On the Privacy of Protocols based on CPA-Secure Homomorphic Encryption

Adi Akavia[1][0000−0003−0853−3576]⋆ and Margarita Vald[2]

[1] University of Haifa, Israel adi.akavia@gmail.com
[2] Tel Aviv University and Intuit Inc., Israel margarita.vald@cs.tau.ac.il

Abstract. Li and Micciancio (Eurocrypto 2021) shattered a widespread misconception regarding the security of protocols based on CPA-secure homomorphic encryption (HE). They showed an attack breaking security of HE-based protocols provided that the protocol employs an HE scheme for *approximate numbers*, like CKKS, and the adversary sees *decrypted* ciphertexts. However, their attack fails when employing exact HE schemes, like BGV, or denying access to decrypted data.

We show that the Li-Micciancio attack is only the tip of the iceberg:

1. We exhibit an input-recovery attack completely breaking the privacy of a wide and natural family of HE-based protocols, including protocols using only *exact* HE-schemes and with an adversary *exposed solely to encrypted data*. This proves that CPA-security is insufficient to ensure privacy in a much broader context than previously known.
2. To address the threat exhibited by our attack we introduce sufficient conditions, on either the encryption scheme or the protocol, that do guarantee privacy: (a) Every HE scheme with a sanitization algorithm (e.g., BGV and FHEW) can be transformed into a "sanitized" scheme so that protocols instantiated with it preserve privacy against malicious adversaries. (b) Moreover, we characterize a natural sub-family of these protocols for which CPA-security does suffice to guarantee privacy, albeit against semi-honest adversaries.

To prove (2a) we define a notion of circuit-privacy$^+$ that lies between semi-honest and malicious circuit-privacy and realize it from existing schemes; this may be of independent interest.

Keywords: homomorphic encryption · CPA-security · protocols · attack

1 Introduction

Background. Homomorphic encryption (HE) supports computing over encrypted data without access to the secret key. HE schemes are typically *exact* in the sense that decrypting results in the exact same value as produced by computing on cleartext values [29,19,30,9,10,18,11,16,13]; an

⋆ The work was supported in part by the Israel Science Foundation grant 3380/19, and the Israel National Cyber Directorate via Haifa, BIU and Tel-Aviv Cyber Centers.

exception is the CKKS [12] *approximate* scheme that adds noise to the underlying cleartext message so that decryption returns a close value, but not the exact one. HE-based protocols are popular for privacy related tasks such as secure outsourcing of computation and private information retrieval; and the golden standard for securing HE-based protocols is utilizing HE schemes that are CPA-secure.

Li and Micciancio [26] recently shattered the misconception that CPA-security of the HE scheme suffices to guarantee security for HE-based protocols. They showed an attack breaking the security of such protocols under the following two conditions:

(a) The protocol employs an approximate HE scheme like CKKS, and
(b) The adversary has access to decrypted ciphertexts.

Essentially, their attack shows that the noise introduced by CKKS and exposed after decryption reveals information that can be leveraged to recover the secret key, despite the CPA-security guarantee of the encryption scheme. Importantly, their attack is valid only for the toxic combination of employing approximate HE schemes together with exposure to decrypted ciphertexts. This attack had a major impact; in particular, all libraries implementing approximate HE schemes either introduced heuristic measures for mitigating the attack [2,3], or revised their security guidelines to forbid exposing decryptions to untrusted entities [1]. For exact schemes, in contrast, Li and Micciancio have proved that their attack does not apply [26]. In view of the Li-Micciancio attack we ask:

> *might CPA-security fail to guarantee security also in protocols employing exact HE schemes and never exposing decryptions?*

Our contribution. In this work we show that the insufficiency of CPA-security expands much farther beyond the combination of approximate HE schemes with exposed decryptions. We demonstrate this insufficiency by exhibiting a new input-recovery attack breaking the security of a wide and natural family of HE-based protocols, including protocols that use only exact HE schemes and where the adversary sees only encrypted data. We then address the security gap indicated by our attack by identifying natural conditions on schemes or protocols and proving they suffice for securing protocols in this family.

The protocols we address are HE-based secure outsourcing 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 call such protocols *client-aided protocols*. Importantly, in client-aided protocols the server sees only encrypted data. The security goal for such protocols is to guarantee privacy for the client's input against the server (see Definition 5 adapted from Definition 2.6.2 in [22]).[3] Client-aided protocols are common; for example, the client can provide a re-encryption service to avoid costly bootstrapping at the server's side by setting G to be the identity function [31]; likewise, the client may compute comparison [7], minimum [4,5], solving linear equations [20,6], ReLU and Max-Pooling [24], and so forth, where the client's computation may be on masked data.

