encryption scheme $\mathcal{E}$ that has a sanitization algorithm $\mathsf{Sanitize}$, its sanitized version $\mathcal{E}^{\mathsf{santz}}$ specified in Definition 8 is circuit-private$^+$ for $\mathcal{C}$.

Proof (of Lemma 2). Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be a $\mathcal{C}$-homomorphic public-key encryption scheme with a sanitization algorithm $\mathsf{Sanitize}$. 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 8. 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 and $(pk, sk) \leftarrow \mathsf{Gen}(\lambda)$. 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.

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

$$\begin{aligned} &\mathsf{Enc}^{\mathsf{santz}}_{pk}\left(C\left(\mathsf{Dec}_{sk}(c_1), \cdots, \mathsf{Dec}_{sk}(c_\ell)\right)\right) \\ &= \mathsf{Sanitize}_{pk}\left(\mathsf{Enc}_{pk}\left(C\left(\mathsf{Dec}_{sk}(c_1), \ldots, \mathsf{Dec}_{sk}(c_\ell)\right)\right)\right) \end{aligned} \tag{6}$$

and

$$\begin{aligned} &\mathsf{Eval}^{\mathsf{santz}}_{pk}\left(C, c_1, \ldots, c_\ell\right) \\ &= \mathsf{Sanitize}_{pk}\left(\mathsf{Eval}_{pk}\left(C, \mathsf{Sanitize}_{pk}(c_1), \ldots, \mathsf{Sanitize}_{pk}(c_\ell)\right)\right) \end{aligned} \tag{7}$$

By definition of the sanitization algorithm, 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 (specifically, $\mathsf{Enc}_{pk}\left(C\left(\mathsf{Dec}_{sk}(c_1), \ldots, \mathsf{Dec}_{sk}(c_\ell)\right)\right)$ and $\mathsf{Eval}_{pk}\left(C, \mathsf{Sanitize}_{pk}(c_1), \ldots, \mathsf{Sanitize}_{pk}(c_\ell)\right)$) decrypt to the same plaintext.

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

$$\forall i \in [\ell] : \Pr[\mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(\mathsf{Dec}_{sk}(c_i))) = \mathsf{Dec}_{sk}(c_i)] \geq 1 - \mathsf{neg}(\lambda) \tag{8}$$

and

$$\Pr\left[\begin{smallmatrix}\mathsf{Dec}_{sk}\left(\mathsf{Enc}_{pk}(C(\mathsf{Dec}_{sk}(c_1),\ldots,\mathsf{Dec}_{sk}(c_\ell)))\right)\\ =C(\mathsf{Dec}_{sk}(c_1),\ldots,\mathsf{Dec}_{sk}(c_\ell))\end{smallmatrix}\right] \geq 1 - \mathsf{neg}(\lambda) \tag{9}$$

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

From Equation 8 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,

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

The $\mathcal{C}$-homomorphism of $\mathcal{E}$ guarantees that also $\mathcal{E}^* = (\mathsf{Gen}, \mathsf{Enc}^{\mathsf{santz}}, \mathsf{Dec}, \mathsf{Eval})$ is $\mathcal{C}$-homomorphic, and hence 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,

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

Combining Equations 9, 10, and 11 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,

$$\begin{aligned}&\mathsf{Dec}_{sk}\left(\mathsf{Eval}_{pk}\left(C, \mathsf{Sanitize}_{pk}(c_1), \ldots, \mathsf{Sanitize}_{pk}(c_\ell)\right)\right)\\ =&\mathsf{Dec}_{sk}\left(\mathsf{Enc}_{pk}\left(C\left(\mathsf{Dec}_{sk}(c_1), \ldots, \mathsf{Dec}_{sk}(c_\ell)\right)\right)\right)\end{aligned} \tag{12}$$

Therefore, we can apply the the statistical sanitization property of $\mathcal{E}$, and 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

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

Combining the latter with Equations 6-7, we obtain that $\mathcal{E}^{\mathsf{santz}}$ is circuit-private$^+$. □

B.2 Proof of Lemma 3.

We prove Lemma 3 showing that if $\mathcal{E}$ is a CPA-secure $\mathcal{C}$-homomorphic public-key encryption scheme that is circuit-private$^+$ for $\mathcal{C}$, then it is funcCPA-secure with respect to $\mathcal{C}$.

Proof (of Lemma 3). 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 $\mathbf{e}' \leftarrow \mathsf{Eval}_{pk}\left(C_n, \mathbf{e}\right)$.
- 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,

$$\Pr[\mathsf{EXP}^{cpa}_{\mathcal{B},\mathcal{E}}(\lambda)=1]=\Pr[\mathsf{EXP}^{Fcpa}_{\mathcal{A}_q,\mathcal{E},\mathcal{C}}(\lambda)=1] \tag{13}$$

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 13 we conclude that

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

As required. □

B.3 Proof of Theorem 8.

We prove that $\mathsf{funcCPA}$-security of the underlying encryption scheme $\mathcal{E}$ implies privacy for $(\mathcal{E},\mathcal{G})$-aided outsourcing protocols.

Proof (Proof of Theorem 8). 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 $\lambda\in\mathbb{N}$:

$$\Pr[\mathcal{D}(\mathsf{view}^{\pi}_{\mathsf{Srv}^*}(x_1,\perp,\lambda))=1]-\Pr[\mathcal{D}(\mathsf{view}^{\pi}_{\mathsf{Srv}^*}(x_0,\perp,\lambda))=1]\geq\frac{1}{p(\lambda)} \tag{14}$$

We show that given $\mathcal{D}$ and Srv^* we can construct an adversary $\mathcal{A}$ that violates the $\mathsf{funcCPA}$ 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 $\mathsf{view}_{\mathsf{Srv}^*}^{\mathsf{EXP}^{Fcpa}}(x_b, \perp, \lambda)$ 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 $b \in \{0,1\}$,

$$\Pr[\mathcal{D}(\mathsf{view}^{\pi}_{\mathsf{Srv}^*}(x_b, \perp, \lambda)) = 1] = \Pr[\mathcal{D}(\mathsf{view}_{\mathsf{Srv}^*}^{\mathsf{EXP}^{Fcpa}}(x_b, \perp, \lambda)) = 1] \quad (15)$$

From Equations 14 and 15 it follows that:

$$\Pr[\mathcal{D}(\mathsf{view}_{\mathsf{Srv}^*}^{\mathsf{EXP}^{Fcpa}}(x_1, \perp, \lambda)) = 1] - \Pr[\mathcal{D}(\mathsf{view}_{\mathsf{Srv}^*}^{\mathsf{EXP}^{Fcpa}}(x_0, \perp, \lambda)) = 1] \geq \frac{1}{p(\lambda)} \quad (16)$$

Therefore, we obtain that:

$$\begin{aligned}
&\Pr[\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda) = 1] \\
&= \frac{1}{2} \cdot \big(\Pr[\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda) = 1 | b = 1] + \Pr[\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda) = 1 | b = 0] \big) \\
&= \frac{1}{2} \cdot \Big(\Pr[\mathcal{D}(\mathsf{view}_{\mathsf{Srv}^*}^{\mathsf{EXP}^{Fcpa}}(x_1, \perp, \lambda)) = 1] + \Pr[\mathcal{D}(\mathsf{view}_{\mathsf{Srv}^*}^{\mathsf{EXP}^{Fcpa}}(x_0, \perp, \lambda)) = 0] \Big) \\
&= \frac{1}{2} + \frac{1}{2} \cdot \Big(\Pr[\mathcal{D}(\mathsf{view}_{\mathsf{Srv}^*}^{\mathsf{EXP}^{Fcpa}}(x_1, \perp, \lambda)) = 1] - \Pr[\mathcal{D}(\mathsf{view}_{\mathsf{Srv}^*}^{\mathsf{EXP}^{Fcpa}}(x_0, \perp, \lambda)) = 1] \Big) \\
&\geq \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{p(\lambda)}
\end{aligned}$$

where the last inequality follows from Equation 16. Combining this with $\mathcal{A}$ being ppt we derive a contradiction to $\mathcal{E}$ being $\mathsf{funcCPA}$ secure. This concludes the proof. □

B.4 Proof of Theorem 9

In this section we given the proof of Theorem 9, showing that (a) if $\mathcal{E}$ is a compact and $\mathcal{C}$-homomorphic encryption scheme, then $\mathcal{E}^f$ is a compact and $\mathcal{C} \times \mathcal{C}$-homomorphic encryption scheme, see in Lemma 5; (b) if $\mathcal{E}$ is CPA-secure then $\mathcal{E}^f$ is CPA-secure, see Lemma 6.

