Computational FHE Circuit Privacy for Free

Anamaria Costache ⬤, Lea Nürnberger ⬤, and Tjerand Silde ⬤

Department of Information Security and Communication Technology
Norwegian University of Science and Technology (NTNU), Norway
{anamaria.costache, lea.nurnberger, tjerand.silde}@ntnu.no

Abstract. Circuit privacy is an important notion in Fully Homomorphic Encryption (FHE), well-illustrated by the Machine Learning-as-a-Service scenario. A scheme is *circuit private* (first defined in Gentry's PhD Thesis) if an adversary cannot learn the circuit evaluated on a ciphertext from the computation result. In this work, we first show that the BGV FHE scheme by Brakerski, Gentry and Vaikuntanathan (ITCS'12) is computationally circuit private in a semi-honest context, and then present an extended construction to make it computationally circuit private against a malicious adversary. We achieve this *without* resorting to expensive mechanisms such as noise flooding. Instead, we argue carefully about the ciphertext and noise distributions that are encountered in BGV.

In more detail, we consider the notion of circuit privacy along four dimensions: whether the adversary is internal or external (i.e. does the adversary hold the secret key or not), and in a semi-honest and malicious setting. Our starting point is Gentry's definition, which we change from statistical to computational indistinguishability. Doing so allows us to prove that the BGV scheme is computationally circuit-private in a semi-honest setting to an external adversary *out of the box*.

We then propose a new definition by extending Gentry's definition to an internal adversary. This is appropriate since the scenario that the client is the adversary (and therefore has access to the decryption key) is a realistic one. Further, we remark that our definition is strictly stronger than Gentry's – our definition requires that a scheme be circuit private according to Gentry's definition and additionally, the distribution of the ciphertext noise in all ciphertexts to be computationally indistinguishable. Given this new definition, and using previous results of Costache, Nürnberger and Player (CT-RSA'23), we show that slight modifications to the BGV scheme will make it fulfill this new definition. Finally, we show how to extend these results to a malicious setting if we require that the client attaches proofs of well-formedness of keys and ciphertexts.

Keywords: Fully Homomorphic Encryption · Circuit Privacy · BGV

1 Introduction

Fully Homomorphic Encryption (FHE) allows one to perform arbitrary computations on encrypted data while keeping the data private. Gentry proposed the first

construction in 2009 [Gen09] and since then, many improvements and constructions have been proposed [BGV12, Bra12, CKKS17, CGGI16, CGGI17, DM14, FV12, GSW13]. One of the most popular applications is Machine-Learning-as-a-Service (MLaaS): a client would like to query a model on their data; however, they do not want to provide their data in the clear to the server that holds the model. On the other hand, the server does not want the client to learn the model. FHE provides the first part, keeping the client's data private. The second part, the requirement that the client learns nothing about the model from seeing the resulting ciphertext except for what it learns from the decrypted result, is an additional property called *circuit privacy*. This property is also fundamental outside of MLaaS scenarios; for example, it allows the construction of more efficient MPC protocols, as detailed in [Klu22].

Circuit privacy was first considered in Gentry's PhD thesis [Gen09], and has since then been the subject of active research [AGHV22, BdMW16, DD22, DS16, Klu22, KS23, OPP14].

1.1 Related Work

Different approaches for achieving circuit privacy have been proposed. The first was the noise flooding technique proposed by Gentry in his PhD Thesis [Gen09]. He achieved this by adding encryptions of 0 with large noise. The drawback of this technique is that, in order to ensure correct decryption after noise flooding, the parameters must be increased to allow for the extra noise, leading to inefficient schemes. Indeed, [KS23] show for the CKKS scheme [CKKS17] that if no special properties of the CKKS scheme are considered, and the adversary is assumed to have the secret key, the noise that needs to be added makes the scheme impractical.

The second approach to obtain circuit privacy is to use bootstrapping. Ducas and Stehlé propose a method they call ciphertext sanitization [DS16] that removes all information about operations carried out on the ciphertext, at the prize of requiring repeated executions of bootstrapping. [AGHV22] show that sanitization even leads to a circuit private scheme when the ciphertext is maliciously generated, but the keys are honestly generated. Since there are only a few schemes (TFHE [CGGI16, CGGI17], FHEW [DM15]) that have an efficient bootstrapping operation, this technique has a considerable impact on the scheme's practicality.

A third approach is to look at specific schemes and to leverage their properties. This is the approach we use in this paper. The work of [BdMW16] and [Klu22] make use of the special noise growth of third generation schemes such as TFHE and GSW [GSW13]. Through careful noise analysis, they show how to make the scheme circuit private by only adding a small amount of noise. These constructions are naturally very dependent on the specific nature of the schemes considered, and a careful analysis needs to be applied in order to ascertain how, or if, this can be carried over to other schemes. The line of work of [OPP14] and [DD22] use yet another approach. They do not consider one

particular scheme but give a general construction from building blocks. Indeed, [OPP14] shows that combining a compact non-circuit private FHE scheme with a non-compact circuit private HE scheme allows one to obtain a malicious circuit private FHE scheme. [DD22] extend this construction and make it multi-hop through an information-theoretic construction, allowing for a slightly weaker definition of circuit privacy.

All of the above works consider what we call statistical circuit privacy. In different set-ups and flavors, they require the distribution of all ciphertexts or the distribution of their noise to be statistically indistinguishable from some distribution. All except for the work of [KS23] define circuit privacy via the ciphertext distribution: a scheme is circuit private if the distribution of any ciphertext is statistically close to another pre-defined distribution. If a scheme is secure, and an adversary cannot perform decryptions, ciphertexts are the only information they have access to. Therefore, the above all assume the scenario of an external adversary. However, in a circuit privacy scenario, the adversary may be a client interacting with a server, and therefore, the adversary may hold the secret key. [KS23] look at this scenario and the additional information the adversary can learn, particularly through the noise distribution in the ciphertext.

1.2 Contributions

In this work, we consider the notion of circuit privacy along four dimensions: whether the adversary is internal or external – this refers to whether the adversary is assumed to have access to the secret key or not, respectively. We also consider the semi-honest and malicious scenarios, corresponding to whether we assume the keys and ciphertexts are honestly generated, respectively.

In this work, we study all possible scenarios: an external adversary, an adversary holding the secret key, semi-honest circuit privacy and malicious circuit privacy. We will follow a different approach from the ones detailed above. Instead of focusing on achieving statistical circuit privacy, we look at the weakest relaxation of this definition that BGV [BGV12] can achieve in its plain version.

We first show that BGV is circuit private against an external adversary if we relax from a statistical to a computational security definition. The obvious advantage of this approach is that this property can be obtained without the need to modify the scheme and therefore allows, with no efficiency loss, to obtain a version of circuit privacy that is sufficiently strong in most cases. In scenarios where computational security is sufficient, we can maintain the scheme's capabilities. This is in contrast to having to resort to noise flooding and ciphertext sanitization techniques, which have a significant negative impact on the scheme's performance. We obtain computational circuit privacy for free.

Next, we consider an adversary that holds the secret key and can observe the ciphertext noise to learn information about the circuit. By strategically using modulus switching, we show that BGV is circuit private against an adversary possessing the secret key. Here again, we do not require any noise to be added or bootstrapping to be applied. However, the scheme's capabilities are slightly reduced compared to plain BGV since we need to force modulus switching at certain

points, which leads to a loss of levels. Additionally, we must introduce stricter trade-offs between the parameters and the number of additions and multiplications by a constant that can be carried out. Here, a multiplication by a constant refers to a plaintext-ciphertext multiplication (as opposed to a ciphertext-ciphertext multiplication). We discuss this in more detail in Section 3.3.

Lastly, we show that BGV still achieves circuit privacy in both scenarios against a malicious adversary if appropriate commitments and proofs of shortness, linear relation, and multiplication are added. We informally present the definitions that we will use throughout this paper in the next subsection and will give a more detailed overview of this paper in Section 1.5.

1.3 Discussion of the results

Our main contributions can be summarised as follows. Firstly, we provide the first clear differentiation of circuit privacy along two dimensions mentioned above (interval vs external adversary, semi-honest vs malicious setting). Secondly, we show that BGV is circuit private according to a computational version of Gentry's definition (see Definition 14). Building on that, we propose a definition of circuit privacy that is semi-honest with an internal adversary. We provide the first construction of a circuit private BGV, both according to the computational variant of Gentry's definition and the new definition we propose, without resorting to expensive mechanisms such as noise flooding or repeated bootstrapping. We also provide the first malicious circuit private construction for BGV.

We remark that our proof strategy (showing the computational variant of Gentry's definition holds by showing BGV ciphertexts are RLWE samples) will never work in a statistical indistinguishability setting, as FHE relies on the computational hardness of RLWE and not statistical. In fact, it is not possible to instantiate FHE by relying on the statistical hardness of RLWE.

Relying on computational assumptions only is not a concern in practice, providing data and circuit privacy at the same security level. The only setting where this relaxation would matter is if the parameters for the instantiated BGV scheme are too weak, and the adversary would break circuit privacy without worrying about the privacy of its data. However, the server should ensure the parameters are set appropriately to guarantee the privacy of its secret circuit.

Finally, we provide an overview of different notions of circuit privacy and how those relate to one another, which may be of independent interest. For reasons of space, these can be found in Supplementary Material A.

1.4 Definitions of Circuit Privacy

The term circuit privacy can be ambiguous since many different statistical, simulation-based, and game-based definitions have been proposed over the past years, capturing slightly different aspects. We give an overview of these definitions and the relations among them in Supplemental Material A. All those definitions follow the blueprint given by Gentry's definition [Gen09], stated again in Definition 1. We show in the Supplemental Material that this definition is the

strongest if the adversary does not hold the secret key; therefore, it is the one we use as a baseline in this work.

Definition 1 ([Gen09]). *We say that a homomorphic encryption scheme $\mathcal{E} = (\mathtt{KeyGen}, \mathtt{Enc}, \mathtt{Eval}, \mathtt{Dec})$ is (statistically) circuit private for circuits in $\mathcal{C_E}$, if for any key set $(\boldsymbol{sk}, \boldsymbol{pk}, \boldsymbol{evk})$ output by $\mathtt{KeyGen}(1^\lambda)$, for any function f on plaintexts that has a corresponding circuit $C_f \in \mathcal{C_E}$ and any fixed ciphertexts $\boldsymbol{ct}_1, \ldots, \boldsymbol{ct}_k$ that are the encryption by $\mathtt{Enc}_\mathcal{E}(\cdot, \boldsymbol{pk})$ of plaintexts $m_1, \ldots, m_k$ respectively, the following holds for the distributions over the random coins in $\mathtt{Enc}_\mathcal{E}(\cdot, \boldsymbol{pk})$ and $\mathtt{Eval}_\mathcal{E}(\cdot, C_f, \boldsymbol{evk})$:*

$$\Delta(\mathtt{Enc}_\mathcal{E}(f(m_1, \ldots, m_k), \boldsymbol{pk}), \mathtt{Eval}_\mathcal{E}(\boldsymbol{ct}_1, \ldots, \boldsymbol{ct}_k, C_f, \boldsymbol{evk})) \leq negl(\lambda).$$

1.5 Non-technical overview of the paper

We proceed to describe in detail the structure of this paper. The results in this paper all stem from two key observations: first, if the inputs into an operation in BGV are Ring Learning With Error (RLWE) samples, so are the outputs; second, as observed in [CNP23a], modulus switching strips away all information about prior operations on the ciphertext.

To use the first observation, we relax Gentry's definition to a computational one. The formal definition will be given in Section 3.1; the informal definition is stated below.

Definition 2 (Informal). *Let $\mathcal{E}$ be a homomorphic encryption scheme for circuits $\mathcal{C_E}$. $\mathcal{E}$ is called computationally circuit private, if for a function f and its corresponding circuit $C_f \in \mathcal{C_E}$ for fixed ciphertexts $\boldsymbol{ct}_1, \ldots, \boldsymbol{ct}_k$ encrypting messages $m_1, \ldots, m_k$ and keys $\boldsymbol{pk}, \boldsymbol{evk}$ the distributions over the random coins in $\mathtt{Enc}(\cdot, \boldsymbol{pk})$ and $\mathtt{Eval}(\cdot, C_f, \boldsymbol{evk})$ are computationally indistinguishable. If $\mathcal{E}$ is a leveled scheme, it is said to be leveled computationally circuit private, if this definition holds when $\mathtt{Enc}(f(m_1, \ldots, m_k), \boldsymbol{pk})$ and $\mathtt{Eval}(\boldsymbol{ct}_1, \ldots, \boldsymbol{ct}_k, C_f, \boldsymbol{evk})$ are at the same level. A scheme is said to be maliciously computationally (leveled) circuit private if the above holds, even if we do not assume the ciphertexts and keys to be honestly generated.*

This definition requires any ciphertext to be computationally indistinguishable from uniform random in the ciphertext space, and we show that BGV fulfils this definition since it is based on the RLWE problem. In more detail, we show that a fresh BGV ciphertext is an RLWE sample and, by the RLWE problem, computationally indistinguishable from uniform random in the ciphertext space. Applying the admissible operations in BGV does not change this; the ciphertexts remain RLWE samples and, therefore, computationally indistinguishable from uniform random in the ciphertext space. It follows that BGV is computationally circuit private according to our new definition.