Our input-recovery attack completely breaks the security of client-aided protocols, and can be mounted on all client-aided protocols regardless of whether they use an approximate or exact CPA-secure HE scheme. We note that there is no contradiction between our attack and the fact that the underlying HE scheme is CPA-secure: our attack proves the insufficiency of CPA-security to guarantee privacy.

Theorem 1 (attack, informal). *There exists CPA-secure HE schemes so that for all client-aided protocols instantiated with such schemes, there is an attack by the server that recovers the client's input.*

To address the threat exhibited by our attack we rigorously study the security of client-aided protocols introducing two natural conditions – one for schemes and the other for protocols – and proving that if either condition holds then privacy is guaranteed.

A sufficient condition on the encryption scheme. We prove that any HE scheme with a sanitization algorithm, e.g., BGV [10] and FHEW [16], can be transformed into a "sanitized" scheme so that all client-aided protocols instantiated with the sanitized scheme preserve privacy against malicious servers. A sanitization algorithm [17] re-randomizes ciphertexts to make them statistically close to other sanitized ciphertexts decrypting to the same value; our sanitized scheme applies the sanitization algorithm on the ciphertexts processed by the encryption scheme.

[3] The client-server terminology is used for convenience, yet the protocol can involve any two parties, one owning the secret key and the other computing over encrypted data.

Theorem 2 (privacy against malicious servers, informal). *Client-aided protocols instantiated with a sanitized CPA-secure encryption scheme preserve privacy against malicious servers.*

A sufficient condition on the protocol. We formalize a property satisfied by natural client-aided protocols, and prove that for such protocols, CPA-security of the encryption scheme implies the protocol preserves privacy, albeit only against semi-honest servers. The property we define (named: *cleartext computability*) is 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. We prove that for cleartext computable protocols, CPA-security guarantees privacy.

Theorem 3 (privacy against semi-honest servers, informal). *Cleartext-computable client-aided protocols instantiated with a CPA-secure encryption scheme preserve privacy against semi-honest servers.*

Our techniques. We 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 $\mathbf{e}' \leftarrow \mathsf{Enc}_{pk}(G(\mathsf{Dec}_{sk}(\mathbf{e})))$ query, where $\mathbf{e}$ is a concatenation of the client's encrypted input with a special "trapdoor" cipephertext 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.

To prove Theorem 2 we introduce an enhanced security notion for encryption schemes, named funcCPA, prove that sanitized schemes are funcCPA-secure, and that funcCPA-security of the scheme implies, for any protocol instantiated with the scheme, privacy against malicious servers. The definition of funcCPA 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 $\mathcal{G}$. Our attack, together with our result that funcCPA guarantees privacy, proves that funcCPA is strictly stronger than CPA. To construct a funcCPA-secure scheme, we first define the notion of circuit-privacy$^+$ that lies between semi-honest and malicious circuit privacy in allowing maliciously formed ciphertexts but requiring honestly generated keys. We then show how to transform any CPA-secure scheme that has a

sanitization algorithm into a CPA-secure circuit-private$^+$ scheme. Finally we prove that CPA-secure circuit-private$^+$ schemes are funcCPA-secure.

Paper organization. Preliminary definitions are given in Section 2; our attack in Section 3; our result on funcCPA schemes in Section 4, and on cleartext computable protocols in Section 5; we conclude in Section 6.

2 Preliminaries

In this section we specify standard terminology, notations and definitions used throughout this paper, including public key encryption and CPA-security, homomorphic encryption, sanitization algorithm and privacy-preserving protocols.

2.1 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 *random variable* A is a function from a finite set S to the non-negative reals with the property that $\sum_{s \in S} A(s) = 1$. 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 $n \in \mathbb{N}$. Two probability ensembles $X = \{X(a,n)\}_{a \in \{0,1\}^*, n \in \mathbb{N}}$ and $Y = \{Y(a,n)\}_{a \in \{0,1\}^*, n \in \mathbb{N}}$ 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 $n \in \mathbb{N}$,

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

The *statistical distance* of two probability ensembles $X = \{X(a,n)\}_{a \in \{0,1\}^*, n \in \mathbb{N}}$ and $Y = \{Y(a,n)\}_{a \in \{0,1\}^*, n \in \mathbb{N}}$ is defined by

$$\Delta(X,Y) = \frac{1}{2} \sum_{s \in S} |\Pr[s \in X] - \Pr[s \in Y]|.$$

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 [21], Definition 2.2.1.

2.2 CPA-Secure Public Key Encryption

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

Definition 1 (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 $m \in \mathcal{M}$,*

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

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

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 $\mathbf{c} \leftarrow \mathsf{Enc}_{pk}(x_b)$ 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 2 (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*.*

A PKE $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ with message space $\mathcal{M}$ has 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 [25].