Lemma 5 (correctness, homomorphism and compactness of $\mathcal{E}^f$). *For every public-key encryption scheme $\mathcal{E}$ with message-space $\mathcal{M}$, and every one-way function f over $\mathcal{M}$, the public-key encryption scheme $\mathcal{E}^f$ specified in Figure 1 is compact, and $\mathcal{C}\times\mathcal{C}$-homomorphic if $\mathcal{E}$ is compact, and $\mathcal{C}$-homomorphic.*

Proof. Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be a compact, $\mathcal{C}$-homomorphic public-key encryption scheme with message-space $\mathcal{M}$, and let f be a one-way function over $\mathcal{M}$. Let $\mathcal{E}^f = (\mathsf{Gen}^f, \mathsf{Enc}^f, \mathsf{Dec}^f, \mathsf{Eval}^f)$ be the encryption scheme specified in Figure 1. We show that the algorithms of $\mathcal{E}^f$ are ppt, and the scheme is correct, $\mathcal{C}\times\mathcal{C}$-homomorphic, and compact. We assume without loss of generality that the message-space and ciphertext-space of $\mathcal{E}$ are distinct; otherwise, change Enc to pad each ciphertext with an additional character that make it syntactically distinct from values in $\mathcal{M}$. Consequently, the condition $f(c_2) \neq f(m^*)$ tested in $\mathcal{E}^f$ trivially holds for all ciphertexts $(c_1, c_2) \leftarrow \mathsf{Enc}^f_{pk^f}(m_1, m_2)$ s.t. $f(m_2) \neq f(m^*)$.

Efficiency of $\mathcal{E}^f$. The algorithms of $\mathcal{E}^f$ involve only a constant number of calls to the algorithms of $\mathcal{E}$ and to computing the forward direction of the one-way function f. All these operations are in ppt, and therefore $\mathcal{E}^f$ is ppt.

Correctness of $\mathcal{E}^f$. Fix some key-pair $(pk^f, sk^f) \leftarrow \mathsf{Gen}^f(1^\lambda)$, where $pk^f = (pk, \mathsf{Enc}_{pk}(m^*), f(m^*))$ and $sk^f = (sk, f(m^*))$ for (pk, sk) in the range of $\mathsf{Gen}(1^\lambda)$ and $m^* \in \mathcal{M}$. Fix some message $m = (m_1, m_2)$ in the message space $\mathcal{M}\times\mathcal{M}$ and let $c = (c_1, c_2) \leftarrow \mathsf{Enc}^f_{pk^f}(m)$. We show that $\mathsf{Dec}^f_{sk^f}(c) = m$ as follows:

- if $f(m_2) \neq f(m^*)$, then $(c_1, c_2) = (\mathsf{Enc}_{pk}(m_1), \mathsf{Enc}_{pk}(m_2))$ and
$$\mathsf{Dec}^f_{sk^f}(c) = (\mathsf{Dec}_{sk}(c_1), \mathsf{Dec}_{sk}(c_2)) = (m_1, m_2) = m$$
where the first equality holds since $c_2 \neq m^*$ by the premise that $\mathcal{M}$ and $\mathcal{C}$ do not intersect, and the second equality holds by the correctness of $\mathcal{E}$.
- if $f(m_2) = f(m^*)$, then $c = m$ (by definition of $\mathsf{Enc}^f_{pk^f}$), implying that $c_2 = m^*$ and therefore $\mathsf{Dec}^f_{sk^f}(c) = c$ (by definition of $\mathsf{Dec}^f_{sk^f}$). So again $\mathsf{Dec}^f_{sk^f}(c) = m$.

We conclude that in both cases, $\mathsf{Dec}^f_{sk^f}(\mathsf{Enc}^f_{pk^f}(m)) = m$.

Compactness of $\mathcal{E}^f$. We show that there exists polynomial $p(\cdot)$ such that the decryption algorithm Dec^f of $\mathcal{E}^f$ can be expressed as a circuit of size $p(\lambda)$. The decryption of $\mathcal{E}^f$ involves the following computations: (a) executing twice the decryption algorithm of $\mathcal{E}$, (b) evaluating the one-way function $f(c_2)$, and (c) testing equality between $f(c_2)$ and the value $f(m^*)$ provided as part of the secret key. All these computations are computable by poly-size circuits: (a) – due to the compactness of $\mathcal{E}$; (b) – since the forward direction of one-way functions is computable in time polynomial in the input size and the input c_2 is of size polynomial in λ due to the decryption algorithm Dec in $\mathcal{E}$ being a ppt algorithm; and (c) – as checking equality of two values of size $\mathsf{poly}(\lambda)$ is computable in time polynomial in λ.

Homomorphism of $\mathcal{E}^f$. Fix some key-pair $(pk^f, sk^f) \leftarrow \mathsf{Gen}^f(1^\lambda)$, where $pk^f = (pk, \mathsf{Enc}_{pk}(m^*), f(m^*))$ and $sk^f = (sk, f(m^*))$ for (pk, sk) in the range of $\mathsf{Gen}(1^\lambda)$ and $m^* \in \mathcal{M}$. Fix a circuit $C = (C_1, C_2) \in \mathcal{C} \times \mathcal{C}$ and a set of inputs $(x_1, \ldots, x_\ell) \in (\mathcal{M} \times \mathcal{M})^\ell$ to C where $x_i = (x_{i,1}, x_{i,2})$ consists of the i-th input to C_1 and the i-th input to C_2, respectively, and let $c_i = (c_{i,1}, c_{i,2}) \leftarrow \mathsf{Enc}^f_{pk^f}(x_i)$.

We show that $\mathsf{Dec}^f_{sk^f}(\mathsf{Eval}^f_{pk^f}(C; c_1, \ldots, c_\ell)) = C(x_1, \ldots, x_\ell)$ with overwhelming probability. First we observe that by definition of Eval^f,

$$\begin{aligned}\mathsf{Eval}^f_{pk^f}(C; c_1, ..., c_\ell) = (&\mathsf{Eval}_{pk}(C_1; \mathsf{Enc}_{pk}(x_{1,1}), ..., \mathsf{Enc}_{pk}(x_{\ell,1}))),\\ &\mathsf{Eval}_{pk}(C_2; \mathsf{Enc}_{pk}(x_{1,2}), ..., \mathsf{Enc}_{pk}(x_{\ell,2})))\end{aligned}$$

Next, by definition of Dec^f,

$$\begin{aligned}\mathsf{Dec}^f_{sk^f}(\mathsf{Eval}^f_{pk^f}(C; c_1, ..., c_\ell)) = (&\mathsf{Dec}_{sk}(\mathsf{Eval}_{pk}(C_1; \mathsf{Enc}_{pk}(x_{1,1}), ..., \mathsf{Enc}_{pk}(x_{\ell,1}))),\\ &\mathsf{Dec}_{sk}(\mathsf{Eval}_{pk}(C_2; \mathsf{Enc}_{pk}(x_{1,2}), ..., \mathsf{Enc}_{pk}(x_{\ell,2}))))\end{aligned}$$