In Section 3.2, we extend the above definition also to consider the noise. That is to say, we now additionally assume that the adversary holds the secret key, i.e. we assume the adversary to be internal.

In Section 3.3, we show that we can achieve this new definition by strategically applying modulus switching. It was observed in [CNP23a] that if the noise of a ciphertext is below a certain threshold, modulus switching reduces the ciphertext noise to simply a rounding noise, independent of the circuit evaluated on the ciphertext. We show how to guarantee that the noise in any ciphertext will always be below this threshold. We do so by introducing restrictions on the scheme's parameters and the number of additions and multiplications by a constant that can be carried out. We can ensure that a final modulus switching at the end of an evaluation strips the noise of the ciphertext of all information from prior evaluations. Therefore, this variant of BGV is computationally circuit private, even if the adversary possesses the secret key. This result may at first seem contradictory to the findings in [KS23], where it is stated that CKKS can only be made circuit private against an adversary holding a secret key if a large amount of noise is added. However, our proof very strongly relies on the properties of modulus switching, which is an operation that can be found in BGV, but not in CKKS. The authors of [KS23] do not consider the re-scaling operation in CKKS, which would be the best equivalent to modulus switching in BGV. Therefore, substantial differences between the results are expected.

In Section 5, we finally show that the proofs from Section 3 carry through even when ciphertexts and public keys are assumed to be maliciously generated, as long as appropriate commitments and proofs of shortness, multiplication, and linear relations are added to the ciphertexts and keys upon generation. Therefore, BGV fulfils both definitions – assuming an internal and an external adversary – of circuit privacy in a malicious context.

2 Preliminaries

2.1 Notation

Let $v \in \mathcal{R} \times \ldots \times \mathcal{R}$ be a vector of ring elements from $\mathcal{R}$. In a slight abuse of notation, we denote by $v[i]$ the $i-$th element of v. The notation $[\cdot]_q$ denotes reduction modulo q (coefficient-wise, when applied to a polynomial). The notation $\lceil \cdot \rfloor$ denotes rounding to the nearest integer (coefficient-wise, when applied to a polynomial), while $\lceil \cdot \rfloor_t$ denotes coefficient-wise rounding to the next integer that has the same value modulo t. Unless otherwise specified, $\log$ denotes $\log_2$.

For any distribution $\mathcal{D}$ we write $x \leftarrow \mathcal{D}$ to denote the fact that x has been drawn from $\mathcal{D}$. For any set S, $x \xleftarrow{\$} S$ denotes the fact that x has been sampled uniformly at random from S.

Let f be any function on plaintexts. Then we denote by C_f the corresponding homomorphic circuit. That is, C_f denotes the adaptation of f to the homomorphic domain, which includes fixing an order of operations, and noise management operations such as modulus switching. We observe that even though we may have $\forall x : f(x) = f'(x)$ for functions f, f', we may not necessarily have $C_f = C_{f'}$.

2.2 Algebraic Background

Let $x^n + 1$ be the $2n^{\text{th}}$ cyclotomic polynomial, and consider the polynomial ring $\mathcal{R} = \mathbb{Z}[x]/(x^n + 1)$. To represent polynomials in $\mathcal{R}$ as vectors, we use the coefficient embedding. For a polynomial $a \in \mathcal{R}$, where $a = \sum_{i=0}^{n-1} a_i x^i$, the value of its coefficient embedding is the vector $(a_0, \ldots, a_{n-1})$.

We denote by $\|p\|_\infty$ the infinity norm of the coefficient embedding of p. For $a, b \in \mathcal{R}$ and for $\gamma_\mathcal{R}$ the expansion factor [LM06] of $\mathcal{R}$, it holds that

$$\|ab\|_\infty \leq \gamma_\mathcal{R} \|a\|_\infty \|b\|_\infty.$$

For an n-dimensional power of two cyclotomic ring $\mathcal{R}$ we have $\gamma_\mathcal{R} = n$.

2.3 Mathematical Background

In order to measure the distance between two distributions, we will make use of the statistical distance. It is defined as follows.

Definition 3. *Let (Ω, E) be a measurable space and let $\mathcal{D}_0$ and $\mathcal{D}_1$ be distributions defined over this space. The statistical distance $\Delta(\cdot, \cdot)$ between $\mathcal{D}_0$ and $\mathcal{D}_1$ is defined as*

$$\Delta(\mathcal{D}_0, \mathcal{D}_1) = \sup_{e \in E} |\mathcal{D}_0(e) - \mathcal{D}_1(e)|.$$

It is possible to think of Δ as being the largest distance the two distributions can have from one another for a given event E.

We make use of the characteristic function to incorporate information about the level of a ciphertext. We will use this in the noise estimates.

Definition 4. *The characteristic function on a set A is defined as follows*

$$\chi_A(x) = \begin{cases} 1 & , \text{ if } x \in A \\ 0 & , \text{ otherwise.} \end{cases}$$

2.4 The BGV FHE Scheme

We first recall the decision variant of the RLWE problem [LPR10].

Definition 5 (Definition 4 in [BGV12]). *For the security parameter λ let $\mathcal{R}$ be a power of two cyclotomic ring. Let $q = q(\lambda) \geq 2$ be an integer and let $\mathcal{R}_q = \mathcal{R}/q\mathcal{R}$. Let χ be a distribution over $\mathcal{R}$. The $RLWE_{d,\lambda,\chi}$ problem is to distinguish the following two distributions: in the first distribution (a_i, b_i) is sampled uniformly randomly from $\mathcal{R}_q^2$. In the second distribution, $s \xleftarrow{\$} \mathcal{R}_q$ is drawn uniformly and (a_i, b_i) is sampled by drawing $a_i \xleftarrow{\$} \mathcal{R}_q, e_i \leftarrow \chi$ and setting $b_i = a_i s + e_i$. The RLWE assumption is, that the $RLWE_{d,\lambda,\chi}$ problem is infeasible.*

In schemes based on the RLWE problem, s is drawn from a secret key distribution $\mathcal{S}$. Typically, this distribution is chosen to be ternary with a specified Hamming weight.

Following [CNP23a], [HS20], we define the `BGV` scheme [BGV12] as a levelled FHE scheme based on the RLWE problem [LPR10]. The ciphertext space is $\mathcal{R}_q = \mathbb{Z}_q[x]/(x^n + 1)$, where q is the ciphertext modulus. The plaintext space is $\mathcal{R}_t = \mathbb{Z}_t[x]/(x^n + 1)$, where t is the plaintext modulus. Messages and ciphertexts will be considered as polynomials in $\mathcal{R}_t$ and $\mathcal{R}_q$, respectively.

The `BGV` scheme is parametrised by the following:

- The ring dimension n
- The plaintext modulus t
- The length L of the moduli chain $Q_L \gg \ldots \gg Q_0$, where $Q_i | Q_{i+1}$ for $i \in \{0, \ldots, L-1\}$
- The decomposition base $\mathbf{D} = \{D_1^\star, \ldots, D_L^\star\}$, with $D_j^\star = \prod_{h=1}^{j-1} D_h$, where the D_h are such that $Q_i = \prod_{h=1}^{\ell} D_h$
- The decomposition parameter ℓ
- The security parameter λ
- The secret key distribution $\mathcal{S}$
- The error distribution χ

`BGV` consists of the algorithms `KeyGen`, `Encrypt`, `Decrypt`, `Add`, `AddConst`, `MultConst`, `Automorphism`, `PreMult`, `ReLinearize`, `KeySwitch` and `ModSwitch`, defined as follows.

`KeyGen(`1^λ`):` Draw $s \leftarrow \mathcal{S}$ and set $(1, s) := \mathtt{sk}$ as the secret key. Sample $a \xleftarrow{\$} \mathcal{R}_{Q_L}$ and $e \leftarrow \chi$. Set $\mathtt{pk} = (\mathtt{pk}[0], \mathtt{pk}[1]) := ([-as - te]_{Q_L}, a)$ as the public key. Let $\mathtt{sk}' = (s_0', \ldots, s_k')$ be another secret key. The evaluation keys for switching a ciphertext with respect to $\mathtt{sk}'$ to a ciphertext with respect to $\mathtt{sk} = (1, s)$ is defined as follows. For $j \in \{0, \ldots, k\}$ and $i \in \{0, \ldots, \ell\}$ sample $a_{ij} \xleftarrow{\$} \mathcal{R}_{Q_L}$ and $e_{ij} \leftarrow \chi$ and set $\mathtt{evk}_j := ([-a_{ij}s - te_{ij} + D_i^\star s_i']_{Q_L}, a_{ij})$. Return $(\mathtt{sk}, \mathtt{pk}, \mathtt{evk})$.

`Encrypt(pk,` m`):` Let $m \in \mathcal{R}_t$ be a message. Let $Q_i, i \in \{0, \ldots, L\}$ be the modulus in the moduli chain of the current level. Sample $u \leftarrow \mathcal{S}$ and $e_1, e_2 \leftarrow \chi$. Return $\mathtt{ct} = (\mathtt{ct}[0], \mathtt{ct}[1]) := ([m + \mathtt{pk}[0]u + te_1]_{Q_i}, [\mathtt{pk}[1]u + te_2]_{Q_i})$.

`Decrypt(sk,ct):` Return $m' = [[< \mathtt{ct}, \mathtt{sk} >]_{Q_i}]_t$.

`ModSwitch((ct,`Q_i`),`Q_j`):` Let $\mathtt{ct} = (\mathtt{ct}[0], \mathtt{ct}[1])$ be at level i. Return $\mathtt{ct}^{ms} := \left(\left\lfloor \frac{Q_j}{Q_i} \mathtt{ct}[0] \right\rceil_t, \left\lfloor \frac{Q_j}{Q_i} \mathtt{ct}[1] \right\rceil_t \right)$, where $\left\lfloor \frac{Q_j}{Q_i} \mathtt{ct}[i] \right\rceil_t$ denotes the rounding of the coefficients of the scaled ciphertext such that $\mathtt{ct}$ and $\mathtt{ct}^{ms}$ encrypt the same message modulo t.

$\texttt{KeySwitch}(\texttt{ct}, \texttt{sk}', \texttt{evk})$: Let $\texttt{ct}'$ be a ciphertext with respect to $\texttt{sk}' = (s_0, \ldots, s_k)$. Recall $\texttt{sk} = (1, s)$. Define $T \subseteq \{1, \ldots, k\}$ to be the set such that for $i \in T$ $s_i' = 1$ or $s_i' = s$. Define $\texttt{ct}_j'[i]$ via

$$\texttt{ct}'[i] = \sum_{j=1}^{\ell} \texttt{ct}_j'[i] D_j^{\star}.$$

For $i \in \{0, \ldots, k\} \setminus T$ compute

$$(\texttt{ct}^{(i)}[0], \texttt{ct}^{(i)}[1]) = \left[\sum_{j=1}^{\ell} (\texttt{ct}_j'[i]\texttt{evk}_j[i][0], \texttt{ct}_j'[i]\texttt{evk}_j[i][1]) \right]_{kQ_i}.$$

For $i \in T$ compute

$$(\texttt{ct}^{(i)}[0], \texttt{ct}^{(i)}[1]) = (k\texttt{ct}'[i], 0) \qquad \text{if } s_i' = 1$$
$$(\texttt{ct}^{(i)}[0], \texttt{ct}^{(i)}[1]) = (0, k\texttt{ct}'[i]) \qquad \text{if } s_i' = s.$$

Return $(\texttt{ct}[0], \texttt{ct}[1]) = \left[\left(\sum_{i=0}^{k} \texttt{ct}^{(i)}[0], \texttt{ct}^{(i)}[1] \right) \right]_{kQ_i}$.

$\texttt{Add}(\texttt{ct}_0, \texttt{ct}_1)$: Bring $\texttt{ct}_0, \texttt{ct}_1$ to the same level by modulus switching the ciphertext at the higher level to the lower level. Return $\texttt{ct} \coloneqq ([\texttt{ct}_0[0] + \texttt{ct}_1[0]]_{Q_i}, [\texttt{ct}_0[1] + \texttt{ct}_1[1]]_{Q_i})$.