Theorem 4 ([25] theorem 10.10). *If a public-key encryption scheme is CPA-secure, then it has indistinguishable multiple encryptions security.*

2.3 Homomorphic Encryption and Sanitization

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 [10]).

Definition 3 (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.*

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 [19], BGV [10], and FHEW [16].[4]

Definition 4 (sanitization algorithm [17]). *A sanitization algorithm for a homomorphic public-key encryption scheme $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$, denoted $\mathsf{Sanitize}$, 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.* $\mathsf{Dec}_{sk}(c) = \mathsf{Dec}_{sk}(c')$*:*

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

2.4 Privacy-Preserving Two-Party Protocols

The protocols considered in this work involve two-parties, client and server, denoted by Clnt and Srv respectively, where the client has input and output, the server has no input and no output, and both receive the security parameter λ. The client and server interact in an interactive protocol denoted by $\pi = \langle \mathsf{Clnt}, \mathsf{Srv} \rangle$. The server's view in an execution of π, on client's input x, no server's input (denoted by $\perp$), and security parameter λ, is a random variable $\mathsf{view}^\pi_{\mathsf{Srv}}(x, \perp, \lambda)$ capturing what the server has learned, and defined by

$$\mathsf{view}^\pi_{\mathsf{Srv}}(x, \perp, \lambda) = (r, m_1, \ldots, m_t)$$

[4] We conjecture that [17] can be extended to newer schemes, published following their paper, including TFHE [13] and CKKS [12]; this is beyond the scope of this work.

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 [22] Definition 2.6.2 Part 2).[5]

Definition 5 (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 there exists a negligible function* $\mathsf{neg}(\cdot)$ *such that for every* $\lambda \in \mathbb{N}$, *every* ppt *distinguisher* $\mathcal{D}$ *that chooses* $x_0, x_1 \in \mathsf{A}$ *s.t.* $|x_0| = |x_1|$, *and every* ppt *server* Srv^* *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 5 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*.

3 CPA-Security Does Not Imply Privacy

In this section we show that CPA-security is insufficient for guaranteeing privacy for HE-based protocols (cf. Theorem 1).

We demonstrate the insufficiency of CPA-security by exhibiting an attack applicable on a wide and natural family of protocols: client-aided protocols. Importantly, in these protocols the server sees only encrypted data, encrypted with a CPA-secure encryption, and never sees decryptions. The HE scheme may be exact, e.g., BGV/FV [10,18] rather than only an approximate scheme, e.g., CKKS [12], as in [26]. In fact, we can transform any CPA-secure encryption scheme, using a one-way function, into a CPA-secure encryption scheme for which our attack works. Our attack completely breaks the security of the protocol in the strong sense that the server is able to completely recover the client's input.

In the following we define the family of client-aided protocols in Section 3.1, and specify our attack in Section 3.2.

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

3.1 Client-Aided Protocols

In this section we formally define the family of *client-aided protocols*, or $(\mathcal{E},\mathcal{G})$*-aided 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).

Definition 6 ($(\mathcal{E},\mathcal{G})$-aided 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 protocol *if it has the following three stage structure:*

1. ***Client's input outsourcing phase (on input** $x \in \mathsf{A}$**):*** Clnt *runs $(sk, pk) \leftarrow \mathsf{Gen}(1^\lambda)$, 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 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 1 (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.

3.2 An Attack on Privacy

We specify our construction of a CPA-secure (possibly, homomorphic) encryption scheme $\mathcal{E}^f$, and show we can break every $(\mathcal{E}^f,\mathcal{G})$-aided protocol.

We show how to construct $\mathcal{E}^f$ from any CPA-secure (possibly, homomorphic) encryption scheme $\mathcal{E}$ with message space $\mathcal{M}$ of super-polynomial size and any one-way function $f \colon \mathcal{M} \to \mathcal{M}$. 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 $\mathsf{Enc}_{pk}(m^*)$. See the formal details in Figure 1 and Theorem 5.

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

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

$\underline{\mathsf{Enc}^f_{pkf}(m)}$: 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

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

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

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

$\underline{\mathsf{Eval}^f_{pkf}(C, c_1, ..., c_\ell)}$: 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 $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$ and a one-way function f over its message space $\mathcal{M}$.

Theorem 5 (properties of $\mathcal{E}^f$). *For every PKE scheme $\mathcal{E}$ and one-way function f over the message-space of $\mathcal{E}$, the scheme $\mathcal{E}^f$ (cf. Figure 1) is a PKE scheme satisfying the following. If $\mathcal{E}$ is CPA-secure, compact, and $\mathcal{C}$-homomorphic, then $\mathcal{E}^f$ is CPA-secure, compact, and $\mathcal{C} \times \mathcal{C}$-homomorphic.*[6]

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 4-5, Appendix A. □

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

Theorem 6 (CPA-security does not imply privacy). *For every PKE scheme $\mathcal{E}$ with message-space $\mathcal{M}$ and every one-way function f over $\mathcal{M}$,*

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

there exists a CPA-secure PKE scheme $\mathcal{E}^f$ so that for every family of functions $\mathcal{G} = \{G_n \colon \mathcal{M} \to \mathcal{M}\}_{n\in\mathbb{N}}$ and every $(\mathcal{E}^f, \mathcal{G})$-aided protocol there is a server's strategy that recovers the client's input.