Finally by the $\mathcal{C}$-homomorphism of $\mathcal{E}$, for every, the latter is equal to:

$$\begin{aligned}&= (C_1(x_{1,1}, \ldots, x_{\ell,1}), C_2(x_{1,2}, \ldots, x_{\ell,2}))\\ &= C(x_1, \ldots, x_\ell)\end{aligned}$$

with overwhelming probability over the random coins of the experiment. We conclude that

$$\Pr[\mathsf{Dec}^f_{sk^f}(\mathsf{Eval}^f_{pk^f}(C; c_1, ..., c_\ell)) \neq C(x_1, \ldots, x_\ell)] < \mathsf{neg}(\lambda)$$

which concludes the proof. □

Lemma 6 (CPA-security of $\mathcal{E}^f$). *Suppose $\mathcal{E}$ is a CPA-secure public-key encryption scheme with message space $\mathcal{M}$, and f is a one-way function over $\mathcal{M}$. Then $\mathcal{E}^f$ is a CPA-secure public-key encryption scheme with message space $\mathcal{M} \times \mathcal{M}$.*

Proof. Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ be CPA-secure public-key encryption scheme with message-space $\mathcal{M}$, and let f be a one-way function over $\mathcal{M}$. Let $\mathcal{E}^f = (\mathsf{Gen}^f, \mathsf{Enc}^f, \mathsf{Dec}^f, \mathsf{Eval}^f)$ be the encryption scheme specified in Figure 1. To prove $\mathcal{E}^f$ is CPA-secure we gradually change $\mathcal{E}$ into $\mathcal{E}^f$ while showing that CPA-security is preserved under all the modifications we introduce. Namely, we first define a sequence of encryption schemes starting from $\mathcal{E}$, going through $\tilde{\mathcal{E}}, \tilde{\mathcal{E}}^f$ and into $\mathcal{E}^f$ (see definitions for $\tilde{\mathcal{E}}, \tilde{\mathcal{E}}^f$ below), and show that each one is CPA-secure based on the CPA-security of the previous encryption schemes.

The encryption scheme $\tilde{\mathcal{E}}$ and its CPA-security. is similar to $\mathcal{E}$ except for encrypting pairs of messages rather than a single message. That is,

- $\tilde{\mathsf{Gen}}$ takes as input the security parameter 1^λ, and outputs $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$
- $\tilde{\mathsf{Enc}}$ takes as input a public key pk and a message $m = (m_1, m_2) \in \mathcal{M} \times \mathcal{M}$, and outputs a ciphertext $(\mathsf{Enc}_{pk}(m_1), \mathsf{Enc}_{pk}(m_2))$
- $\tilde{\mathsf{Dec}}$ takes as input a sk and a ciphertext $c = (c_1, c_2)$, and outputs $(\mathsf{Dec}_{sk}(c_1), \mathsf{Dec}_{sk}(c_2))$
- $\tilde{\mathsf{Eval}}$ takes as input a public key pk, a function $C = (C_1, C_2) \in \mathcal{C} \times \mathcal{C}$ and ℓ ciphertexts $c_1 = (c_{1,1}, c_{1,2}), \ldots, c_\ell = (c_{\ell,1}, c_{\ell,2})$, and outputs $(\mathsf{Eval}_{pk}(C_1; c_{1,1}, \ldots, c_{\ell,1}), \mathsf{Eval}_{pk}(C_2; c_{1,2}, \ldots, c_{\ell,2}))$

By Theorem 14 the CPA-security of $\mathcal{E}$ implies that it has indistinguishable multiple encryptions security, implying that $\tilde{\mathcal{E}}$ is CPA-secure scheme.

The key augmented encryption scheme $\tilde{\mathcal{E}}^f$ and its CPA-security. The scheme $\tilde{\mathcal{E}}^f$ is similar to $\tilde{\mathcal{E}}$ except for augmenting the public pk with $\mathsf{Enc}_{pk}(m^*)$ and $f(m^*)$ for a random messages $m^* \in \mathcal{M}$. That is, $\tilde{\mathsf{Gen}}^f$ on input the security parameter 1^λ samples $(pk, sk) \leftarrow \tilde{\mathsf{Gen}}(1^\lambda)$ and a uniformly random message $m^* \in \mathcal{M}$, and outputs (pk^f, sk^f) for

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

and the rest of the algorithms remain the same, i.e., $\tilde{\mathsf{Enc}}^f_{pk^f}(m)$ outputs $\tilde{\mathsf{Enc}}_{pk}(m)$, $\tilde{\mathsf{Dec}}^f_{sk^f}(c)$ outputs $\tilde{\mathsf{Dec}}_{sk}(c)$, and $\tilde{\mathsf{Eval}}^f_{pk^f}(C; c_1, , ..., c_\ell)$ outputs $\tilde{\mathsf{Eval}}_{pk}(C; c_1, , ..., c_\ell)$.

We now show that $\tilde{\mathcal{E}}^f$ is CPA-secure based on the CPA-security of $\tilde{\mathcal{E}}$. Suppose towards contradiction that $\tilde{\mathcal{E}}^f$ is not CPA-secure, namely, there exists a ppt adversary $\tilde{\mathcal{A}}^f$ and a polynomial $p()$ such that:

$$\Pr[\mathsf{EXP}^{cpa}_{\tilde{\mathcal{A}}^f,\tilde{\mathcal{E}}^f}(\lambda)=1] \geq \frac{1}{2}+\frac{1}{p(\lambda)}. \tag{17}$$

We construct a ppt adversary $\tilde{\mathcal{A}}$ participating in the CPA experiment $\mathsf{EXP}^{cpa}_{\tilde{\mathcal{A}},\tilde{\mathcal{E}}}(\lambda)$ for $\tilde{\mathcal{E}}$.

The adversary $\tilde{\mathcal{A}}$ internally runs $\tilde{\mathcal{A}}^f$ while augmenting the public key with $\mathsf{Enc}_{pk}(m^*)$ and $f(m^*)$ for a randomly chosen $m^* \in \mathcal{M}$. It forwards Chal the two messages $x_0, x_1 \in \mathcal{M} \times \mathcal{M}$ chosen by $\tilde{\mathcal{A}}^f$, and feeds back the challenge ciphertext received. Finally, it outputs the bit $\tilde{\mathcal{A}}^f$ outputs.

The view of $\tilde{\mathcal{A}}^f$ when it is run internally by $\tilde{\mathcal{A}}$ is identical to the view of $\tilde{\mathcal{A}}^f$ in the CPA experiment $\mathsf{EXP}^{cpa}_{\tilde{\mathcal{A}}^f,\tilde{\mathcal{E}}^f}(\lambda)$. Together with Equation 17 we obtain that

$$\Pr[\mathsf{EXP}^{cpa}_{\tilde{\mathcal{A}},\tilde{\mathcal{E}}}(\lambda)=1] = \Pr[\mathsf{EXP}^{cpa}_{\tilde{\mathcal{A}}^f,\tilde{\mathcal{E}}^f}(\lambda)] \geq \frac{1}{2}+\frac{1}{p(\lambda)}$$

in contradiction to $\tilde{\mathcal{E}}$ being CPA-secure, and hence we conclude that $\tilde{\mathcal{E}}^f$ is CPA-secure.