$\texttt{AddConst}(\texttt{ct}, c \in \mathcal{R}_{Q_i})$: Return $\texttt{ct} \coloneqq ([\texttt{ct}[0] + c]_{Q_i}, \texttt{ct}[1])$.

$\texttt{MultConst}(\texttt{ct}, c \in \mathcal{R}_{Q_i} \setminus \{0\})$: Return $\texttt{ct} \coloneqq ([c \cdot \texttt{ct}[0]]_{Q_i}, [c \cdot \texttt{ct}[1]]_{Q_i})$.

$\texttt{Mult}(\texttt{ct}_0, \texttt{ct}_1, \texttt{evk})$: Bring $\texttt{ct}_0, \texttt{ct}_1$ to the same level by modulus switching the ciphertext at the higher level to the lower level.
Calculate $\texttt{ct}_{\text{pre-mult}} = (\texttt{ct}_{\text{pre-mult}}[0], \texttt{ct}_{\text{pre-mult}}[1], \texttt{ct}_{\text{pre-mult}}[2]) \coloneqq$
$([\texttt{ct}_0[0]\texttt{ct}_1[0]]_{Q_i}, [\texttt{ct}_0[0]\texttt{ct}_1[1]+\texttt{ct}_0[1]\texttt{ct}_1[0]]_{Q_i}, [\texttt{ct}_0[1]\texttt{ct}_1[1]]_{Q_i})$. Define $\texttt{ct}_j^{\text{pre-mult}}[2]$
such that

$$\texttt{ct}_{\text{pre-mult}}[2] = \sum_{j=1}^{\ell} \texttt{ct}_j^{\text{pre-mult}}[2] D_j^{\star}.$$

Compute

$$\texttt{ct} \coloneqq k(\texttt{ct}_{\text{pre-mult}}[0], \texttt{ct}_{\text{pre-mult}}[1]) + \sum_{j=1}^{\ell} (\texttt{ct}_j^{\text{pre-mult}}[2]\texttt{evk}_2[j][0], \texttt{ct}_j^{\text{pre-mult}}[2]\texttt{evk}_2[j][1]).$$

Output

$$\texttt{ct}_{\text{mult}} = \texttt{ModSwitch}((\texttt{ct}, Q_{sp}), Q_{i-1}).$$

$\texttt{Automorphism}(\texttt{ct}, \sigma)$: Let $\sigma \in \text{Aut}(\mathcal{R}_{Q_i})$ be an automorphism on $\mathcal{R}_{Q_i}$.
Compute $\texttt{ct} \coloneqq \texttt{KeySwitch}((\sigma(\texttt{ct}[0]), \sigma(\texttt{ct}[1])), \texttt{evk})$. Return

$$\texttt{ct}_{\text{auto}} = \texttt{ModSwitch}((\texttt{ct}, Q_{sp}), Q_{i-1}).$$

2.5 Noise of a BGV Ciphertext

Since the adversary knows the message m, we will work with the noise rather than the critical quantity [CS16]. This allows to simplify the proofs, as the distribution of the noise is centered at zero (and that of the critical quantity is not).

Definition 6. *We define the noise $\nu_i(\boldsymbol{ct})$ of a ciphertext encrypting m at level $i \in \{0, \ldots, L\}$ as*

$$\nu_i \colon \mathcal{R}_{Q_i}^2 \to \mathcal{R}_{Q_i}$$
$$\boldsymbol{ct} \mapsto [< \boldsymbol{ct}, \boldsymbol{sk} >]_{Q_i} - m,$$

2.6 Commitments

We will now define commitments to be used in the maliciously secure scheme defined and instantiated in Section 5. Informally, a commitment scheme is a cryptographic protocol that allows a party called Sender to commit to a value and send it to a party called Receiver, while keeping that value hidden. The Sender can then open the commitment to the Receiver at a later stage.

Definition 7 (Commitment Scheme). *A commitment scheme consists of the algorithms* Setup, Com, Open *defined as follows:*

Setup(1^λ) *Takes as input the security parameter λ, and outputs a commitment key* ck.

Com(ck, x) *Takes as input the commitment key* ck *and a message x, samples randomness $\mathbf{r}$, and outputs a commitment* com $\leftarrow$ Com(ck, x, $\mathbf{r}$) *and opening* op $= (x, \mathbf{r})$.

Open(ck, com, op) *Takes as input the commitment key* ck, *a commitment* com *and an opening* op, *and verifies that* com $=$ Com(pk, op). *It outputs 1 if the relation holds and 0 otherwise.*

A commitment scheme should satisfy the following two security properties.

Definition 8 (Hiding). *A commitment scheme is said to be* hiding *if, for all commitment keys* ck $\leftarrow$ Setup(1^λ) *and honestly sampled randomness $\mathbf{r}$, the distributions of two commitments* Com(ck, x, $\mathbf{r}$) *and* Com(ck, x', $\mathbf{r}$) *are indistinguishable to an adversary given the public key, messages x and x' chosen by the adversary, and only one of the commitments.*

Definition 9 (Binding). *A commitment scheme is said to be* binding *if, for all commitment keys* ck $\leftarrow$ Setup(1^λ) *and (potentially maliciously created) commitment* com $\leftarrow$ Com*(ck, x, $\mathbf{r}$), *an adversary cannot compute another valid opening* op$'$ $= (x', \mathbf{r}')$ *with different messages $x \neq x'$.*

We remark that a commitment scheme can be either both computationally hiding and binding, computationally hiding and statistically binding, or the opposite, but not both statistically hiding and binding at the same time.

2.7 Zero-Knowledge Proofs

Let $\mathcal{L}$ be a language and $\mathscr{R}$ a relation on $\mathcal{L}$. Then, x is an element in $\mathcal{L}$ if there exists a witness w such that $(x, w) \in \mathscr{R}$. We define a non-interactive proof zero-knowledge proof (NIZK) as follows.

Definition 10 (NIZK). *A non-interactive proof zero-knowledge proof for a relation $\mathscr{R}$ consists of the three algorithms* Setup, Prove, Verify*:*

Setup(1^λ) *Takes as input the security parameter λ, and outputs a common reference string* crs*.*

Prove(crs, x, w) *Takes as input the common reference string* crs*, a statement x and a witness w, and outputs a proof π for the relation $(x, w) \in \mathscr{R}$ on $\mathcal{L}$.*

Verify(crs, x, π) *Takes as input the common reference string* crs*, a statement x and a proof π, and either accepts (outputs 1) or rejects (outputs 0).*

We require the following security properties for a NIZK.

Definition 11 (Completeness). *A NIZK is said to be* complete *if, for all* crs $\leftarrow$ Setup(1^λ) *and $\pi \leftarrow$ Prove(crs, x, w), Verify(crs, x, π) returns 1 when $(x, w) \in \mathscr{R}$. That is, the verification algorithm always accepts if the statement is indeed true.*

Definition 12 (Soundness). *A NIZK is said to be* sound *if, for all* crs $\leftarrow$ Setup(1^λ) *and $\pi \leftarrow$ Prove*(crs, x, $\cdot$), Verify(crs, x, π) returns 0 except with negligible probability when $(x, w) \notin \mathscr{R}$. That is, the verification algorithm never accepts if the statement is false.*

Definition 13 (Zero-Knowledge). *An NIZK is said to be* (honest-verifier) zero-knowledge *if, for all* crs $\leftarrow$ Setup(1^λ), *a simulator SIM on input* (crs, x) *can produce a proof π' that is indistinguishable to an honestly generated proof $\pi \leftarrow$ Prove(crs, x, w) for each relation $(x, w) \in \mathscr{R}$ on $\mathcal{L}$.*

3 A Semi-Honest Circuit Private Variant of BGV

In this section we consider the semi-honest scenario, that is we assume the keys and the ciphertexts to be honestly generated. In Section 3.1 we give a formalized version of Definition 2 as stated in the introduction, and adapt it to a leveled setting. We then prove that BGV is computationally circuit private following this definition, assuming the adversary does not hold the secret key. In Section 3.2 we show that this definition is not strong enough if the adversary holds the secret key: then it can learn information about the circuit from the noise distribution, even if the underlying scheme is computationally circuit private. We therefore propose a new definition of computational circuit privacy that takes the noise distribution into account and show that this is naturally fulfilled by BGV too.

3.1 `BGV` is Circuit Private

Based on Gentry's original definition, we propose the following definition of computational circuit privacy.

Definition 14. *Let $\mathcal{E} = (KeyGen, Enc, Eval, Dec)$ be a homomorphic encryption scheme for circuits in $\mathcal{C}_\mathcal{E}$. Let λ be a security parameter. We say $\mathcal{E}$ is computationally circuit private if for any $(sk, pk, evk) \leftarrow KeyGen(1^\lambda)$, for any function f on plaintexts that has a corresponding circuit $C_f \in \mathcal{C}_\mathcal{E}$ and any fixed ciphertexts $ct_1, \ldots, ct_k$ that are an encryption by $Enc(\cdot, pk)$ of $m_1, \ldots, m_k$ the two distributions over the random coins in $Enc(\cdot, pk)$ and $Eval(\cdot, C_f, evk)$ of $Enc(f(m_1, \ldots, m_k), pk)$ and $Eval(ct_1, \ldots, ct_k, C_f, evk)$ are computationally indistinguishable. We say that $\mathcal{E}$ is maliciously computationally circuit private if the keys and the ciphertexts may be maliciously generated.*

Since `BGV` is a scheme that is mostly used in leveled mode, the following definition is more natural.

Definition 15. *We say that a homomorphic encryption scheme $\mathcal{E} = (KeyGen, Enc, Eval, Dec)$ is leveled computationally circuit private if Definition 14 holds if $Enc(f(m_1, \ldots, m_k), pk)$ and $Eval(ct_1, \ldots, ct_k, C_f, evk)$ are at the same level.*

If a bootstrappable leveled scheme fulfils Definition 15 it can also be made to fulfil Definition 14: if it fulfils Definition 15, the output of a circuit at level i will be indistinguishable from a fresh encryption at level i. Assume the bootstrapping function outputs a ciphertext at level j, and define the encryption function to encrypt at the same level j. Then, the levelled scheme fulfils Definition 14. We note that this is at the cost of losing levels during encruption and needing to bootstrap after an evaluation, even if the noise level or application does not call for it. Since the only information that is leaked by using the leveled definition is partial information about the circuit's depth, it seems more natural to apply the leveled definition to schemes in leveled mode. Therefore, this is the definition we will use for the first part of this section.

We prove the following theorem, which is the main result of this section.

Theorem 1 (Computational Circuit Privacy). *Let `BGV` be the scheme as defined in Section 2.4. Then `BGV` is leveled circuit private according to Definition 15.*

Proof. Let $ct = (ct[0], ct[1])$ be a `BGV` ciphertext. From the decryption function of `BGV`, we have $[m + te']_{Q_i} = [ct[0] + ct[1]s]_{Q_i}$, where $te' \in \mathcal{R}_{Q_i}$ is some error polynomial. Therefore, we can express $ct[0]$ modulo Q_i as $ct[0] = m - ct[1]s + te'$ and ct as

$$ct = (m - ct[1]s + te', ct[1]). \tag{1}$$

In this form, we can easily see that ct is an RLWE sample whenever $ct[1]$ is computationally indistinguishable from uniform random in $\mathcal{R}_{Q_i}$. The RLWE

problem furthermore guarantees that if $\mathtt{ct}[1]$ is of the form $as + e$ for $a \xleftarrow{\$} \mathcal{R}_{Q_i}$ public, $s \in \mathcal{R}_{Q_i}$ secret and $e \leftarrow \chi$ a secret error polynomial drawn from some error distribution χ, it is computationally indistinguishable from uniform random in $\mathcal{R}_{Q_i}$. Therefore, we proceed as follows. We first show that a fresh ciphertext is an RLWE sample. We then show that, applying the allowed operations to a $\mathtt{BGV}$ ciphertext that is an RLWE sample gives an output such that the second component again is of the form "$as + e$" or otherwise (computationally) indistinguishable from uniform random. Therefore, the output $\mathtt{ct}_{\mathrm{out}}$ is again an RLWE sample. Thus, it is computationally indistinguishable from uniform random in $(\mathcal{R}_{Q_i})^2$. We use this to show that all intermediary and final results in a homomorphic computation using $\mathtt{BGV}$ are RLWE samples. Therefore, all ciphertexts are computationally indistinguishable from uniform random and thus also from one another. Therefore, we have that $\mathtt{Enc}(f(m_1, \ldots, m_k), \mathtt{pk})$ and $\mathtt{Eval}(\mathtt{ct}_1, \ldots, \mathtt{ct}_k, C_f, \mathtt{evk})$ for any $\mathtt{ct}_i = \mathtt{Enc}(m_i, \mathtt{pk}), i \in \{0, \ldots, k\}$ and $C_f \in \mathcal{C}_{\mathtt{BGV}}$, are computationally indistinguishable from uniform random at their respective level and therefore also from one another. Thus $\mathtt{BGV}$ fulfils Definition 15. We will proceed to prove these claims.