Proof. Denote $\mathcal{E} = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$. Assume without loss of generality that $\mathcal{G}$ contains the identity function $\mathcal{I}$.[7] 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 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 $\mathbf{c}'_1 = x$. □

Remark 2. Our attack can be mounted by malicious servers on every $(\mathcal{E}^f, \mathcal{G})$-aided protocol. By semi-honest servers, the attack can be mounted on (contrived) protocols where the server's prescribed behavior includes sending a query $(\mathbf{e}, \mathcal{I})$ for $\mathbf{e} = (\mathbf{c}_1, \mathsf{Enc}_{pk}(m^*))$ as in Step 2 of our attack.

4 A Sufficient Strengthening of CPA

We present sufficient conditions on the encryption scheme that guarantee privacy for client-aided protocols against malicious servers. Specifically,

[7] In case $\mathcal{G}$ 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.

it suffices to use a sanitized CPA-secure scheme, or more generally any funcCPA-secure scheme (cf. Definition 7 and Theorem 2).

In the following we first present our strengthening of CPA-security to *function-chosen-plaintext attack* or funcCPA-security (Section 4.1); show that funcCPA-security of an encryption scheme $\mathcal{E}$ is sufficient to guarantee privacy for any $(\mathcal{E}, \mathcal{G})$-aided protocol (Section 4.2); and present a realization of funcCPA-secure schemes from standard properties, specifically, from any CPA-secure HE that has a sanitization algorithm (Section 4.3).

4.1 funcCPA-Security: A Strengthening of CPA

In this section we define and a new security notion of public-key encryption that we name *function-chosen-plaintext attack* (funcCPA-security). The definition captures a stronger adversary than the standard CPA 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:

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 $\mathbf{c} \leftarrow \mathsf{Enc}_{pk}(x_b)$ is computed and given to $\mathcal{A}$. We call $\mathbf{c}$ the challenge ciphertext. $\mathcal{A}$ continues to have access to $\mathsf{Enc}_{pk}(\mathcal{G}(\mathsf{Dec}_{sk}(\cdot)))$ 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 7 (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*

$\mathcal{G} = \{G_n \colon \mathcal{M} \to \mathcal{M}\}_{n\in\mathbb{N}}$ *(*$\mathsf{funcCPA}$*-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* $\lambda \in \mathbb{N}$*,*

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

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.*

4.2 funcCPA implies Privacy

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

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

Proof. Informally, 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. The formal details follow.

Let π be a $(\mathcal{E}, \mathcal{G})$-aided 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 $\lambda \in \mathbb{N}$:

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

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

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

From Equations 1 and 2 it follows that:

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

Therefore, we obtain that:

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

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

4.3 Construction of funcCPA Secure Encryption

In this section we show how to transform any CPA-secure HE scheme $\mathcal{E}$ that has a sanitization algorithm (e.g. [19,10,16]) 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 8.

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* $\mathcal{E}$*;*
– $\mathsf{Enc}^{\mathit{santz}}$ *takes a public key* pk *and a message* $m \in \mathcal{M}$ *and outputs:*

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

– $\mathsf{Eval}^{\mathit{santz}}$ *takes a public key* pk*, a circuit* $C \in \mathcal{C}$*, and ciphertexts* $c_1, \ldots, c_\ell$ *and outputs:*

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

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 $\mathsf{Sanitize}$. We show that if $\mathcal{E}$ is CPA-secure, then $\mathcal{E}^{\mathsf{santz}}$ in $\mathsf{funcCPA}$-secure.

Theorem 8 ($\mathcal{E}^{\mathsf{santz}}$ is funcCPA-secure). *If* $\mathcal{E}$ *is a* $\mathcal{C}$*-homomorphic CPA-secure PKE scheme with a sanitization algorithm, then the sanitized scheme* $\mathcal{E}^{\mathit{santz}}$ *is funcCPA-secure w.r.t.* $\mathcal{C}$*.*[8]

Proof. To prove the theorem we first enhance the definition of circuit privacy to *circuit-privacy*$^+$ (cf. Definition 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 1 below); and show that if a $\mathcal{C}$-homomorphic CPA-secure encryption scheme is circuit-privacy$^+$ for $\mathcal{C}$, then it is $\mathsf{funcCPA}$-secure w.r.t. $\mathcal{C}$ (cf. Lemma 2 below). We conclude that $\mathcal{E}^{\mathsf{santz}}$ is $\mathsf{funcCPA}$-secure w.r.t. $\mathcal{C}$. □