Proof of CPA-security of $\mathcal{E}^f$ based on the CPA-security of $\tilde{\mathcal{E}}^f$. Informally, the CPA-security follows from the CPA-security of $\tilde{\mathcal{E}}^f$ together with the fact that the punctured code in Enc, Dec, and Eval algorithms is executed only only with negligible probability due to m^* being randomly sampled.

Suppose towards contradiction that $\mathcal{E}^f$ is not CPA-secure, namely, there exists a ppt adversary $\mathcal{A}^f$ and a polynomial $p()$ such that:

$$\Pr[\mathsf{EXP}^{cpa}_{\mathcal{A}^f,\mathcal{E}^f}(\lambda)=1] \geq \frac{1}{2}+\frac{1}{p(\lambda)}. \tag{18}$$

We construct a ppt adversary $\tilde{\mathcal{A}}^f$ participating in the CPA experiment $\mathsf{EXP}^{cpa}_{\tilde{\mathcal{A}}^f,\tilde{\mathcal{E}}^f}(\lambda)$ for $\tilde{\mathcal{E}}^f$. The adversary $\tilde{\mathcal{A}}^f$ behaves as follows:

1. upon receiving from Chal a public key $pk^f = (pk, \mathsf{Enc}_{pk}(m^*), f(m^*))$ generated by $(pk^f, sk^f) \leftarrow \tilde{\mathsf{Gen}}^f(1^\lambda)$, it forwards pk^f to $\mathcal{A}^f$.
2. Upon receiving from $\mathcal{A}^f$ two messages $x_0 = (x_{0,1}, x_{0,2}), x_1 = (x_{1,1}, x_{1,2}) \in \mathcal{M} \times \mathcal{M}$, it forwards to Chal the message x_0, x_1 if $f(x_{i,2}) \neq f(m^*)$ for both $i \in \{0,1\}$, and aborts otherwise.

3. Upon receiving the challenge ciphertext $c \leftarrow \tilde{\mathsf{Enc}}^{f}_{pk^f}(x_b)$ for a uniformly random bit $b \in \{0,1\}$, it forwards c to $\mathcal{A}^f$.
4. $\tilde{\mathcal{A}}^f$ outputs whatever $\mathcal{A}^f$ outputs.

The adversary $\tilde{\mathcal{A}}^f$ is ppt since $\mathcal{A}^f$ is ppt and the condition in 2 is efficiently testable.

Denote by E the event that $\tilde{\mathcal{A}}^f$ aborts in $\mathsf{EXP}^{cpa}_{\tilde{\mathcal{A}}^f,\tilde{\mathcal{E}}^f}(\lambda)$, i.e., the event that $\mathcal{A}^f$ in $\mathsf{EXP}^{cpa}_{\mathcal{A}^f,\mathcal{E}^f}(\lambda)$ sends a message $m = (m_1, m_2)$ s.t. $f(m_2) = f(m^*)$ to the challenger Chal in the chosen pair of message. Observe that,

$$\Pr[\mathsf{EXP}^{cpa}_{\tilde{\mathcal{A}}^f,\tilde{\mathcal{E}}^f}(\lambda) = 1] = \Pr[\mathsf{EXP}^{cpa}_{\mathcal{A}^f,\mathcal{E}^f}(\lambda) = 1 \text{ and } \neg E]. \tag{19}$$

Moreover,

$$\begin{aligned}
&\Pr[\mathsf{EXP}^{cpa}_{\mathcal{A}^f,\mathcal{E}^f}(\lambda) = 1 \text{ and } \neg E] \\
&= \Pr[\mathsf{EXP}^{cpa}_{\mathcal{A}^f,\mathcal{E}^f}(\lambda) = 1] - \Pr[\mathsf{EXP}^{cpa}_{\mathcal{A}^f,\mathcal{E}^f}(\lambda) = 1 \text{ and } E] \\
&\geq \Pr[\mathsf{EXP}^{cpa}_{\mathcal{A}^f,\mathcal{E}^f}(\lambda) = 1 \text{ and } E] - \Pr[E] \\
&\geq \frac{1}{2} + \frac{1}{p(\lambda)} - \Pr[E]
\end{aligned}$$

where the last inequality follows from Equation 18.

To conclude the proof it is left to show that E occurs with at most a negligible probability, by the premise that f is one-way and $\mathcal{E}$ is CPA-secure. Toward this, we first show that the probability that $\tilde{\mathcal{A}}^f$ aborts is the same (up to a negligible difference) regardless of whether it is given a valid public key $pk^f = (pk, c, f(m^*))$ where $c \leftarrow \mathsf{Enc}_{pk}(m^*)$ or an invalid key where $c \leftarrow \mathsf{Enc}_{pk}(r)$ for a uniformly random message $r \in \mathcal{M}$ independent of m^*. Denote by $\tilde{\mathcal{E}}^{f-inv}$ the scheme $\tilde{\mathcal{E}}^f$ but with $pk^f = (pk, \mathsf{Enc}_{pk}(r), f(m^*))$ for a uniformly random message $r \in \mathcal{M}$. Similarly, we denote by E' the event that $\tilde{\mathcal{A}}^f$ aborts in $\mathsf{EXP}^{cpa}_{\tilde{\mathcal{A}}^f,\tilde{\mathcal{E}}^{f-inv}}(\lambda)$.

We prove (1) a negligible probability gap between abort events: $|\Pr[E] - \Pr[E']| < \mathsf{neg}(\lambda)$ relying on the CPA-security of $\mathcal{E}$, and (2) a negligible probability of abort: $\Pr[E'] < \mathsf{neg}(\lambda)$ relying on the one-wayness of f.

Proof of a negligible probability gap between abort events. Assume towards contradiction that there exists a polynomial $p()$, such that

$$|\Pr[E'] - \Pr[E]| \geq \frac{1}{p(\lambda)} \tag{20}$$

We construct an adversary $\mathcal{B}_{cpa}$ that breaks the CPA-security of $\mathcal{E}$. That is, $\mathcal{B}_{cpa}$ participates in $\mathsf{EXP}^{cpa}_{\mathcal{B}_{cpa},\mathcal{E}}(\lambda)$ and behaves as follows:

1. Given a public key pk generated by $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$, $\mathcal{B}_{cpa}$ sends to Chal two independent uniformly random messages $m_0, m_1 \in \mathcal{M}$.
2. Upon receiving the challenge ciphertext $c = \mathsf{Enc}_{pk}(m_b)$ from Chal (on a randomly sampled bit b by Chal), $\mathcal{B}_{cpa}$ internally executes $\tilde{\mathcal{A}}^f$ on $pk^f = (pk, c, f(m_0))$ while playing the role of the challenger (i.e, it receives two messages x_0, x_1 from $\tilde{\mathcal{A}}^f$, picks a random bit t, and feeds $\tilde{\mathcal{A}}^f$ with $\tilde{\mathsf{Enc}}^f_{pk^f}(x_t)$).
3. $\mathcal{B}_{cpa}$ outputs $b' = 1$ if $\tilde{\mathcal{A}}^f$ aborts, and $b' = 0$ otherwise.

Clearly $\mathcal{B}_{cpa}$ is ppt, since $\tilde{\mathcal{A}}^f$ is ppt.