Let $\mathtt{ct}_{\mathrm{fresh}}$ be a fresh ciphertext. By the security proof of $\mathtt{BGV}$, a fresh ciphertext is an RLWE sample.

Let $\mathtt{ct}_i = (\mathtt{ct}_i[0], \mathtt{ct}_i[1]), i \in \{0, 1\}$ be two fresh $\mathtt{BGV}$ ciphertexts, and therefore RLWE samples, encrypted with respect to the same public key, where $\mathtt{ct}_i[1] = au_i + te_{i,1}$. We now look at how the allowed operations affect the distribution of $\mathtt{ct}[1]$.

$\mathtt{Add:}$ Let $\mathtt{ct}_{\mathrm{add}} = \mathtt{Add}(\mathtt{ct}_0, \mathtt{ct}_1)$. We have

$$\mathtt{ct}_{\mathrm{add}}[1] = [\mathtt{ct}_0[1] + \mathtt{ct}_1[1]]_{Q_i} = [a(u_0 + u_1) + t(e_{0,1} + e_{1,1})]_{Q_i}.$$

Then, $a \xleftarrow{\$} \mathcal{R}_{Q_i}, s := u_0 + u_1 \leftarrow \mathcal{S} + \mathcal{S} := \mathcal{S}'$, and $e := te_{0,1} + te_{1,1} \leftarrow t\chi + t\chi := \chi'$. Since the RLWE problem does not specify the nature of χ and $\mathcal{S}$, $\mathtt{ct}_{\mathrm{add}}[1]$ is of the form $as + e$ and therefore computationally indistinguishable from uniform random.

$\mathtt{AddConst:}$ Let $\mathtt{ct}_{\mathrm{AddConst}} = \mathtt{AddConst}(\mathtt{ct}_0, \mathtt{const}) = (\mathtt{ct}_0[0] + \mathtt{const}, \mathtt{ct}_0[1])$. Since $\mathtt{ct}_{\mathrm{AddConst}}[1] = \mathtt{ct}_0[1]$, it is trivially computationally indistinguishable from uniform random.

$\mathtt{MultConst:}$ Let $\mathtt{ct}_{\mathrm{MultConst}} = \mathtt{MultConst}(\mathtt{ct}_0, \mathtt{const} \neq 0)$. Since $\mathtt{ct}_{\mathrm{MultConst}}[1] = [\mathtt{const} \cdot \mathtt{ct}_0[1]]_{Q_i} = [a \cdot \mathtt{const} \cdot u_0 + t \cdot \mathtt{const} \cdot e_{0,1}]_{Q_i}$ we have $a \xleftarrow{\$} \mathcal{R}_{Q_i}, s := \mathtt{const} \cdot u_0 \leftarrow \mathtt{const} \cdot \mathcal{S} := \mathcal{S}'$ and $e := t \cdot \mathtt{const} \cdot e_{0,1} \leftarrow t \cdot \mathtt{const} \cdot \chi := \chi'$. Therefore, $\mathtt{ct}_{\mathrm{MultConst}}[1]$ is of the form "$as + e$" and computationally indistinguishable from uniform random.

Mult: Let $\mathtt{ct}_{\mathrm{Mult}} = \mathtt{Mult}(\mathtt{ct}_0, \mathtt{ct}_1)$. Define $\mathtt{ct}'[0] = [\mathtt{ct}_0[0]\mathtt{ct}_1[0]]_{Q_i}, \mathtt{ct}'[1] = [\mathtt{ct}_0[0]\mathtt{ct}_1[1] + \mathtt{ct}_0[1]\mathtt{ct}_1[0]]_{Q_i}, \mathtt{ct}'[2] = [\mathtt{ct}_0[1]\mathtt{ct}_1[1]]_{Q_i}$. We have

$$
\mathtt{ct}_{\mathrm{Mult}}[1] = [k\mathtt{ct}'[1] + \sum_{j=1}^{\ell} \mathtt{ct}'_j[2]a_{2j}[1]]_{kQ_i}
$$

$$
= [k(\mathtt{ct}_0[0]\mathtt{ct}_1[1] + \mathtt{ct}_0[1]\mathtt{ct}_1[0]) + \sum_{j=1}^{\ell} \mathtt{ct}'_j[2]a_{2j}[1]]_{kQ_i}
$$

$$
= [ak(m_0 u_1 - 2asu_0 u_1 - 2etu_0 u_1 + e_{0,1}tu_1 + e_{1,1}stu_0 + m_1 u_0
$$
$$
+ e_{1,1}tu_0 - e_{0,1}stu_0)
$$
$$
+ tk(e_{1,1}m_0 - ee_{1,1}tu_0 + e_{0,1}e_{1,1}t + e_{0,1}m_1 - ee_{0,1}u_1 + e_{0,1}e_{1,1}t)
$$
$$
+ \sum_{j=1}^{\ell} \mathtt{ct}'_j[2]a_{2j}[1])]_{kQ_i}.
$$

Then, we have $a \xleftarrow{\$} \mathcal{R}_{Q_i}, e := tk(e_{1,1}m_0 - ee_{1,1}tu_0 + e_{0,1}e_{1,1}t + e_{0,1}m_1 - ee_{0,1}u_1 + e_{0,1}e_{1,1}t) + \sum_{j=1}^{\ell} \mathtt{ct}'_j[2]a_{2j}[1]) \leftarrow \chi'$ and $s := k(m_0 u_1 - 2asu_0 u_1 - 2etu_0 u_1 + e_{0,1}tu_1 + e_{1,1}stu_0 + m_1 u_0 + e_{1,1}tu_0 - e_{0,1}stu_0) \leftarrow \mathcal{S}'$. Since all components of s that are public are masked with an element that is secret, we can assume s to be secret. Since there are no specifications of the nature of $\mathcal{S}'$ and χ', $\mathtt{ct}_{\mathrm{mult}}[1]$ is of the form $as + e$ and therefore computationally indistinguishable from uniform random.

Auto: Let $\mathtt{ct}_{\mathrm{auto}} = \mathtt{Auto}(\mathtt{ct}_0, \sigma)$. We have

$$
\mathtt{ct}_{\mathrm{auto}} = \mathtt{KeySwitch}(\sigma(\mathtt{ct}_0), (1, \sigma(s)), \mathtt{evk})
$$

$$
= \left[\left(k\sigma(\mathtt{ct}_0[0]) + \sum_{j=1}^{\ell} \mathtt{ct}'_{0,j}[1]a_{1j}[0] \right)_{kQ_i}, \left[\sum_{j=1}^{\ell} \mathtt{ct}'_{0,j}[1]a_{1j}[1] \right]_{kQ_i} \right).
$$

Therefore,

$$
\mathtt{ct}_{\mathrm{auto}}[1] = \left[\sum_{j=1}^{\ell} \mathtt{ct}'_{0,j}[1]a_{1j}[1] \right]_{kQ_i}
$$

$$
= \left[\sum_{j=1}^{\ell} \mathtt{ct}'_{0,j}[1](au_j^{\mathtt{evk}} + te_j^{\mathtt{evk}}) \right]_{kQ_i}
$$

$$
= \left[a \left(\sum_{j=1}^{\ell} \mathtt{ct}'_{0,j}[1]u_k^{\mathtt{evk}} \right) + t \left(\sum_{j=1}^{\ell} \mathtt{ct}'_{0,j}[1]e_j^{\mathtt{evk}} \right) \right]_{kQ_i}.
$$

With $s := \sum_{j=1}^{\ell} \mathtt{ct}'_{0,j}[1] u_k^{\mathtt{evk}} \leftarrow \mathcal{S}'$ and $e := t \sum_{j=1}^{\ell} \mathtt{ct}'_{0,j}[1] e_j^{\mathtt{evk}} \leftarrow \chi'$, $\mathtt{ct}_{\mathrm{auto}}[1]$ is of the form $as + e$ and therefore computationally indistinguishable from uniform in $\mathcal{R}_{Q_i}$.

$\mathtt{ModSwitch}$: Let $\mathtt{ct}_{\mathrm{ModSwitch}} = \mathtt{ModSwitch}((\mathtt{ct}_0, Q_i), Q_j)$. Then we have $\mathtt{ct}_{\mathrm{ModSwitch}}[1] = \left[\left\lfloor \frac{Q_j}{Q_i} \mathtt{ct}_0[1] \right\rceil \right]_{kQ_i}$. So the distribution of $\mathtt{ct}_{\mathrm{ModSwitch}}[1]$ is the distribution of $\mathtt{ct}_0[1]$, scaled by $\frac{Q_j}{Q_i}$. Therefore, if $\mathtt{ct}_0[1]$ is of the form $as + e$ and computationally indistinguishable from uniform random, then so is $\mathtt{ct}_{\mathrm{ModSwitch}}[1]$.

We see that if the second component of an input to any operation is of the form $as + e$, then so is the second component of the output, with respect to the same public element a. Therefore, using the result of one operation as the input to another one will still yield an element whose second component is of the form "$as + e$", with respect to different s and e, but the same public element a. By Equation 1, the output ciphertext of any operation is therefore an RLWE sample if the input ciphertext is one. Therefore, all the ciphertexts appearing as final or intermediary results in the evaluation of a circuit on $\mathtt{BGV}$ ciphertexts are RLWE samples, and are therefore computationally indistinguishable from uniform random in $\mathcal{R}_{Q_i} \times \mathcal{R}_{Q_i}$. We can therefore conclude that if $\mathtt{Enc}(f(m_1, \ldots, m_k), \mathtt{pk})$ and $\mathtt{Eval}(C_f, \mathtt{ct}_1, \ldots, \mathtt{ct}_k, \mathtt{evk})$ are at the same level, then both their distributions are computationally indistinguishable from uniform random over the same ring. This concludes the proof. $\qquad\square$

Additionally, we note that the above result implies that the critical quantity is the underlying error of the RLWE sample, and therefore the ratio between the magnitude of the critical quantity and the ciphertext modulus determines the hardness of the underlying RLWE sample as shown in [APS15]. This illustrates that the critical quantity is relevant for both correctness and security, and as such is of paramount importance in the analysis of FHE schemes.

3.2 A New Definition of Circuit Privacy

Theorem 1 shows that $\mathtt{BGV}$ is leveled computationally circuit private in its original form, without the need for any post-processing or sanitisation. We obtain computational circuit privacy for free. This is counter-intuitive. Indeed, one would think that a ciphertext resulting from two additions should be distinguishable from a ciphertext resulting from one. The fact that the effect of this operation is not visible in the ciphertext is mainly due to the fact that part of the randomness in encryption is drawn uniformly randomly from $\mathcal{R}_{Q_i}$. This makes it very likely that a wrap-around modulo Q_i occurs in the ciphertext parts, thereby obliterating any information on the distribution. However, to ensure correctness, no such wrap-around can be allowed in the ciphertext noise. Therefore, even if no information about the circuit can be deduced from the ciphertext parts, it may be deduced from the ciphertext noise.

To give an example, [MP19] proved in Lemma 1 that the noise of a fresh ciphertext is normally distributed, with mean 0 and variance $(\frac{4}{3}n + 1)t^2\sigma_0^2$, where $\chi = \mathcal{N}(0, \sigma_0^2)$. Let $\mathtt{ct}_0, \mathtt{ct}_1$ be two fresh ciphertexts. Since the noise of the sum of two ciphertexts is the sum of the noises, the noise of $\mathtt{ct}_0 + \mathtt{ct}_1$ is distributed as $\mathcal{N}(0, 2(\frac{4}{3}n + 1)t^2\sigma_0^2)$. The two distributions are therefore clearly distinguishable, and will move even further apart the more additions are performed. Thus, even though we showed that $\mathtt{BGV}$ is computationally circuit private according to Definition 15, we see that this does not necessarily capture all the information that could be extracted from a ciphertext.

To obtain the noise of a ciphertext however, the adversary would need access to the secret key. Assuming that the adversary has access to the secret key is not unreasonable in this scenario. If we consider for example a Machine Learning as a Service scenario, the adversary easily can become the client, and therefore the secret-key holder.

We will therefore extend Definition 14 and 15 by additionally requiring that the noise distributions be statistically indistinguishable.