Circuit-privacy$^+$*.* Our definition of circuit-privacy$^+$ addresses maliciously generated ciphertexts by quantifying over all ciphertexts in the ciphertext space, rather than only over ciphertexts that were properly formed by applying the encryption algorithm on a message. Prior definitions of circuit privacy either considered the semi-honest settings where both the keys and the ciphertext are properly formed [23,19,8], or considered settings where both keys and ciphertexts may be maliciously formed [23,28,15,27]. 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* $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$ *and the random coins in* Enc *and* Eval*: For every circuit* $C \in \mathcal{C}$ *over* ℓ *inputs and*

[8] We slightly abuse notations and allow $\mathsf{funcCPA}$ with respect to a circuit family.

ciphertexts $c_1, \ldots, c_\ell$ in the ciphertext space of $\mathcal{E}$ the following distributions are statistically close:

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

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

Lemma 1 ($\mathcal{E}^{\mathsf{santz}}$ 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 $\mathsf{Sanitize}$; See the formal proof details in Appendix B.1. □

Circuit-privacy$^+$ implies funcCPA. We prove that a sufficient condition for a HE scheme to be $\mathsf{funcCPA}$-secure is that it is CPA-secure and circuit-private$^+$. We remark that Lemma 2 holds even if the schemes satisfies only a weaker notion of circuit-privacy$^+$ where we require only computational indistinguishability rather than statistical.

Lemma 2 (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 proof idea is to carefully replace $\mathsf{Enc}_{pk}(\mathcal{G}(\mathsf{Dec}_{sk}(\cdot)))$ oracle calls with Eval operations. The formal details follow.

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

- Upon receiving pk from challenger, forward it to $\mathcal{A}$.
- Answer queries $(\mathbf{e}, n)$ to $\mathsf{Enc}_{pk}(\mathcal{C}(\mathsf{Dec}_{sk}(\cdot)))$ by $\mathbf{e}' \leftarrow \mathsf{Eval}_{pk}\left(C_n, \mathbf{e}\right)$.
- 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,

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

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

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

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

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

As required. □

5 Sufficiency of CPA for Cleartext Computable Protocols

We define a natural property for $(\mathcal{E},\mathcal{G})$-aided protocols (called *cleartext computable*), and show that for protocols satisfying this property, CPA-security guarantees privacy against semi-honest servers (cf. Theorem 3).

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 protocol (cf. Definition 6). 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 10 (cleartext computable). *An $(\mathcal{E}, \mathcal{G})$-aided 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 protocols, CPA-security of $\mathcal{E}$ implies that the protocol preserves privacy against semi-honest servers.

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

Proof. 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 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 $\lambda \in \mathbb{N}$:

$$\begin{aligned} &\Pr[\mathcal{D}(\mathsf{view}^{\pi}_{\mathsf{Srv}}(x_1, \bot, \lambda)) = 1] \\ &- \Pr[\mathcal{D}(\mathsf{view}^{\pi}_{\mathsf{Srv}}(x_0, \bot, \lambda)) = 1] \geq p(\lambda) \end{aligned} \tag{5}$$

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 $\mathsf{view}_{\mathsf{Srv}}^{\mathsf{EXP}^{cpa}}(x_b, \perp, \lambda)$ 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 $b \in \{0, 1\}$,

$$\begin{aligned} &\Pr[\mathcal{D}(\mathsf{view}_{\mathsf{Srv}}^{\pi'}(x_b, \perp, \lambda)) = 1] \\ &= \Pr[\mathcal{D}(\mathsf{view}_{\mathsf{Srv}}^{\mathsf{EXP}^{cpa}}(x_b, \perp, \lambda)) = 1] \end{aligned} \tag{6}$$

Furthermore, the CPA security of $\mathcal{E}$ and cleartext computability of π guarantees (as shown in Lemma 3 below) that the server's view in π and π' is computationally indistinguishable. In particular, for every $x \in \mathsf{A}$

$$\begin{aligned} &\Pr[\mathcal{D}(\mathsf{view}_{\mathsf{Srv}}^{\pi'}(x, \perp, \lambda)) = 1] \\ &- \Pr[\mathcal{D}(\mathsf{view}_{\mathsf{Srv}}^{\pi}(x, \perp, \lambda)) = 1] \leq \mathsf{neg}(\lambda) \ . \end{aligned} \tag{7}$$