Observe that in $\mathsf{EXP}^{cpa}_{\mathcal{B}_{cpa},\mathcal{E}}(\lambda)$, the event E corresponds to the case of an abort on $c = \mathsf{Enc}_{pk}(m_0)$, i.e. when $b = 0$; whereas E' corresponds to the case of an abort on $c = \mathsf{Enc}_{pk}(m_1)$, i.e. when $b = 1$. That is,

$$\Pr[b' = 1 | b = 0] = \Pr[E]$$
$$\Pr[b' = 1 | b = 1] = \Pr[E'].$$

Therefore,

$$\begin{aligned}
&\Pr[\mathsf{EXP}^{cpa}_{\mathcal{B}_{cpa},\mathcal{E}}(\lambda) = 1] \\
&= \Pr[\mathsf{EXP}^{cpa}_{\mathcal{B}_{cpa},\mathcal{E}}(\lambda) = 1 | b = 0] \cdot \Pr[b = 0] + \Pr[\mathsf{EXP}^{cpa}_{\mathcal{B}_{cpa},\mathcal{E}}(\lambda) = 1 | b = 1] \cdot \Pr[b = 1] \\
&= \Pr[b' = 0 | b = 0] \cdot \Pr[b = 0] + \Pr[b' = 1 | b = 1] \cdot \Pr[b = 1] \\
&= \frac{1}{2} \cdot \big((1 - \Pr[b' = 1 | b = 0]) + \Pr[b' = 1 | b = 1]\big) \\
&= \frac{1}{2} \cdot \big((1 - \Pr[E]) + \Pr[E']\big) \\
&= \frac{1}{2} + \frac{1}{2} \cdot \big(\Pr[E'] - \Pr[E]\big) \\
&\geq \frac{1}{2} + \frac{1}{2 \cdot p(\lambda)}
\end{aligned}$$

where the last inequality follows from Equation 20, and w.l.o.g assumption that $\Pr[E'] \geq \Pr[E]$ (otherwise $\mathcal{B}_{cpa}$ returns $b' = 0$ in case of an abort). This contradicts the CPA-security of $\mathcal{E}$, and hence implies $|\Pr[E'] - \Pr[E]| < \mathsf{neg}(\lambda)$.

Proof of a negligible abort probability. Suppose for contradiction that there exists a polynomial $p(\cdot)$ such that

$$Pr[E'] \geq \frac{1}{p(\lambda)} \tag{21}$$

We construct a ppt adversary $\mathcal{B}_{owf}$ that inverts f, and behaves as follows:

1. Given $f(m^*)$ for a uniformly random $m^* \in \mathcal{M}$, $\mathcal{B}_{owf}$ first generates keys $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$, chooses a uniformly random $r \in \mathcal{M}$, computes $\mathsf{Enc}_{pk}(r)$ and sets $pk^f = (pk, \mathsf{Enc}_{pk}(r), f(m^*))$.
2. Next, $\mathcal{B}_{owf}$ executes $\mathsf{EXP}^{cpa}_{\tilde{\mathcal{A}}^f, \tilde{\mathcal{A}}^{f-inv}}$ with the public key pk^f, and plays the role of the challenger Chal.
3. If $\tilde{\mathcal{E}}^f$ aborts, i.e., it received two messages $m_0 = (m_{0,1}, m_{0,2}), m_1 = (m_{1,1}, m_{1,2}) \in \mathcal{M} \times \mathcal{M}$, such that $f(m_{i,2}) = f(m^*)$ for either $i \in \{0, 1\}$, then $\mathcal{B}_{owf}$ outputs $m_{i,2}$ for the relevant i as a pre-image for its input $f(m^*)$. Otherwise, $\mathcal{B}_{owf}$ fails to invert f.

It follows from the construction of $\mathcal{B}_{owf}$ together with Equation 21 that

$$\Pr[\mathcal{B}_{owf} \text{ invers } f] = \Pr[E'] \geq \frac{1}{p(\lambda)} \tag{22}$$

which is a contradiction to f being a one-way function.

We have proven that CPA-security $\mathcal{E}$ together with one-wayness of f implies CPA-security $\mathcal{E}^f$ which concludes the proof. □

C Omitted Proofs from Section 4

We bring here formal proof details omitted from Section 4.

C.1 Proof of Lemma 4

Proof (Proof of Lemma 4). Assume by contradiction that Lemma 4 does not hold. That is, there exists a ppt adversary $\mathcal{A}$ and a polynomial $p(\cdot)$ such that for infinitely many λ and L,

$$\begin{aligned} &\Pr[\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda, L) = 1] \\ &- \Pr[\mathsf{EXP}^{Fcpa^\#}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda, L) = 1] \geq \frac{1}{p(\lambda)}. \end{aligned} \tag{23}$$

We define a series of hybrid executions that gradually move between $\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda, L)$ execution (where Chal responds with $\mathsf{Enc}_{pk_{\ell-1}}(G_n(\mathsf{Dec}_{sk_\ell}(\mathbf{e}_\ell)))$) to $\mathsf{EXP}^{Fcpa^{\#}}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda, L)$ execution (where Chal responds with an encryption of the image of G_n on an arbitrary message). Let L denote the number of levels. We define $L+1$ hybrids as follows:

Hybrid H_0 is defined as the execution of $\mathsf{EXP}^{Fcpa}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda, L)$.

Hybrid H_j $(j = 1, \ldots, L)$ is similar to H_0 except that the queries to $\{\mathsf{Enc}_{pk_{\ell-1}}(\mathcal{G}(\mathsf{Dec}_{sk_\ell}(\cdot)))\}_{\ell\in[L]}$ oracle for $\ell \leq j$, each query $(\mathbf{e}_\ell, n)$ is answered by $\mathsf{Enc}_{pk_{\ell-1}}(G_n(m))$ for an arbitrary $m \in \mathcal{M}$ (instead of sending $\mathsf{Enc}_{pk_{\ell-1}}(G_n(\mathsf{Dec}_{sk_\ell}(\mathbf{e}_\ell)))$ as in Definition 11, Step 2).

Note that in each pair of adjacent hybrids H_{j-1} and H_j for $j \in [L]$ the difference is that in H_j all the queries of level j ciphertexts are answered using $G_n(m)$ for an arbitrary m instead of $\mathsf{Dec}_{sk_j}(\mathbf{e}_j)$.

Denote by $\mathsf{EXP}^{H_j}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda, L)$ the output of the experiment in hybrid H_j.

By the hybrid argument it follows from Equation 23 that there exists $j \in [L]$ such that:

$$\begin{aligned} &\Pr[\mathsf{EXP}^{H_{j-1}}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda, L) = 1] \\ &- \Pr[\mathsf{EXP}^{H_j}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda, L) = 1] \geq \frac{1}{L} \cdot \frac{1}{p(\lambda)} \end{aligned} \tag{24}$$

We show that Equation 24 contradicts $\mathcal{E}$ being CPA secure. That is, we construct an adversary $\mathcal{B}$ that communicates with the challenger Chal in the CPA indistinguishability experiment $\mathsf{EXP}^{cpa}_{\mathcal{B},\mathcal{E}}$ and wins with a non-negligible advantage over half. Concretely, $\mathcal{B}$ participates in $\mathsf{EXP}^{cpa}_{\mathcal{B},\mathcal{E}}$ by internally running $\mathcal{A}$ as follows:

1. Upon receiving $\{pk\}_{\ell\in\{0,\ldots,L\}}$ from Chal, $\mathcal{B}$ computes new keys (pk'_ℓ, sk'_ℓ) for every $j \leq \ell \leq L$, and forwards $\{pk_\ell\}_{\ell<j} \cup \{pk'_\ell\}_{j\leq\ell\leq L}$ while answering each query of $\mathcal{A}$ as follows:
   (a) For queries to oracle with $(\mathbf{e}_\ell, n)$ for ciphertexts of level $\ell < j$ respond with $\mathsf{Enc}_{pk_{\ell-1}}(G_n(m))$ for an arbitrary $m \in \mathcal{M}$.
   (b) For queries to oracle with $(\mathbf{e}_\ell, n)$ for ciphertexts of level $\ell > j$ respond with $\mathsf{Enc}_{pk'_{\ell-1}}(G_n(\mathsf{Dec}_{sk'_\ell}(e_\ell)))$.
   (c) For queries to oracle with $(\mathbf{e}_\ell, n)$ for ciphertexts of level $\ell = j$ proceeds as follows:
      i. $\mathcal{B}$ sets $m_0 = G_n(\mathsf{Dec}_{sk'_j}(\mathbf{e}_j))$, samples uniformly random $m \in \mathcal{M}$, and sends m_0 and $m_1 = G_n(m)$ and $j-1$ to Chal.

ii. Upon receiving from Chal the challenge ciphertext $c^* \leftarrow \mathsf{Enc}_{pk_{j-1}}(m_{b^*})$ for uniformly random $b^* \leftarrow \{0,1\}$, forward ciphertext c^* to $\mathcal{A}$.