Definition 16. *Let $\mathcal{E} = (\mathit{KeyGen}, \mathit{Enc}, \mathit{Eval}, \mathit{Dec})$ be a homomorphic encryption scheme based on RLWE. We call $\mathcal{E}$ (leveled) computationally circuit private against an adversary holding the secret key if it fulfils Definition 14 (Definition 15) and additionally it holds true that*

$$\Delta(\nu(\mathit{Enc}(f(m_1, \ldots, m_k), \mathit{pk})), \nu(\mathit{Eval}(C_f \mathit{ct}_1, \ldots, \mathit{ct}_k, \mathit{evk}))) \leq negl(\lambda)$$

for all $C_f \in \mathcal{C}_{\mathcal{E}}$ and $\mathit{ct}_i = \mathit{Enc}(m_i, \mathit{pk}), i \in \{0, \ldots, k\}$. Again, we call $\mathcal{E}$ maliciously (leveled) computationally circuit private against an adversary holding the secret key, if the above holds even when the ciphertexts and the public keys are not assumed to be well-formed.

We proceed by showing how minor modifications to $\mathtt{Eval}$ in $\mathtt{BGV}$ make it leveled computationally circuit private even if the adversary holds the secret key, and without the need for any expensive machinery as the Ducas-Stehlé washing machine or noise flooding.

3.3 $\mathtt{BGV}$' - a leveled Semi-honest Circuit Private Variant of $\mathtt{BGV}$

We define the scheme $\mathtt{BGV}$' to consist of the algorithms $\mathtt{KeyGen}$, $\mathtt{Enc}$', $\mathtt{Eval}$' and $\mathtt{Dec}$, where $\mathtt{KeyGen}$ and $\mathtt{Dec}$ are identical to the algorithms of the same name in $\mathtt{BGV}$. We define $\mathtt{Eval}$' to allow for the same set of operations as $\mathtt{BGV}$, that is $\mathcal{C}_{\mathtt{BGV}} = \mathcal{C}_{\mathtt{BGV'}}$. $\mathtt{Eval}$' and $\mathtt{Enc}$' are defined as follows.

$\mathtt{Enc}'(\mathtt{pk}, m, 0 \leq i < L)$ Set $(\mathtt{ct}, Q_L) \leftarrow \mathtt{Enc}(\mathtt{pk}, m)$.
    Return $\mathtt{ct} \leftarrow \mathtt{ModSwitch}((\mathtt{ct}, Q_L), Q_i)$.

$\mathtt{Eval}'(C_f, \mathtt{ct}_0, \ldots, \mathtt{ct}_k, \mathtt{evk})$ Compute $C_f(\mathtt{ct}_0, \ldots, \mathtt{ct}_k)$.
    Return $\mathtt{ct} \leftarrow \mathtt{ModSwitch}((C_f(\mathtt{ct}_0, \ldots, \mathtt{ct}_k), Q_{i+1}), Q_i)$

There are two major differences between `BGV'` and `BGV` that will allow us to prove that `BGV'` is computationally circuit private according to Definition 16. The first is that we modulus switch a fresh encryption at least one level down, and the second one is that we force a modulus switching after a completed evaluation.

We show that `BGV'` is circuit private according to Definition 16. The results relies on the fact that the output noise from the `ModSwitch` operation is simply a rounding noise, provided that the input noise was smaller than $\frac{Q_i}{Q_{i-1}}$. Informally put, in that case, the `ModSwitch` operation will strip away all the information that was contained in the input noise.

The noise after modulus switching therefore is circuit independent. We analyse the conditions that must be fulfilled in order for `BGV'` to be circuit private according to Definition 16.

The fact that our result principally stems from the noise output after modulus switching also explains the difference in the conclusion with [KS23]. In their work, the authors conclude for `CKKS` that their Definition of circuit privacy (see Definition 22 in Supplemental Material A) can only be achieved by adding exponential noise. They however do not consider rescaling, a noise management operation in `CKKS` that is close to modulus switching in `BGV`. Since our proof entirely depends on modulus switching, strong differences in the results are natural.

The requirements that we impose upon the scheme are as follows. The impact of these constraints on the effectiveness of the scheme will be discussed later in the paper.

Requirement 1

$$||\nu_i(\mathtt{ct})||_\infty \le \sqrt{\frac{Q_i}{NQ_{i-1}} + t^2} - t \approx \sqrt{\frac{Q_i}{NQ_{i-1}}}, j \in \{1, \dots, k+m\}.$$

Requirement 2

$$\frac{t\ell D^\star_{max} B_e}{k_i} \le \frac{Q_i}{Q_{i-1}},$$

where $D^\star_{max}$ is the maximal digit in the decomposition of $\mathtt{ct}_2$ during key switching, ℓ is the number of such digits, and k_i is the factor by which the ciphertext modulus gets scaled to obtain the special modulus used during key-switching.

Leveled Semi-Honest Circuit Privacy In the remainder of this subsection, we prove the following theorem, which is the main result of this subsection.

Theorem 2 (Computational Circuit Privacy Against Adversary with Secret Key).
Let `BGV'` be the scheme as defined above, where values from the error distribution χ are bounded by $B_e \le \frac{1}{2tN}\frac{Q_i}{Q_{i-1}}$ and the secret key distribution $\mathcal{S}$ draws polynomials with Hamming weight h from the set $\{-1, 0, 1\}$. Then `BGV'` is leveled circuit private following Definition 16, as long as no more than α additions and μ multiplications by a constant $\mathtt{const} \in \{0, \dots, t-1\}$ are performed in a row, where α and μ are defined as $\alpha + t^\mu = \kappa$, and $t \le \sqrt{\frac{Q_i}{Q_{i-1}N^3}}\frac{2}{\kappa}$.

The proof of Theorem 2 heavily relies on the results from [CNP23a]. In that work, Lemma 8 states that if the Requirements 1 and 2 are fulfilled, the noise after modulus switching is independent of prior operations and only depends on the ring dimension, the plaintext modulus and the Hamming weight of the secret key, all of which do not carry information about the circuit. The proof will show that for any circuit $C \in \mathcal{C}_{\texttt{BGV}'}$, if the Requirements 1 and 2 are fulfilled for the input, they are fulfilled for the output. In the first part of the proof, we show that fresh ciphertexts fulfil the requirements, then proceed by showing that the operations that are defined for $\texttt{BGV}$ ciphertexts do not change this. Therefore, the final modulus switching will remove all information about the circuit from the noise. In further detail, our proof consists of the following steps.

1. Prove that for a fresh ciphertext $\texttt{ct}_{\text{fresh}} = \texttt{Enc}'(m, \texttt{pk})$ we have

$$\nu_i(\texttt{ct}_{\text{fresh}}) = \tau_0 + \tau_1 s + \tau_2,$$

   where τ_i is the error polynomial after coefficient-wise rounding to the next integer, such that the encrypted plaintext modulo t is unchanged.

2. Prove that if Requirements 1 and 2 are fulfilled for a ciphertext $\texttt{ct}$, then $\nu_i(\texttt{ct} = \texttt{Mult}(\texttt{ct}_0, \texttt{ct}_1)) = \nu_i(\texttt{ModSwitch}((\texttt{ct}, Q_j), Q_i)) = \tau_0 + \tau_1 s + \tau_2$ and $\nu_i(\texttt{AUTOMORPHISM}(\texttt{ct})) = \tau_0 + \tau_1 s + \tau_2$. That is, the noise only consists of the rounding noise and the resulting ciphertext no longer contains information about prior operations.

3. Prove that

$$||\tau_0 + \tau_1 s + \tau_2||_\infty \leq \sqrt{\frac{Q_i}{Q_{i-1} N}}, \text{ for all } i \in \{1, \dots, L\},$$

   and therefore if a ciphertext $\texttt{ct}$ fulfils Requirements 1 and 2 before $\texttt{ModSwitch}$ and $\texttt{AUTOMORPHISM}$, it will also fulfil them afterwards. Therefore, if the requirements where fulfilled before this step, they will be fulfilled after.

4. Prove that $\texttt{ADDCONST}$ does not affect the noise term.

5. Prove that for any ciphertext $\texttt{ct}$, we have

$$\nu_i(\underbrace{\texttt{ADD}(\texttt{ADD}(\dots \texttt{ADD}(\texttt{ct}, \texttt{ct}) \dots)))}_{\text{const-times}} = \nu_i(\texttt{MULTCONST}(\texttt{ct}, \texttt{const})),$$

   for all $\texttt{const} \in \{0, \dots, t-1\}$. Therefore, the operation multiplication by a constant $\texttt{const}$ can be expressed by $\texttt{const}$ additions, if we restrict multiplication by a constant to taking as input only degree 0 polynomials. We therefore do not consider multiplication by constant separately, since it is covered by additions. We extend the result to higher degree polynomials in Corollary 1 in Appendix C.

6. Prove that we then have

$$\kappa ||\tau_0 + \tau_1 s + \tau_2||_\infty \leq \sqrt{\frac{Q_i}{Q_{i-1} N}}.$$

18

7. Prove that the noise of $\mathtt{ct_{out}} = C_f(\mathtt{ct}_1, \dots, \mathtt{ct}_k)$, $f(x_1, \dots, x_k) = \sum_{i=1}^{k} x_i$ can be bounded by

$$k\|\tau_0 + \tau_1 s + \tau_2\|_\infty,$$

for any possible representation of f, as long as $k \leq \kappa$. Together with Step 6 we have now shown the following: if we do not perform more than α additions or μ multiplications by constant, where $\alpha + t^\mu = \kappa$, the output of a circuit consisting of additions and multiplications by constants fulfil the requirements if the inputs did.

8. Conclude that `BGV'` is circuit private following Definition 16. We show that for any circuit with the above restrictions, the output of the circuit will fulfil the requirements. Since we have shown in Step 2 that for a circuit fulfilling the requirements, modulus switching reduces the noise to the rounding noise and therefore strips away all information about the circuit, the result follows.

As a first step to the proof of the theorem, we need noise estimates for all the operations. They are given in the following lemmas. Due to space limitations the proofs of Lemma 1 to 4 are deferred to Appendix B.

Lemma 1. *Let $\{0, \dots, L\}$ be the set of indices in the moduli chain, excluding the special moduli and let Q_L be the top modulus. Let $\mathtt{ct}_0$ and $\mathtt{ct}_1$ be two ciphertexts with respect to ciphertext moduli Q_i and Q_j respectively. Let $\mathtt{ct}_{add} := \mathtt{ADD}(\mathtt{ct}_0, \mathtt{ct}_1)$. Then the noise of $\mathtt{ct}_{add}$ is given by*

$$
\begin{aligned}
\nu_{\min(i,j)}(\mathtt{ct}_{add}) = &(1 - \chi_{\{j+1,\dots,L\}}(i))\nu_i(\mathtt{ct}_0) \\
&+ \chi_{\{j+1,\dots,L\}}(i\nu_j(\mathtt{ModSwitch}((\mathtt{ct}_0, Q_i), Q_j)) \\
&+ \chi_{\{j,\dots,L\}}(i)\nu_j(\mathtt{ct}_1) \\
&+ (1 - \chi_{\{j,\dots,L\}}(i))\nu_i(\mathtt{ModSwitch}((\mathtt{ct}_1, Q_j), Q_i)).
\end{aligned}
$$

Lemma 2. *Let $\{0, \dots, L\}$ be the set of indices in the moduli chain, excluding the special moduli and let Q_L be the top modulus. Let $\mathtt{ct}_0$ and $\mathtt{ct}_1$ be two ciphertexts with respect to ciphertext moduli Q_i and Q_j respectively. Let $\mathtt{ct}_{mult} := \mathtt{MULT}(\mathtt{ct}_0, \mathtt{ct}_1)$. Then the noise of $\mathtt{ct}_{mult}$ is given by*

$$\nu_{\min(i,j)}(\mathtt{ct}_{mult}) = \nu_{\min(i,j)}(\mathtt{MULT}(\mathtt{ct}_0, \mathtt{ct}_1)) = \tau_0 + \tau_1 s + \tau_2,$$

where τ_i is the rounding error of coefficient-wise rounding the polynomial to the nearest integer having the same value modulo t.

Lemma 3. *Let $(\mathtt{ct}, Q_i)$ be a ciphertext and $\mathtt{const} \in \{0, \dots, t-1\}$ a constant in the plaintext space. The critical quantity after $\mathtt{ADDCONST}((\mathtt{ct}, Q_i), \mathtt{const})$ is given by*

$$\nu_i(\mathtt{ADDCONST}((\mathtt{ct}, Q_i), \mathtt{const})) = \nu_i(\mathtt{ct}).$$

The critical quantity after $\mathtt{MULTCONST}((\mathtt{ct}, Q_i), \mathtt{const})$ is given by

$$\nu_i(\mathtt{MULTCONST})((\mathtt{ct}, Q_i), \mathtt{const}) = \mathtt{const} \cdot \nu_i(\mathtt{ct}).$$

19

Lemma 4. *Let* $(\mathtt{ct}, Q_i)$ *be a ciphertext. Then the noise of* $\mathtt{AUTOMORPHISM}((\mathtt{ct}, Q_i))$ *is as follows*

$$\nu_i(\mathtt{AUTOMORPHISM}((\mathtt{ct}, Q_i))) \approx \tau_0 + \tau_1 s + \tau_2.$$

Using these lemmas, we can now proceed to the proof of Theorem 2.