Putting Equation 7 together Lemma 3 and Equations 5-6 it follows that

$$\begin{aligned} &\Pr[\mathcal{D}(\mathsf{view}_{\mathsf{Srv}}^{\mathsf{EXP}^{cpa}}(x_1, \perp, \lambda)) = 1] \\ &- \Pr[\mathcal{D}(\mathsf{view}_{\mathsf{Srv}}^{\mathsf{EXP}^{cpa}}(x_0, \perp, \lambda)) = 1] \geq p(\lambda) - \mathsf{neg}(\lambda). \end{aligned} \tag{8}$$

Therefore, we obtain that:

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

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

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

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

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

where the last inequality follows from Equation 8. Combining this with $\mathcal{A}$ being ppt we derive a contradiction to $\mathcal{E}$ being CPA secure. This concludes the proof. □

Let $\pi' = \langle \mathsf{Clnt}', \mathsf{Srv}\rangle$ be as defined in the proof of Theorem 9, i.e., it is identical to $\pi = \langle \mathsf{Clnt}, \mathsf{Srv}\rangle$ except that Clnt', upon receiving server's queries (e, n), instead of responding as in Step 2 in Figure 6, 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 3. *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 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 3 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 $\lambda\in\mathbb{N}$:

$$\begin{aligned}&\Pr[\mathcal{D}(\mathsf{view}^{\pi'}_{\mathsf{Srv}}(x,\perp,\lambda))=1]\\ -&\Pr[\mathcal{D}(\mathsf{view}^{\pi}_{\mathsf{Srv}}(x,\perp,\lambda))=1]\geq p(\lambda)\ .\end{aligned} \tag{9}$$

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 Figure 6, 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 9 that there exists $j \in [q]$ such that:

$$\begin{aligned} &\Pr[\mathcal{D}(\mathsf{view}^{\mathsf{H}_j}_{\mathsf{Srv}}(x, \perp, \lambda)) = 1] \\ &- \Pr[\mathcal{D}(\mathsf{view}^{\mathsf{H}_{j-1}}_{\mathsf{Srv}}(x, \perp, \lambda)) = 1] \geq \frac{p(\lambda)}{q} \end{aligned} \tag{10}$$

We show that Equation 10 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 10).
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

$$\begin{aligned}
&\Pr[\mathsf{EXP}^{cpa}_{\mathcal{A},\mathcal{E}}(\lambda) = 1] \\
&= \frac{1}{2} \cdot \left(\Pr[\mathsf{EXP}^{cpa}_{\mathcal{A},\mathcal{E}}(\lambda) = 1 | b = 1] + \Pr[\mathsf{EXP}^{cpa}_{\mathcal{A},\mathcal{E}}(\lambda) = 1 | b = 0] \right) \\
&= \frac{1}{2} + \frac{1}{2} \left(\Pr[\mathcal{D}(\mathsf{view}^{\mathsf{H}_j}_{\mathsf{Srv}}(x, \perp, \lambda)) = 1] - \Pr[\mathcal{D}(\mathsf{view}^{\mathsf{H}_{j-1}}_{\mathsf{Srv}}(x, \perp, \lambda)) = 1] \right) \\
&\geq \frac{1}{2} + \frac{1}{2} \cdot \frac{p(\lambda)}{q}
\end{aligned} \tag{11}$$

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

6 Conclusions

This work proves that CPA-security does not guarantee privacy for HE-based client-aided protocols in a much broader context than previously known, and presents (stronger) sufficient requirements, for schemes or protocols, that do guarantee privacy. (1) We present a new attack proving that CPA-security of the underlying HE-scheme does not imply privacy, even for protocols employing exact HE-scheme and exposing only encrypted data. (2) We prove that instantiating client-aided protocols with sanitized CPA-secure schemes guarantees privacy against malicious servers; and (3) for cleartext computable protocols, instantiating them with CPA-secure schemes guarantees privacy against semi-honest servers. Our attack cautions against reliance on CPA-security to guarantee privacy in client-aided protocols, whereas our security proofs provide a easy-to-use machinery for proving that client-aided protocols preserve privacy.

To prove (2) we introduce the notion of $\mathsf{funcCPA}$-security, and prove it guarantees privacy against malicious adversaries and holds for sanitized CPA-secure scheme. An open problem is to construct $\mathsf{funcCPA}$-secure

schemes attaining competitive efficiency with state-of-the-art CPA-secure schemes. Two possible approaches follow. (i) Realize funcCPA-security from other standard properties, beyond sanitization, or directly prove it holds for existing un-sanitized schemes. (ii) Devise sanitization algorithms for newer and faster schemes published subsequently to [17], e.g. TFHE [14] and CKKS [12].