(d) Once $\mathcal{A}$ generates x_0, x_1 and ℓ choose a random $b \in \{0,1\}$ and respond with $c \leftarrow \mathsf{Enc}_{pk_\ell}(x_b)$ to $\mathcal{A}$.

2. Let b' be the output of $\mathcal{A}$. $\mathcal{B}$ outputs 0 if $b' = b$ and 1 otherwise.

We note that if $b^* = 0$, then the challenge ciphertext c^* is an encryption under pk_{j-1} of $G_n(\mathsf{Dec}_{sk_j}(\mathsf{e}_j))$ and of a random element in the range of G_n otherwise; moreover, since $\mathcal{E}$ has independent level keys then the "fake" and real keys are identically distributed, i.e., $\{sk_\ell, pk_\ell\}_{\ell\in\{0,\ldots,L\}} \equiv \{sk_\ell, pk_\ell\}_{\ell<j} \cup \{sk'_\ell, pk'_\ell\}_{j\le\ell\le L}$ Therefore, if $b^* = 0$ then the view of $\mathcal{A}$ is exactly as in H_{j-1} and otherwise as in H_j. We obtain that

$$\begin{aligned}
&\Pr[\mathsf{EXP}^{cpa}_{\mathcal{B},\mathcal{E}}(\lambda) = 1] \\
=&\frac{1}{2}\cdot\big(\Pr[\mathsf{EXP}^{cpa}_{\mathcal{B},\mathcal{E}}(\lambda) = 1|b^* = 0] + \Pr[\mathsf{EXP}^{cpa}_{\mathcal{B},\mathcal{E}}(\lambda) = 1|b^* = 1]\big) \\
=&\frac{1}{2}\cdot\big(\Pr[b' = b|b^* = 0] + \Pr[b' \neq b|b^* = 1]\big) \\
=&\frac{1}{2}\cdot\big(\Pr[b' = b|b^* = 0] + (1 - \Pr[b' = b|b^* = 1])\big) \\
=&\frac{1}{2} + \frac{1}{2}\Big(\Pr[\mathsf{EXP}^{H_{j-1}}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda, L) = 1] - \Pr[\mathsf{EXP}^{H_j}_{\mathcal{A},\mathcal{E},\mathcal{G}}(\lambda, L) = 1]\Big) \\
\ge&\frac{1}{2} + \frac{1}{2}\cdot\frac{1}{L}\cdot\frac{1}{p(\lambda)}
\end{aligned} \tag{25}$$

and since L is polynomial in λ we obtain

$$\Pr[\mathsf{EXP}^{cpa}_{\mathcal{B},\mathcal{E}}(\lambda) = 1] \ge \frac{1}{2} + \frac{1}{2}\cdot\frac{1}{q(\lambda)}$$

for some polynomial $q(\cdot)$, in contradiction to the CPA-security of $\mathcal{E}$; this concludes the proof. □

C.2 Proof of Theorem 12

Proof (of Theorem 12). 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 $\mathsf{funcCPA}$-security of the scheme.

The adversary $\mathcal{B}$ participates in the $\mathsf{funcCPA}$-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 $\mathsf{funcCPA}$ 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 $\mathsf{funcCPA}$-security of $\mathcal{E}$. □

C.3 Proof of Proposition 2

This section presents the proof of Proposition 2.

Proof (of Proposition 2). The proof is case-by-case, presenting in each case an algorithm Alg producing ciphertexts so that from their decryption the secret key is efficiently computable (with probability $1 - \mathsf{neg}(\lambda)$ when accounting for possible decryption errors). The secret key can then be transformed to the encoding required by the decryption circuit.

Case I: $\mathcal{E}$ is LWE-based with least significant bit encoding. Let $\delta_i \in \{0,1\}^n$ denote the indicator vector $\delta_i(j) = 1$ if-and-only-if $j = i$. Let $\mathsf{Alg}_1(pk)$ be the algorithm that given pk computes $c \leftarrow \mathsf{Enc}_{pk}(0)$, computes $c_i = c + (0, \delta_i) \mod q$ for $i \in [n]$ and outputs $(c_1, \ldots, c_n)$. Observe that, for all $i \in [n]$, $\mathsf{Dec}_{sk}(c_i)$ returns the ith coordinate of the secret key $sk =$

$(1, s_1, \ldots, s_n) \in \mathbb{Z}_q^{n+1}$ as follows:

$$\begin{aligned}
\mathsf{Dec}_{sk}(c_i) &= \Big[\big[\langle c_i, sk\rangle\big]_q\Big]_p \text{ (by definition of } \mathsf{Dec}) \\
&= \Big[\big[\langle c, sk\rangle + \langle (0, \delta_i), sk\rangle\big]_q\Big]_p \text{ (by definition of } c_i \text{ and linearity of inner-product}) \\
&= \Big[\big[(aq + bp + 0) + s_i\big]_q\Big]_p \text{ for } a, b \in \mathbb{Z} \text{ s.t. } |bp| < q \\
&\quad \text{(by correctness of } \mathsf{Dec}_{sk}(c) \text{ and definition of } \delta_i) \\
&= s_i \text{ (since } s \text{ has small coefficients)}
\end{aligned}$$

Namely, given $(\mathsf{Dec}_{sk}(c_1), \ldots, \mathsf{Dec}_{sk}(c_n)) = (s_1, \ldots, s_n)$ we can efficiently recover the secret key sk.

Case II: $\mathcal{E}$ is RLWE-based with least significant bit encoding. Let $\delta(X) \in R_q$ be the polynomial $\delta(X) = X$. Let $\mathsf{Alg}_2(pk)$ be the algorithm that given pk computes $c \leftarrow \mathsf{Enc}_{pk}(0)$, computes $c' = c + (0, \delta)$ in R_q^2 and outputs c'. Recall that $sk = (\mathbf{1}, s) \in R_q^2$ for $s(X) = \sum_{i=0}^{d-1} s_i X^i$ a polynomial, and denote the coefficients of s by $(s_{d-1}, \ldots, s_0)$. We show that $\mathsf{Dec}_{sk}(c')$ returns the polynomial $s' = \delta \cdot s$ in R_q, from which sk can be efficiently computed.

$$\begin{aligned}
\mathsf{Dec}_{sk}(c') &= \Big[\big[\langle c', sk\rangle\big]_q\Big]_p \text{ (by definition of } \mathsf{Dec}) \\
&= \Big[\big[\langle c, sk\rangle + \langle (0, \delta), sk\rangle\big]_q\Big]_p \text{ (by definition of } c' \text{ and linearity of inner-product}) \\
&= \Big[\big[(aq + bp + 0) + s'\big]_q\Big]_p \text{ for } a, b \in \mathbb{Z} \text{ s.t. } |bp| < q \text{ and } s' = \delta \cdot s \\
&\quad \text{(by correctness of } \mathsf{Dec}_{sk}(c) \text{ and definition of } sk = (1, s)) \\
&= s' \text{ (since } s' \text{ has small coefficients)}
\end{aligned}$$

We show that s' has coefficients $(s_{d-2}, \ldots, s_0, -s_{d-1})$:

$$\delta(X) \cdot s(X) = \sum_{i=0}^{d-1} s_i X^{i+1} = -s_{d-1} + \sum_{i=1}^{d-1} s_{i-1} X^i$$

where the last equality follows from $X^d = -1 \mod F[X]$. We conclude that, from the coefficients $(s'_{d-1}, \ldots, s'_0)$ of $s' \leftarrow \mathsf{Dec}_{sk}(c')$, we can efficiently compute the coefficients of s by setting $s_{d-1} := -s'_0$ and $s_i := s'_{i+1}$ for all $i \in \{0, \ldots, d-2\}$, and outputting $sk = (\mathbf{1}, s_{d-1}, \ldots, s_0)$. □

D Omitted Proofs from Section 5

We bring here formal proof details omitted from Section 5.