Proof. **Step 1.** Let $\mathtt{ct} = \mathtt{Enc}'(m, \mathtt{pk}) = \mathtt{ModSwitch}(([m+\mathtt{pk}[0]u+te_0]_{Q_L}, [\mathtt{pk}[1]u+te_1]_{Q_L}), Q_i)$. Then we have

$$\nu_i(\mathtt{ct}) = \left\lfloor \frac{Q_i}{Q_L}[m + (-as - te)u + te_0]_{Q_L} \right\rceil_t + \lfloor [au + te_1]_{Q_L} \rceil_t s - \left\lfloor \frac{Q_i}{Q_{i-1}} m \right\rceil_t$$

$$= \frac{Q_i}{Q_L}(t(-eu + e_1 s + e_0)) + \tau_0 + \tau_1 s + \tau_2.$$

Since

$$\begin{aligned}
\|t(-eu + e_1 s + e_0)\|_\infty &\le t(\|-eu\|_\infty + \|e_1 s\|_\infty + \|e_0\|_\infty) \\
&\le t(N \|e\|_\infty \|u\|_\infty + N \|e_1\|_\infty \|s\|_\infty + \|e_0\|_\infty) \\
&\le t(N \|e\|_\infty + N \|e_1\|_\infty + \|e_0\|_\infty) \\
&\le B_e(1 + 2tN) \\
&\le \frac{1}{2tN} \frac{Q_i}{Q_{i-1}}(1 + 2tN) \approx \frac{Q_i}{Q_{i-1}}.
\end{aligned}$$

Therefore, the first term will become negligible compared to $\tau_0 + \tau_1 s + \tau_2$, and we have

$$\nu_i(\mathtt{ct}_{\mathrm{fresh}}) = \tau_0 + \tau_1 s + \tau_2.$$

Step 2. This step has been proven in Lemmas 2 and 4.

Step 3. We can approximate $\tau_0 + \tau_1 s + \tau_2$ as follows.

$$\begin{aligned}
\|\tau_0 + \tau_1 s + \tau_2\|_\infty &\le \|\tau_0\| + \|\tau_1 s\|_\infty + \|\tau_2\|_\infty \\
&\le \|\tau_0\|_\infty + N \|\tau_1\|_\infty \|s\|_\infty + \|\tau_2\|_\infty \\
&\le \frac{t}{2} + \frac{Nt}{2} + \frac{t}{2} \approx \frac{tN}{2}.
\end{aligned}$$

We can approximate $\|\tau_i\|_\infty, i \in \{0, 1, 2\}$ by $\frac{t}{2}$, since it is the error that arises from rounding up or down to the next integer modulo t, and therefore never grows larger than $\frac{t}{2}$. Since we imposed that $t < \sqrt{\frac{Q_i}{Q_{i-1}N^3}} \frac{2}{\kappa}$ we obtain

$$\|\tau_0 + \tau_1 s + \tau_2\|_\infty \le \frac{tN}{2} \le \frac{\sqrt{\frac{Q_i}{N^3 Q_{i-1}}} \frac{2}{\kappa} N}{2} = \frac{1}{\kappa} \sqrt{\frac{Q_i}{Q_{i-1}N}}.$$

Therefore, if Requirement 1 was fulfilled before modulus switching, it is still fulfilled afterwards. This is where the bound on t comes from.

Step 4. This step immediately follows from Lemma 3.

Step 5. We therefore only need to consider the two operations `ADD` and `MULTCONST`. First, we notice the following.

Let $\mathtt{ct}_1, \mathtt{ct}_2, \mathtt{ct}_3$ be ciphertexts at the same level $Q_i, i \in \{1, \ldots, L\}$. If ciphertexts at the same level are added, the noise respects associativity. That is, we have

$$\nu_i\big(\mathtt{ADD}(\mathtt{ADD}(\mathtt{ct}_1, \mathtt{ct}_2), \mathtt{ct}_3)\big) = \nu_i\big(\mathtt{ADD}(\mathtt{ct}_1, \mathtt{ADD}(\mathtt{ct}_2, \mathtt{ct}_3))\big),$$

since we do not have to modulus switch and it hence does not matter which ciphertexts are added first. More formally,

$$
\begin{aligned}
&\nu_i\big(\mathtt{ADD}((\mathtt{ct}_k, Q_i), (\mathtt{ct}_l, Q_i))\big) \\
=& (1 - \chi_{\{i+1,\ldots,L\}}(i)\nu_i(\mathtt{ct}_k) + \chi_{\{i+1,\ldots,L\}}(i\nu_i(\mathtt{ModSwitch}((\mathtt{ct}_1, Q_i), Q_i)) \\
&+ \chi_{\{i,\ldots,L\}}(i)\nu_i(\mathtt{ct}_l) + (1 - \chi_{\{i,\ldots,L\}}(i)\nu_i(\mathtt{ModSwitch}(\mathtt{ct}_l))) \\
=& \nu_i(\mathtt{ct}_k) + \nu_i(\mathtt{ct}_l).
\end{aligned}
$$

Since we have no modulus switches in the circuit, the same basic arithmetic as in the plaintext ring holds true also for ciphertexts, and we can write, for `const` being a plaintext polynomial of degree 0,

$$\nu_i\big(\underbrace{\mathtt{ADD}(\mathtt{ADD}(\ldots \mathtt{ADD}}_{\mathrm{const\ -\ times}}(\mathtt{ct}, \mathtt{ct}), \ldots), \mathtt{ct}\big) = \nu_i(\mathtt{MULTCONST}(\mathtt{ct}, \mathtt{const})).$$

Since we can express `MULTCONST` through `ADD` with the same noise, we do not need to consider `MULTCONST` further.

Step 6. We have seen in Step 3 that we can bound $||\tau_0 + \tau_1 s + \tau_2||_\infty$ as

$$||\tau_0 + \tau_1 s + \tau_2||_\infty \leq \frac{1}{\kappa}\sqrt{\frac{Q_i}{Q_{i-1}N}}.$$

We therefore see immediately that we can bound $\kappa \, ||\tau_0 + \tau_1 s + \tau_2||_\infty$ by $\frac{Q_i}{Q_{i-1}N}$.

Step 7. We can assume all ciphertexts $\mathtt{ct}_1, \ldots, \mathtt{ct}_k$ to be either fresh ciphertexts or the result of multiplications, automorphism or modulus switches. If one of the input ciphertexts is the result of an addition between i ciphertexts, the noise bound is given by $k' = k + i$. We therefore do not need to consider this separately. As argued in Step 5, the noise of multiplications by constant can be

given via additions, so the same applies to ciphertexts resulting of multiplications by constant. As seen in Lemma 3 additions by constant do not change the noise, therefore we do not need to consider them here. Therefore, by Lemmas 2 and 4 and previous steps, all ciphertexts can be assumed to have noise $\tau_0 + \tau_1 s + \tau_2$.

Let $\mathtt{ct} = C_f(\mathtt{ct}_1, \ldots, \mathtt{ct}_k)$. The ciphertext $\mathtt{ct}$ has the highest noise if $\mathtt{ct}_1, \ldots, \mathtt{ct}_k$ are at the same level. This is due to the following. If a subset of i ciphertexts $\mathtt{ct}_{j_1}, \ldots, \mathtt{ct}_{j_i}$ is at a level ℓ_1, whereas the rest is at a level ℓ_0 by Lemma 1 we have $\nu_{\ell_1}(\mathtt{ct}_{j_1} + \ldots + \mathtt{ct}_{j_i}) = i(\tau_0 + \tau_1 s + \tau_2)$. However, to make addition with the ciphertexts at level ℓ_0 possible, the result of the addition will have to be modulus switched down. Now, since $i < k < \kappa$, $\mathtt{ct}$ still fulfils Requirement 1.

Therefore, by Step 2, modulus switching will reduce the noise of the addition of the first i ciphertexts to $\tau_0 + \tau_1 s + \tau_2$. Now, by Lemma 1, adding the remaining $k - i$ ciphertexts results in a final noise of

$$\nu_{\ell_0}(\mathtt{ct}) = (k - i + 1)(\tau_0 + \tau_1 s + \tau_2) \leq k(\tau_0 + \tau_1 s + \tau_2).$$

Therefore, the noise is largest in the case where no intermediary modulus switches are applied. This happens if all ciphertexts are at the same level. In this case, Lemma 1 gives us for the noise of $\mathtt{ct}$

$$\nu_i(\mathtt{ct}) = k(\tau_0 + \tau_1 s + \tau_2).$$

Hence, the claim holds true.

Step 8. Fresh ciphertexts fulfil Requirements 1 and 2. If no more than α additions and μ multiplications with constant are performed in a row, then a ciphertext that is the output of such a circuit fulfils these requirements. Therefore, the final modulus switching as defined in $\mathtt{Eval}'$ will reduce the noise of the output ciphertext to $\tau_0 + \tau_1 s + \tau_2$, which is independent of the circuit that was evaluated on the ciphertext. Since each fresh ciphertext is modulus switched, they also have noise $\tau_0 + \tau_1 s + \tau_2$.

Since the noise terms of a fresh encryption and a ciphertext resulting of an evaluation of f are the same, it is easy to see that their distributions are the same as well. $\mathtt{BGV}'$ is therefore circuit private according to Definition 16. $\qquad \square$

The proof induces a parameter trade-off between the number of additions that can be carried out in a row, the plaintext modulus and the ciphertext moduli ratio. This is because we only perform a $\mathtt{ModSwitch}$ after a multiplication, and we need that all input ciphertexts have noise smaller than $\frac{Q_i}{Q_{i-1}}$. This will ensure that the $\mathtt{ModSwitch}$ operation will strip away all the information about the circuit that is carried in the noise. Since a multiplication by a large constant has substantial influence on the noise growth, the larger the plaintext modulus is, the fewer multiplications by constant can be carried out, or the larger the ratio $\frac{Q_i}{Q_{i-1}}$ needs to be. This of course holds for $\mathtt{BGV}$ in general. However, the

requirements lead to a smaller noise budget then if BGV was not supposed to be circuit private and therefore harsher trade-offs need to be introduced.

There are two ways of getting better parameters, or even being able to use the same parameters as for non-circuit-private BGV. The first would be to swap the worst-case bounds out for average case bounds on BGV as presented in [CNP23a] or [MP19]. We chose worst-case bounds in this work, since average-case bounds can fail with a small probability. Using average-case bounds would thus have introduced a failure probability in Theorem 2. Therefore, the advantage of an adversary holding the secret key against the computational circuit privacy of BGV' would no longer only be the probability that the adversary guesses the circuit correctly, that is $\frac{1}{|\mathcal{C}_{\text{BGV}'}|}$, but additionally would depend on the probability of one of the bounds failing, which in turn would be dependent on the circuit. Using average-case bounds would therefore be theoretically very cumbersome and would weaken the theorem. In practice however, it may be a worthwhile trade-off to accept this additional failure probability in order to get tighter noise estimates and therefore be able to perform more operations without increasing the parameters. The second option is to apply modulus switching after a batch of additions, which would then guarantee that the requirements are always fulfilled. This would allow to use the usual parameters to the detriment of needing more levels to evaluate a circuit.

As a second remark, we have proven that it is possible to achieve leveled computational circuit privacy according to Definition 16 for BGV' without noise flooding. It is possible to extend BGV' to allow it to fulfil the unleveled definition. Again, this is done by forcing a bootstrapping and then a modulus switching. The bootstrapping allows to get to a higher modulus Q_B, the modulus switching allows reducing the noise thereafter to $\tau_0 + \tau_1 s + \tau_2$. If any fresh ciphertext then is modulus switched down to level Q_{B-1}.

4 Lattice-Based Commitments and Zero-Knowledge Proofs

To extend our results to the malicious setting, we need lattice-based commitments and zero-knowledge proofs of shortness, linearity, and multiplication.