References

1. Microsoft SEAL security guidelines, 2020.
2. HElib guidelines for security of approximate-numbers homomorphic encryption, 2021.
3. PALISADE security guidelines for ckks, 2021.
4. 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.
5. A. Akavia, M. Leibovich, Y. S. Resheff, R. Ron, M. Shahar, and M. Vald. Privacy-preserving decision trees training and prediction. In *Machine Learning and Knowledge Discovery in Databases*, pages 145–161. Springer International Publishing, 2021.
6. A. Akavia, 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.
7. R. Bost, R. A. Popa, S. Tu, and S. Goldwasser. Machine learning classification over encrypted data. In *NDSS*, volume 4324, page 4325, 2015.
8. F. Bourse, R. Del Pino, M. Minelli, and H. Wee. Fhe circuit privacy almost for free. In *Annual International Cryptology Conference*, pages 62–89. Springer, 2016.
9. 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.
10. 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.
11. Z. Brakerski and V. Vaikuntanathan. Efficient fully homomorphic encryption from (standard) lwe. volume 2011, pages 97–106, 10 2011.
12. 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.
13. I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. TFHE: fast fully homomorphic encryption over the torus. *J. Cryptol.*, 33(1):34–91, 2020.

14. I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène. Tfhe: Fast fully homomorphic encryption over the torus. *Journal of Cryptology*, 2019.
15. W. Chongchitmate and R. Ostrovsky. Circuit-private multi-key fhe. In *PKC (2)*, pages 241–270. Springer, 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 M. Fischlin and J.-S. Coron, editors, *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 et al. Fully homomorphic encryption using ideal lattices. In *Stoc*, volume 9, pages 169–178, 2009.
20. I. Giacomelli, S. Jha, M. Joye, C. D. Page, and K. Yoon. Privacy-preserving ridge regression with only linearly-homomorphic encryption. In B. Preneel and F. Vercauteren, editors, *Applied Cryptography and Network Security - 16th International Conference, ACNS 2018, Leuven, Belgium, July 2-4, 2018, Proceedings*, volume 10892 of *Lecture Notes in Computer Science*, pages 243–261. Springer, 2018.
21. O. Goldreich. *The Foundations of Cryptography - Volume 1, Basic Techniques*. Cambridge University Press, 2001.
22. C. Hazay and Y. Lindell. *Efficient Secure Two-Party Protocols: Techniques and Constructions*. Springer-Verlag, Berlin, Heidelberg, 1st edition, 2010.
23. Y. Ishai and A. Paskin. Evaluating branching programs on encrypted data. In S. P. Vadhan, editor, *Theory of Cryptography*, pages 575–594, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
24. 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.
25. J. Katz and Y. Lindell. *Introduction to Modern Cryptography (Chapman & Hall/Crc Cryptography and Network Security Series)*. Chapman & Hall/CRC, 2007.
26. B. Li and D. Micciancio. On the security of homomorphic encryption on approximate numbers. Springer-Verlag, 2021.
27. G. Malavolta. Circuit privacy for quantum fully homomorphic encryption.
28. R. Ostrovsky, A. Paskin-Cherniavsky, and B. Paskin-Cherniavsky. Maliciously circuit-private fhe. In *Annual Cryptology Conference*, pages 536–553. Springer, 2014.
29. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In J. Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, pages 223–238, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.
30. M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Monaco / French Riviera, May 30 - June 3, 2010. Proceedings*, pages 24–43, 2010.
31. W. Wang, Y. Jiang, Q. Shen, W. Huang, H. Chen, S. Wang, X. Wang, H. Tang, K. Chen, K. E. Lauter, and D. Lin. Toward scalable fully homomorphic encryption through light trusted computing assistance. *CoRR*, abs/1905.07766, 2019.

A Proof of Theorem 5

In this section we given the proof of Theorem 5, 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 4; (b) if $\mathcal{E}$ is CPA-secure then $\mathcal{E}^f$ is CPA-secure, see Lemma 5.

Lemma 4 (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

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

which concludes the proof. □

Lemma 5 (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 4 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:

$$\Pr[\mathsf{EXP}^{cpa}_{\tilde{\mathcal{A}}^f, \tilde{\mathcal{E}}^f}(\lambda) = 1] \geq \frac{1}{2} + p(\lambda). \tag{12}$$

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 12 we obtain that

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

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:

$$\Pr[\mathsf{EXP}^{cpa}_{\mathcal{A}^f, \mathcal{E}^f}(\lambda) = 1] \geq \frac{1}{2} + p(\lambda). \tag{13}$$

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,

$$\Pr[\mathsf{EXP}^{cpa}_{\tilde{\mathcal{A}}^f, \tilde{\mathcal{E}}^f}(\lambda) = 1] = \Pr[\mathsf{EXP}^{cpa}_{\mathcal{A}^f, \mathcal{E}^f}(\lambda) = 1 \text{ and } \neg E]. \tag{14}$$

Moreover,

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

where the last inequality follows from Equation 13.