D.1 Proof of Theorem 13

We prove that cleartext computable $(\mathcal{E}, \mathcal{G})$-aided outsourcing protocols preserve privacy against semi-honest servers, if $\mathcal{E}$ is CPA-secure and $\mathcal{G}$ is admissible.

Proof (of Theorem 13). Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a CPA-secure public-key encryption scheme with message space $\mathcal{M}$, $\mathcal{G} = \{G_n \colon \mathcal{M} \to \mathcal{M}\}_{n \in \mathbb{N}}$ a family of admissible functions over $\mathcal{M}$, and π 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 $|x_0| = |x_1|$, and a polynomial $p(\cdot)$ such that for infinitely many $\lambda \in \mathbb{N}$:

$$\begin{aligned} &\Pr[\mathcal{D}(\mathsf{view}^{\pi}_{\mathsf{Srv}}(x_1, \bot, \lambda)) = 1] \\ -&\Pr[\mathcal{D}(\mathsf{view}^{\pi}_{\mathsf{Srv}}(x_0, \bot, \lambda)) = 1] \geq \frac{1}{p(\lambda)} \end{aligned} \tag{26}$$

We show below that given $\mathcal{D}$ we can construct an adversary $\mathcal{A}$ that violate the CPA security of $\mathcal{E}$.

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

1. Upon receiving pk output x_0, x_1 (as computed by $\mathcal{D}$).
2. Upon receiving $\mathsf{Enc}_{pk}(x_b)$ behave exactly as Srv behaves while executing π upon receiving $\mathbf{c_x}$ and pk from Clnt, except that every message $(\mathbf{e}, n)$ (where $\mathbf{e}$ is an encryption and $n \in \mathbb{N}$) sent from Srv to Clnt is answered by $\mathcal{A}$ as follows: $\mathcal{A}$ samples uniformly at random m from the domain of G_n, computes $\mathbf{e}' \leftarrow \mathsf{Enc}_{pk}(G_n(m))$, and behaves as Srv upon receiving $\mathbf{e}'$ as the response from Clnt.
3. Run the distinguisher $\mathcal{D}$ on $\mathsf{view}_{\mathsf{Srv}}$ (Srv's view in $\mathcal{A}$ during step 2) and output whatever $\mathcal{D}$ outputs.

The adversary $\mathcal{A}$ is ppt due to the admissibility of $\mathcal{G}$ and Srv and $\mathcal{D}$ being ppt. Note that π is almost perfectly simulated except that the queries to Clnt are simulated using encryption of the image of G_n on a randomly sampled elements in its domain. Let π' denote this variant of π that is simulated by $\mathcal{A}$, namely π' is a protocol identical to π except that each query $(\mathbf{e}, n)$ to Clnt is answered by the encryption of $G_n(m)$ for a randomly sampled m from the domain of G_n. We denote by $\mathsf{view}^{\mathsf{EXP}^{cpa}}_{\mathsf{Srv}}(x_b, \bot, \lambda)$ the view of Srv, simulated by $\mathcal{A}$, in the execution of $\mathsf{EXP}^{cpa}_{\mathcal{A},\mathcal{E}}$ with bit b being selected by the challenger. By definition of π' it holds that for every

$b \in \{0,1\}$,

$$\begin{aligned} &\Pr[\mathcal{D}(\mathsf{view}^{\pi'}_{\mathsf{Srv}}(x_b, \perp, \lambda)) = 1] \\ &= \Pr[\mathcal{D}(\mathsf{view}^{\mathsf{EXP}^{cpa}}_{\mathsf{Srv}}(x_b, \perp, \lambda)) = 1] \end{aligned} \tag{27}$$

Furthermore, the CPA security of $\mathcal{E}$ and cleartext computability of π guarantees (as shown in Lemma 7 below) that the server's view in π and π' is computationally indistinguishable. In particular, for every $x \in \mathsf{A}$

$$\begin{aligned} &\Pr[\mathcal{D}(\mathsf{view}^{\pi'}_{\mathsf{Srv}}(x, \perp, \lambda)) = 1] \\ &- \Pr[\mathcal{D}(\mathsf{view}^{\pi}_{\mathsf{Srv}}(x, \perp, \lambda)) = 1] \leq \mathsf{neg}(\lambda)\ . \end{aligned} \tag{28}$$

Putting Equation 28 together Lemma 7 and Equations 26-27 it follows that

$$\begin{aligned} &\Pr[\mathcal{D}(\mathsf{view}^{\mathsf{EXP}^{cpa}}_{\mathsf{Srv}}(x_1, \perp, \lambda)) = 1] \\ &- \Pr[\mathcal{D}(\mathsf{view}^{\mathsf{EXP}^{cpa}}_{\mathsf{Srv}}(x_0, \perp, \lambda)) = 1] \geq \frac{1}{p(\lambda)} - \mathsf{neg}(\lambda). \end{aligned} \tag{29}$$

Therefore, we obtain that:

$$\begin{aligned} &\Pr[\mathsf{EXP}^{cpa}_{\mathcal{A},\mathcal{E}}(\lambda) = 1] \\ &= \frac{1}{2} \cdot \big(\Pr[\mathsf{EXP}^{cpa}_{\mathcal{A},\mathcal{E}}(\lambda) = 1 | b = 1] + \Pr[\mathsf{EXP}^{cpa}_{\mathcal{A},\mathcal{E}}(\lambda) = 1 | b = 0] \big) \\ &= \frac{1}{2} \cdot \Pr[\mathcal{D}(\mathsf{view}^{\mathsf{EXP}^{cpa}}_{\mathsf{Srv}}(x_1, \perp, \lambda)) = 1] \\ &+ \frac{1}{2} \cdot \Pr[\mathcal{D}(\mathsf{view}^{\mathsf{EXP}^{cpa}}_{\mathsf{Srv}}(x_0, \perp, \lambda)) = 0] \\ &= \frac{1}{2} + \frac{1}{2} \Big(\Pr[\mathcal{D}(\mathsf{view}^{\mathsf{EXP}^{cpa}}_{\mathsf{Srv}}(x_1, \perp, \lambda)) = 1] - \Pr[\mathcal{D}(\mathsf{view}^{\mathsf{EXP}^{cpa}}_{\mathsf{Srv}}(x_0, \perp, \lambda)) = 1] \Big) \\ &\geq \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{p(\lambda)} - \mathsf{neg}(\lambda) \end{aligned}$$

where the last inequality follows from Equation 29. Combining this with $\mathcal{A}$ being ppt we derive a contradiction to $\mathcal{E}$ being CPA secure. This concludes the proof. □

Let $\pi' = \langle \mathsf{CInt}', \mathsf{Srv} \rangle$ be as defined in the proof of Theorem 13, i.e., it is identical to $\pi = \langle \mathsf{CInt}, \mathsf{Srv} \rangle$ except that CInt', upon receiving server's

queries $(\mathbf{e}, n)$, instead of responding as in step 2 in Definition 5, responds by sending the encryption of $G_n(m)$ for a uniformly random message m from the domain of G_n. We show that the server is indifferent to the correctness of answers it receives from the client in the sense that its view in π and π' is indistinguishable.