Notation. Let η be an integer such that $\binom{n}{\eta} \cdot 2^\eta > 2^{128}$ and define the two sets $\mathfrak{C}_\eta = \{c \in \mathcal{R}_{Q_L} \mid \|c\|_\infty = 1 \wedge \|c\|_1 = \eta\}$ and $\bar{\mathfrak{C}}_\eta = \{c - c' \mid c, c' \in \mathfrak{C}_\eta \wedge c \neq c'\}$. Let $B_{\text{Com}}, B_{\mathcal{S}}, B_{\mathcal{S}^2}, B_\chi \in \mathbb{N}$ be bounds and let $\bar{\sigma}_\chi \in \mathbb{R}^+$ be the standard deviation of a discrete distribution $\mathcal{D}_{\bar{\sigma}_\chi}$ over $\mathcal{R}_{Q_L}$ where $\bar{\sigma}_\chi = \eta \cdot B_\chi \cdot \sqrt{n}$ for a bound B_χ.

Rejection Sampling. To ensure that the proofs do not leak any information about the secret values, we use rejection sampling to force the output to be independent of the secrets [Lyu09,Lyu12]. When the standard deviation is roughly the ℓ_2 norm of the challenge times the secret vector, the algorithm accepts roughly $1/3$ of the time, when also rejecting samples where the inner product of the secret vector and the output vector is negative [LNS21].

Commitments. We can commit to elements in $\mathcal{R}_{Q_L}$ using the BDLOP commitment scheme [BDL$^+$18]. For simplicity, we present the scheme instantiated over rings instead of modules, committing to only one ring element at a time.

Setup(1^λ): Takes in a security parameter λ, samples uniformly random a_1, a_2, a_3 from $\mathcal{R}_{Q_L}$ and outputs the public commitment key ck defined as:

$$\mathsf{ck} = \begin{bmatrix} \mathbf{a}_1 \ 0 \\ \mathbf{a}_2 \ 1 \end{bmatrix} = \begin{bmatrix} 1 \ a_1 \ a_2 \ 0 \\ 0 \ 1 \ a_3 \ 1 \end{bmatrix}.$$

Com(ck, x): Takes as input the commitment key ck and an element x in $\mathcal{R}_{Q_L}$, samples $\mathbf{r}$ of length 3 from $\mathcal{R}_{Q_L}$ such that $\|\mathbf{r}\|_\infty \leq B_{\mathsf{Com}}$, and computes:

$$\mathsf{com} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 1 \ a_1 \ a_2 \ 0 \\ 0 \ 1 \ a_3 \ 1 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ x \end{bmatrix}.$$

It outputs the commitment com and the opening $\mathsf{op} = (x, (\mathbf{r}, 1))$.

Open(ck, com, op) The algorithm takes in the commitment key ck, the commitment com and the opening $\mathsf{op} = (x, (\mathbf{r}, f))$ where $f \in \bar{\mathfrak{C}}_\eta$. It verifies:

$$f \cdot \mathsf{com} \overset{?}{=} \begin{bmatrix} 1 \ a_1 \ a_2 \ 0 \\ 0 \ 1 \ a_3 \ 1 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ f \cdot x \end{bmatrix} \quad \text{and } \forall i \in [3]\colon \|r_i\|_2 \overset{?}{\leq} 4\bar{\sigma}_{\mathsf{Com}}\sqrt{n}.$$

It outputs 1 if the relations hold and 0 otherwise.

The BDLOP commitment scheme is hiding if the Learning With Errors (LWE) problem is hard for vectors of ℓ_∞ norm B_{Com} over a lattice of dimension $2n$, and the scheme is binding if the Short Integer Solution (SIS) problem is hard for vectors of ℓ_2 norm $16\bar{\sigma}_{\mathsf{Com}}\sqrt{\eta n}$ over a lattice of dimension $3n$ [BDL$^+$18].

Proof of Shortness. Using the techniques by Lyubashevsky [Lyu09, Lyu12], we can prove knowledge of an opening x such that $\|x\|_\infty \leq B_{\mathcal{X}}$ as follows.

1. Sample vector $\mathbf{y}$ of length 3 over $\mathcal{R}_{Q_L}$ according to $\mathcal{D}_{\bar{\sigma}_{\mathsf{Com}}}$ and y_4 according to $\mathcal{D}_{\bar{\sigma}_{\mathcal{X}}}$. Compute $\mathbf{w} = \mathsf{ck} \cdot \bar{\mathbf{y}}$ where $\bar{\mathbf{y}} = \mathbf{y}||y_4$.
2. Hash $\mathbf{w}$ to an element c in $\mathfrak{C}_\eta$ and compute $\bar{\mathbf{z}} = \bar{\mathbf{y}} + c \cdot \bar{\mathbf{r}}$ where $\bar{\mathbf{r}} = \mathbf{r}||x$.
3. Rejection sample with respect to $(\bar{\mathbf{y}}, \bar{\mathbf{z}})$. If it outputs 1 then output $(c, \bar{\mathbf{z}})$ and otherwise restart the protocol by sampling a new $\bar{\mathbf{y}}$.

The verifier checks if $\|\bar{z}_i\|_2 \leq 2\bar{\sigma}_{\mathsf{Com}}\sqrt{n}$ for all $i \in [3]$, if $\|\bar{z}_4\|_2 \leq 2\bar{\sigma}_{\mathcal{X}}\sqrt{n}$ and if the hash of $\mathsf{ck} \cdot \mathbf{z} - c \cdot \mathsf{com}$ equals c. It outputs 1 if all checks verify, and otherwise it outputs 0.

This proof ensures that the commitment is honestly computed and that the prover has knowledge of secret value x bounded by $4\bar{\sigma}_{\mathcal{X}}\sqrt{n}$ in the ℓ_2 norm.

The proof system [LNP22] can potentially be more efficient, but are also more complicated and the parameters need to be carefully adjusted to this setting.

Proof of Linear Relations. The proof of linear relations can be constructed in a similar way as above [BDL+18]. Let com be a commitment to x with randomness $\mathbf{r}$, com' be a commitment to x' with randomness $\mathbf{r}'$, and let g be a public element in $\mathcal{R}_{Q_L}$. Then, the proof that $x' = g \cdot x$ proceeds as follows.

1. Sample vectors $\mathbf{y}$ and $\mathbf{y}'$ of length 3 over $\mathcal{R}_{Q_L}$ according to $\mathcal{D}_{\bar{\sigma}_{\mathsf{Com}}}$ and compute $\mathbf{w} = \mathbf{a}_1 \cdot \mathbf{y}$ and $\mathbf{w}' = \mathbf{a}_1 \cdot \mathbf{y}'$ and $u = g \cdot \mathbf{a}_2 \cdot \mathbf{y} - \mathbf{a}_2 \cdot \mathbf{y}'$.
2. Hash $(\mathbf{w}, \mathbf{w}', u)$ to an element c in $\mathfrak{C}_\eta$, and compute $\mathbf{z} = \mathbf{y} + c \cdot \mathbf{r}, \mathbf{z}' = \mathbf{y}' + c \cdot \mathbf{r}'$.
3. Rejection sample with respect to $(\mathbf{y}, \mathbf{z})$, and $(\mathbf{y}', \mathbf{z}')$. If it outputs 1 then output $(c, \mathbf{z}, \mathbf{z}')$ and otherwise restart the protocol by sampling new $(\mathbf{y}, \mathbf{y}')$.

The verifier checks if $\|z_i, z_i'\|_2 \leq 2\bar{\sigma}_{\mathsf{Com}}\sqrt{n}$ for all $i \in [3]$ and if the hash of $(\mathbf{a}_1 \cdot \mathbf{z} - c \cdot c_1, \mathbf{a}_1 \cdot \mathbf{z}' - c \cdot c_1', g \cdot \mathbf{a}_2 \cdot \mathbf{z} - \mathbf{a}_2 \cdot \mathbf{z}' + c \cdot (c_2' - g \cdot c_2))$ equals c. It outputs 1 if all checks verify, and otherwise it outputs 0.

This proof ensures that the commitments are honestly computed and that the prover has knowledge of secret values x and x' such that $x' = g \cdot x$. We can easily extend the proof to more terms, containing one vector $\mathbf{z}$ for each term.

Proof of Multiplication. The proof of multiplication uses the proof of linear relations as a building block. Let com be a commitment to a secret value s with ℓ_∞ norm bounded by $B_\mathcal{S}$ and let com' be a commitment to x'. Let $B_{\mathcal{S}^2} = B_\mathcal{S}^2 \cdot n$. We extend the multiplication proof of [ALS20] to prove that $x' = s^2$ as follows.

1. Sample vectors $\mathbf{y}$ and $\mathbf{y}'$ of length 3 over $\mathcal{R}_{Q_L}$ according to $\mathcal{D}_{\bar{\sigma}_{\mathsf{Com}}}$, y_4 according to $\mathcal{D}_{\bar{\sigma}_\mathcal{S}}$ and y_4' according to $\mathcal{D}_{\bar{\sigma}_{\mathcal{S}^2}}$. Define $\bar{\mathbf{y}} = \mathbf{y}||y_4$ and $\bar{\mathbf{y}}' = \mathbf{y}'||y_4'$. Then compute $\mathbf{w} = \mathsf{ck} \cdot \bar{\mathbf{y}}$ and $\mathbf{w}' = \mathsf{ck} \cdot \bar{\mathbf{y}}'$, and commit to $s \cdot y_4 - y_4'$ and y_4^2 as com_1 and com_0 with randomness $\mathbf{r}_1$ and $\mathbf{r}_0$.
2. Hash $(\mathbf{w}, \mathbf{w}', \mathsf{com}_0, \mathsf{com}_1)$ to an element c in $\mathfrak{C}_\eta$ and compute $\bar{\mathbf{z}} = \bar{\mathbf{y}} + c \cdot \bar{\mathbf{r}}$ and $\bar{\mathbf{z}}' = \bar{\mathbf{y}}' + c \cdot \bar{\mathbf{r}}'$ where $\bar{\mathbf{r}} = \mathbf{r}||s$ and $\bar{\mathbf{r}}' = \mathbf{r}'||s^2$.
3. Rejection sample with respect to $(\bar{\mathbf{y}}, \bar{\mathbf{z}})$, and $(\bar{\mathbf{y}}', \bar{\mathbf{z}}')$. If it outputs 0 then restart the protocol by sampling a new $(\bar{\mathbf{y}}, \bar{\mathbf{y}}')$. Otherwise, compute a proof of linearity π_{Lin} for $\bar{z}_4^2 - c \cdot \bar{z}_4' = (s \cdot y_4 - y_4') \cdot c + y_4^2$ with respect to com_1 and com_1. Output $(\mathsf{com}_0, \mathsf{com}_1, \pi_{\mathsf{Lin}}, c, \bar{\mathbf{z}}, \bar{\mathbf{z}}')$.

The verifier checks if $\|\bar{z}_i, \bar{z}_i'\|_2 \leq 2\bar{\sigma}\sqrt{n}$ for all $i \in [3]$, $\|\bar{z}_4\|_2 \leq 2\bar{\sigma}_\mathcal{S}\sqrt{n}$, $\|\bar{z}_4'\|_2 \leq 2\bar{\sigma}_{\mathcal{S}^2}\sqrt{n}$, and if the hash of $(\mathbf{a}_1 \cdot \mathbf{z} - c \cdot c_1, \mathbf{a}_1 \cdot \mathbf{z}' - c \cdot c_1', \mathsf{com}_0, \mathsf{com}_1)$ equals c. It also checks that the proof of linearity holds. It outputs 1 if all checks verify, and otherwise it outputs 0.

This proof ensures that the commitments are honestly computed, that com is a commitment to a secret value s with ℓ_2 norm bounded by $4\bar{\sigma}_\mathcal{S}\sqrt{n}$ and that com' is a commitment to s^2 with ℓ_2 norm bounded by $4\bar{\sigma}_{\mathcal{S}^2}\sqrt{n}$.

Norms. The proofs π_{KeyGen} and π_{Enc} ensure that the upper bound in the ℓ_∞ norm of elements sampled from a given distribution over a set $\mathcal{X}$ is $4 \cdot \eta \cdot B_\mathcal{X} \cdot n^{3/2}$ where $B_\mathcal{X}$ is the ℓ_∞ norm of elements from the set $\mathcal{X}$. Here, we have that $\mathcal{X} \in \{\mathcal{S}, \mathcal{S}^2, \chi\}$.

5 Maliciously Secure Key Generation and Encryption

To ensure computational circuit privacy against malicious adversaries corrupting either the key generation or the encryption process, we must ensure that these algorithms were computed honestly. We achieve this by extending the algorithms to be verifiable, that is, the algorithms also output commitments to all secret key material and randomness used to generate public keys and ciphertexts in addition to zero-knowledge proofs (ZKPs), ensuring that these values have bounded norms and are used correctly to create the output.

   The evaluation algorithm is the same, except that it also takes the ZKPs as input and verifies that they are correct before evaluating the circuit.