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

$$|\Pr[E'] - \Pr[E]| \geq p(\lambda) \tag{15}$$

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 15, 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

$$Pr[E'] \geq p(\lambda) \tag{16}$$

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 16 that

$$\Pr[\mathcal{B}_{owf} \text{ invers } f] = \Pr[E'] \geq p(\lambda) \tag{17}$$

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. □

B Omitted Proofs from Section 4

We bring here formal proof details omitted from Section 4.

B.1 Proof of Lemma 1.

We prove Lemma 1 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 1). 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 $\mathcal{E}^{\mathsf{santz}}$,

$$\begin{aligned} &\mathsf{Enc}^{\mathsf{santz}}_{pk}\left(C\left(\mathsf{Dec}_{sk}(c_1), \cdots, \mathsf{Dec}_{sk}(c_\ell)\right)\right) \\ &= \mathsf{Sanitize}_{pk}\left(\mathsf{Enc}_{pk}\left(C\left(\mathsf{Dec}_{sk}(c_1), \ldots, \mathsf{Dec}_{sk}(c_\ell)\right)\right)\right) \end{aligned} \tag{18}$$

and

$$\begin{aligned} &\mathsf{Eval}^{\mathsf{santz}}_{pk}\left(C, c_1, \ldots, c_\ell\right) \\ &= \mathsf{Sanitize}_{pk}\left(\mathsf{Eval}_{pk}\left(C, \mathsf{Sanitize}_{pk}(c_1), \ldots, \mathsf{Sanitize}_{pk}(c_\ell)\right)\right) \end{aligned} \tag{19}$$

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 $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$:

$$\forall i \in [\ell] : \Pr[\mathsf{Dec}_{sk}(\mathsf{Enc}_{pk}(\mathsf{Dec}_{sk}(c_i))) = \mathsf{Dec}_{sk}(c_i)] \geq 1 - \mathsf{neg}(\lambda) \tag{20}$$

and

$$\Pr\left[\begin{smallmatrix}\mathsf{Dec}_{sk}\left(\mathsf{Enc}_{pk}(C(\mathsf{Dec}_{sk}(c_1),\ldots,\mathsf{Dec}_{sk}(c_\ell)))\right) \\ = C(\mathsf{Dec}_{sk}(c_1),\ldots,\mathsf{Dec}_{sk}(c_\ell))\end{smallmatrix}\right] \geq 1 - \mathsf{neg}(\lambda) \tag{21}$$

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

From Equation 20 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,

$$\begin{aligned} &\mathsf{Dec}_{sk}\left(\mathsf{Eval}_{pk}\left(C, \mathsf{Sanitize}_{pk}(c_1), \ldots, \mathsf{Sanitize}_{pk}(c_\ell)\right)\right) \\ =&\mathsf{Dec}_{sk}\left(\mathsf{Eval}_{pk}\left(C, \mathsf{Sanitize}_{pk}(\mathsf{Enc}_{pk}(\mathsf{Dec}_{sk}(c_1))), \ldots, \mathsf{Sanitize}_{pk}(\mathsf{Enc}_{pk}(\mathsf{Dec}_{sk}(c_\ell)))\right)\right) \end{aligned} \tag{22}$$

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,

$$\begin{aligned}
&\mathsf{Dec}_{sk}\left(\mathsf{Eval}_{pk}\left(C, \mathsf{Sanitize}_{pk}(\mathsf{Enc}_{pk}(\mathsf{Dec}_{sk}(c_1))), \ldots, \mathsf{Sanitize}_{pk}(\mathsf{Enc}_{pk}(\mathsf{Dec}_{sk}(c_\ell)))\right)\right) \\
=&\mathsf{Dec}_{sk}\left(\mathsf{Eval}_{pk}\left(C, (\mathsf{Enc}^{\mathsf{santz}}_{pk}(\mathsf{Dec}_{sk}(c_1))), \ldots, (\mathsf{Enc}^{\mathsf{santz}}_{pk}(\mathsf{Dec}_{sk}(c_\ell)))\right)\right) \\
=&C\left(\mathsf{Dec}_{sk}(c_1), \ldots, \mathsf{Dec}_{sk}(c_\ell)\right)
\end{aligned} \tag{23}$$

Combining Equations 21, 22, and 23 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,

$$\begin{aligned}
&\mathsf{Dec}_{sk}\left(\mathsf{Eval}_{pk}\left(C, \mathsf{Sanitize}_{pk}(c_1), \ldots, \mathsf{Sanitize}_{pk}(c_\ell)\right)\right) \\
=&\mathsf{Dec}_{sk}\left(\mathsf{Enc}_{pk}\left(C\left(\mathsf{Dec}_{sk}(c_1), \ldots, \mathsf{Dec}_{sk}(c_\ell)\right)\right)\right)
\end{aligned} \tag{24}$$

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 18-19, we obtain that $\mathcal{E}^{\mathsf{santz}}$ is circuit-private$^+$. □