Lemma 7. *Let $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a CPA-secure public-key encryption scheme with a message space $\mathcal{M}$. Let $\mathcal{G} = \{G_n \colon \mathcal{M} \to \mathcal{M}\}_{n \in \mathbb{N}}$ be a family of admissible functions. If π is a cleartext computable $(\mathcal{E}, \mathcal{G})$-aided outsourcing protocol for $F \colon \mathsf{A} \to \mathsf{B}$, then for every efficiently samplable $x \in \mathsf{A}$, and all $\lambda \in \mathbb{N}$ the following holds:*

$$\mathit{view}^{\pi'}_{\mathsf{Srv}}(x, \perp, \lambda) \approx_c \mathit{view}^{\pi}_{\mathsf{Srv}}(x, \perp, \lambda)$$

Proof. Assume by contradiction that Lemma 7 does not hold. That is, there exists a ppt distinguisher $\mathcal{D}$ that chooses $x \in \mathsf{A}$ and a polynomial $p(\cdot)$ such that for infinitely many $\lambda \in \mathbb{N}$:

$$\begin{aligned} &\Pr[\mathcal{D}(\mathsf{view}^{\pi'}_{\mathsf{Srv}}(x, \perp, \lambda)) = 1] \\ &- \Pr[\mathcal{D}(\mathsf{view}^{\pi}_{\mathsf{Srv}}(x, \perp, \lambda)) = 1] \geq \frac{1}{p(\lambda)}. \end{aligned} \tag{30}$$

We define a series of hybrid executions that gradually move between $\pi = \langle \mathsf{Clnt}, \mathsf{Srv} \rangle$ execution (where Clnt responds with $\mathsf{Enc}_{pk}(G_n(\mathsf{Dec}_{sk}(\mathbf{e})))$) to $\pi' = \langle \mathsf{Clnt}', \mathsf{Srv} \rangle$ execution (where Clnt' responds with an encryption of the image of G_n on a random message). Let q denote the number of queries made to Clnt in π. We define $q + 1$ hybrids as follows:

Hybrid H_0 is defined as the execution of $\langle \mathsf{Clnt}, \mathsf{Srv} \rangle$.

Hybrid H_j $(j = 1, \ldots, q)$ is similar to H_0 except that the last j queries to Clnt, each query $(\mathbf{e}, n)$ is answered by sampling a uniformly random m in the domain of G_n and responding with $\mathsf{Enc}_{pk}(G_n(m))$ (instead of sending $\mathsf{Enc}_{pk}(G_n(\mathsf{Dec}_{sk}(\mathbf{e})))$ as in Definition 5, Step 2).

Note that in each pair of adjacent hybrids H_{j-1} and H_j for $j \in [q]$ the difference is that in H_j the $(q + 1 - j)$'th query is answered using $G_n(m)$ for a random m instead of $\mathsf{Dec}_{sk}(\mathbf{e})$.

Denote by $\mathsf{view}^{\mathsf{H}_j}_{\mathsf{Srv}}(x, \perp, \lambda)$ the view of Srv in the hybrid H_j.

By the hybrid argument it follows from Equation 30 that there exists $j \in [q]$ such that:

$$\begin{aligned} &\Pr[\mathcal{D}(\mathsf{view}^{\mathsf{H}_j}_{\mathsf{Srv}}(x, \perp, \lambda)) = 1] \\ &- \Pr[\mathcal{D}(\mathsf{view}^{\mathsf{H}_{j-1}}_{\mathsf{Srv}}(x, \perp, \lambda)) = 1] \geq \frac{1}{q} \cdot \frac{1}{p(\lambda)} \end{aligned} \tag{31}$$

We show that Equation 31 contradicts $\mathcal{E}$ being CPA secure. That is, we construct an adversary $\mathcal{A}$ that communicates with the challenger Chal in the CPA indistinguishability experiment $\mathsf{EXP}^{cpa}_{\mathcal{A},\mathcal{E}}$ and wins with a non-negligible advantage over half. Concretely, $\mathcal{A}$ participates in $\mathsf{EXP}^{cpa}_{\mathcal{A},\mathcal{E}}$ as follows:

1. $\mathcal{A}$ computes the client's cleartext response $\mathsf{clear\text{-}res}_{\mathsf{Srv}}(x, r, \lambda) = (y_1, \ldots, y_q)$ (using the efficiently computable function h from Definition 15).
2. Upon receiving pk from Chal, $\mathcal{A}$ computes $\mathbf{c_x} \leftarrow \mathsf{Enc}_{pk}(x)$, samples a random tape r for Srv, and executes Srv with randomness r on $(\mathbf{c_x}, pk)$ while answering each query of Srv as follows:
   (a) For the first $q-j$ queries of Srv, $\mathcal{A}$ encrypts under pk the responses $y_1, \ldots, y_{q-j}$ associated with these queries, and sends the resulting ciphertexts to Srv.
   (b) For the $(q-j+1)$'th query of Srv, denoted $(\mathbf{e}, n)$, $\mathcal{A}$ proceeds as follows:
      i. $\mathcal{A}$ sets $m_0 = y_{q-j+1}$, samples uniformly random m_1 from the domain of G_n, and sends m_0 and $G_n(m_1)$ to Chal.
      ii. Upon receiving from Chal the challenge ciphertext $c \leftarrow \mathsf{Enc}_{pk}(m_b)$ for uniformly random $b \leftarrow \{0,1\}$, $\mathcal{A}$ forwards this ciphertext c to Srv.
   (c) For the rest of the queries $(\mathbf{e}', n')$, $\mathcal{A}$ samples uniformly random m in the domain of $G_{n'}$, and sends $\mathsf{Enc}_{pk}(G_{n'}(m))$ to Srv.
3. $\mathcal{A}$ executes the distinguisher $\mathcal{D}$ on the view of Srv during the execution of Step 2 above, denoted $\mathsf{view}_{\mathsf{Srv}}$, and outputs whatever $\mathcal{D}$ outputs.

We note that if $b = 0$, then the challenge ciphertext c is the encryption of y_{q-j+1} and since π is cleartext computable we get that $\mathsf{view}_{\mathsf{Srv}}$ is exactly as in H_{j-1} and otherwise as in H_j. Therefore, we obtain that

$$
\begin{aligned}
&\Pr[\mathsf{EXP}^{cpa}_{\mathcal{A},\mathcal{E}}(\lambda) = 1] \\
&= \frac{1}{2} \cdot \left(\Pr[\mathsf{EXP}^{cpa}_{\mathcal{A},\mathcal{E}}(\lambda) = 1 | b = 1] + \Pr[\mathsf{EXP}^{cpa}_{\mathcal{A},\mathcal{E}}(\lambda) = 1 | b = 0] \right) \\
&= \frac{1}{2} + \frac{1}{2} \left(\Pr[\mathcal{D}(\mathsf{view}^{\mathsf{H}_j}_{\mathsf{Srv}}(x, \perp, \lambda)) = 1] - \Pr[\mathcal{D}(\mathsf{view}^{\mathsf{H}_{j-1}}_{\mathsf{Srv}}(x, \perp, \lambda)) = 1] \right) \\
&\geq \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{q} \cdot \frac{1}{p(\lambda)}
\end{aligned}
\tag{32}
$$

– a contradiction to the CPA-security of $\mathcal{E}$; this concludes the proof. □