5.1 Extended Definitions

We define verifiable key generation and verifiable encryption over $\mathcal{R}_{Q_L}$ as follows:

$\text{VerKeyGen}(1^\lambda)$: Run $\text{KeyGen}(1^\lambda)$ and receive $(\text{sk}, \text{pk}, \text{evk})$ in addition to e and e_i for all i. Then sample commitment randomness $\mathbf{r}_s, \mathbf{r}_e, \mathbf{r}_{s^2}, \mathbf{r}_{e_i}$ for each $i \in \{0, \ldots, m = \log_\omega(Q_L)\}$ and create commitments $\text{com}_s = \text{Com}(\text{ck}, s, \mathbf{r}_s)$, $\text{com}_e = \text{Com}(\text{ck}, e, \mathbf{r}_e), \text{com}_{s^2} = \text{Com}(\text{ck}, s^2, \mathbf{r}_{s^2})$ and $\text{com}_{e_i} = \text{Com}(\text{ck}, e_i, \mathbf{r}_{e_i})$ for all i. Finally, create a ZKP π_{KeyGen} proving the relation $\mathscr{R}_{\text{KeyGen}}$ below, including all the commitments in the proof. Output $(\text{sk}, \text{pk}, \text{evk}, \pi_{\text{KeyGen}})$.

$\text{VerEncrypt}(\text{pk}, m)$: Run $\text{Enc}(\text{pk}, m)$ and receive ct in addition to u, e_1 and e_2. Sample commitment randomness $\mathbf{r}_u, \mathbf{r}_{e_1}, \mathbf{r}_{e_2}, \mathbf{r}_m$ and create commitments $\text{com}_u = \text{Com}(\text{ck}, u, \mathbf{r}_u), \text{com}_{e_1} = \text{Com}(\text{ck}, e_1, \mathbf{r}_{e_1}), \text{com}_{e_2} = \text{Com}(\text{ck}, e_2, \mathbf{r}_{e_2})$ and $\text{com}_m = \text{Com}(\text{ck}, m, \mathbf{r}_m)$. Finally, create a ZKP π_{Enc} proving the relation $\mathscr{R}_{\text{Enc}}$ below, including all commitments in the proof. Output $(\text{ct}, \pi_{\text{Enc}})$.

   We define the following relations for the zero-knowledge proofs given in the verifiable key generation and verifiable encryption algorithms:

– *Verifiable key generation:* The ZKP π_{KeyGen} in the key generation proves knowledge of a short key s and short error term e corresponding to pk and that com_s and com_e are proper commitments to s and e. Furthermore, it proves knowledge of a short key s^2 being the square of the key s and short error terms e_i corresponding to the given evk and that com_{s^2} and com_{e_i} are proper commitments to s^2 and e_i for all i. For publicly fixed $B_e, t, h \in \mathbb{N}$ and decomposition base $\mathbf{D} = \{D_1^\star, \ldots, D_L^\star\}$, the relation $\mathscr{R}_{\text{KeyGen}}$ is defined as:

$$\mathscr{R}_{\text{KeyGen}} := \left\{ (x, w) \left| \begin{array}{l} x := (\text{pk}, \text{evk}, \pi_{\text{KeyGen}}) \ \wedge \ w := (s, s^2, e, \{e_i\}, \mathbf{r}_s, \mathbf{r}_{s^2}, \mathbf{r}_e, \{\mathbf{r}_{e_i}\}) : \\ \text{pk}[0] = -\text{pk}[1]s - te \ \wedge \ \forall i \ \text{evk}[0][i] = -\text{evk}[1][i]s - te_i + D_i^\star s^2 \\ \quad \wedge \ \|s\|_\infty \le 1 \ \wedge \ \|s^2\|_\infty \le h \ \wedge \ \forall i \ \|e, \{e_i\}\|_\infty \le B_e \ \wedge \\ \text{Open}(\text{ck}, \text{com}_s, (s, \mathbf{r}_s)) \ \wedge \ \text{Open}(\text{ck}, \text{com}_{s^2}, (s^2, \mathbf{r}_{s^2})) \ \wedge \\ \text{Open}(\text{ck}, \text{com}_e, (e, \mathbf{r}_e)) \ \wedge \ \forall i \ \text{Open}(\text{ck}, \text{com}_{e_i}, (e_i, \mathbf{r}_{e_i})) \end{array} \right. \right\}.$$

– *Verifiable encryption:* The ZKP $\pi_{\mathtt{Enc}}$ in the encryption algorithm proves that the given ciphertext $\mathtt{ct}$ is generated using the encryption algorithm. For publicly fixed $B_e, t \in \mathbb{N}$, the relation $\mathscr{R}_{\mathtt{Enc}}$ is defined as:

$$
\mathscr{R}_{\mathtt{Enc}} := \left\{ (x,w) \left|
\begin{array}{l}
x := (\mathtt{pk}, \mathtt{ct}, \pi_{\mathtt{Enc}}) \ \wedge \ w := (u, e_1, e_2, m, \mathbf{r}_u, \mathbf{r}_{e_1}, \mathbf{r}_{e_2}, \mathbf{r}_m,) : \\
\quad \mathtt{ct}[0] = m + \mathtt{pk}[0]u + te_1 \ \wedge \ \mathtt{ct}[1] = \mathtt{pk}[1]u + te_2 \\
\quad \wedge \ \|u\|_\infty \le 1 \ \wedge \ \|e_1, e_2\|_\infty \le B_e \ \wedge \ \|m\| \le t \ \wedge \\
\quad \mathsf{Open}(\mathtt{ck}, \mathtt{com}_u, (u, \mathbf{r}_u)) \ \wedge \ \mathsf{Open}(\mathtt{ck}, \mathtt{com}_{e_1}, (e_1, \mathbf{r}_{e_1})) \\
\quad \wedge \ \mathsf{Open}(\mathtt{ck}, \mathtt{com}_{e_2}, (e_2, \mathbf{r}_{e_2})) \ \wedge \ \mathsf{Open}(\mathtt{ck}, \mathtt{com}_m, (m, \mathbf{r}_m))
\end{array}
\right. \right\}.
$$

These ZKPs can be instantiated with proofs of shortness [Lyu09, Lyu12, LNP22, LN17], linear relations [BDL+18], and multiplication [ALS20], using commitments [BDL+18], as described above.

5.2 Proving Verifiable Key Generation

In the actively secure key generation, we commit to $s, s^2, e, \{e_i\}$ individually for each $i \in [\{0, \ldots, m = \log_\omega(Q_L)\}]$, where each commitment is of size $2 \cdot n \cdot \log_2 q$ bits. We give a ZKP of shortness for e and each e_i, where each proof is of size $3 \cdot n \cdot \log_2(6 \cdot \bar{\sigma}_{\mathsf{Com}}) + n \cdot \log_2(6 \cdot \bar{\sigma}_\chi)$ bits. We give one proof of multiplication for s and s^2, which is of size $4 \cdot n \cdot \log_2 q + 4 \cdot n \cdot \log_2(6 \cdot \bar{\sigma}_{\mathsf{Com}}) + 6 \cdot n \cdot \log_2(6 \cdot \bar{\sigma}_{\mathsf{Com}}) + n \cdot \log_2(6 \cdot \bar{\sigma}_{\mathcal{S}}) + n \cdot \log_2(6 \cdot \bar{\sigma}_{\mathcal{S}^2})$ bits. Finally, we give one proof of linearity for $\mathtt{pk}$ and $m + 1$ proofs of linearity for $\mathtt{evk}$. The proof for $\mathtt{pk}$ consists of two committed values, and hence, the proof is of size $6 \cdot n \cdot \log_2(6 \cdot \bar{\sigma}_{\mathsf{Com}})$ bits. Each part of $\mathtt{evk}$ consists of three committed values, and hence, the proof is of size $9 \cdot n \cdot \log_2(6 \cdot \bar{\sigma}_{\mathsf{Com}})$ bits. To conclude, the verifiability ZKP $\pi_{\mathtt{KeyGen}}$ is of size:

$$
n((2m + 16)\log_2 q + (12m + 31)\log_2(6\bar{\sigma}_{\mathsf{Com}})
$$
$$
+(m + 2)\log_2(6\bar{\sigma}_\chi) + \log_2(6\bar{\sigma}_{\mathcal{S}}) + \log_2(6\bar{\sigma}_{\mathcal{S}^2})) \ \text{bits.}
$$

5.3 Proving Verifiable Encryption

In the actively secure encryption algorithm, we commit to u, e_1, e_2, and m. We then compute four ZKPs of shortness and two ZKPs of linear relations. The proof for $\mathtt{ct}[0]$ consists of three committed values, and the proof for $\mathtt{ct}[1]$ consists of two committed values. To conclude, the verifiability proof $\pi_{\mathtt{Enc}}$ is of size:

$$
n(4\log_2 q + 31\log_2(6\bar{\sigma}_{\mathsf{Com}})) \ \text{bits.}
$$

5.4 Maliciously Secure Circuit Privacy

We conclude by proving computational circuit privacy in the setting of a malicious adversary, when the prover is required to prove that he has followed the key generation and encryption algorithms honestly. For circuit privacy we only require integrity, but note that this changes the CPA security of the client ciphertext since we send additional information (commitments and proofs).

However, it follows directly that the extended ciphertexts are CPA secure if the commitments are hiding, and the proofs are zero-knowledge. Thus, we only need to prove circuit privacy where we focus on the binding property of the commitment and the soundness property of the zero-knowledge proofs to ensure that the keys and ciphertexts are proper RLWE samples.

Theorem 3 (Maliciously Computational Circuit Privacy). *Let vBGV be the verifiable BGV scheme as defined in Section 5.1. Assuming that the RLWE problem is hard, and that vBGV is instantiated with a binding commitment scheme and sound proofs of shortness, linear relations, and multiplication, then vBGV is maliciously leveled computationally circuit private if the BGV variant it is based on fulfils Definition 15, and it is maliciously leveled computationally circuit private against an adversary holding the secret key if the BGV variant it is based on fulfils Definition 16.*

Proof. It was proven in Theorem 1 that BGV is circuit private according to Definition 15 if the input ciphertexts and public keys are proper RLWE samples. In vBGV, the client needs to compute commitments $\mathsf{Com}_s, \mathsf{Com}_{s^2}, \mathsf{Com}_e, \mathsf{Com}_{e_i}$ to the randomnesses s, s^2, e, e_i used in key generation, and compute a zero-knowledge proof that shows that

$$\mathtt{pk}[0] = -\mathtt{pk}[1]s - te$$
$$\mathtt{evk}[0][i] = -\mathtt{evk}[1][i]s - te_i + D_i^\star s^2,$$

where s, e, and all e_i are sufficiently short and the openings of the commitments are valid. The client then additionally needs to commit to randomnesses u, e_1 and e_2 through commits $\mathsf{Com}_u, \mathsf{Com}_{e_1}$, and Com_{e_2} and to a message m through Com_m. It needs to prove that the encryption relations

$$\mathtt{ct}[0] = m + \mathtt{pk}[0]u + te_1$$
$$\mathtt{ct}[1] = \mathtt{pk}[1]u + te_2$$

hold, openings of the commitments verify, and randomness is sufficiently short.

Assume then that the client would try to create a public key and an evaluation key maliciously or an external attacker would try to inject maliciously generated keys into the server. Then the proofs of linear relation would still have to hold for the keys, since otherwise the server will reject them. If the linear relations hold, however, the public and evaluation keys are RLWE samples by definition. Similarly, the proof of linear relation for the ciphertexts ensures that they are RLWE samples with respect to the same public key. Since Theorem 1 carries through whenever it is assured that the public keys and the input ciphertexts used for evaluation are RLWE samples, it also carries through for vBGV even if we assume the keys and ciphertexts to be potentially maliciously generated.

The proof of Theorem 2 carries through, if Theorem 1 and Requirements 1 and 2 hold. If for the underlying BGV variant of vBGV Theorem 1 holds, then the parameters have been chosen such that those two requirements hold if randomnesses drawn from the secret key distribution have an infinity norm bounded

by 1 and randomnesses drawn from the error distribution have an infinity norm bound by B_e. Since the zero-knowledge proof of shortness is sound, the server will not accept the proof unless these bounds hold. Since the commitment scheme is binding, an attacker cannot compute the commitments on randomnesses different than those used in the key generation and the zero-knowledge proof of shortness. Therefore, it is ensured that Requirements 1 and 2 hold. But then, the proof of Theorem 2 carries through, even though the ciphertexts or public keys may be maliciously generated. The result follows. □

6 Conclusion

We have shown that `BGV` naturally fulfils Definition 15 and can be adapted to fulfil Definition 14 in a straightforward way. We then get semi-honest computational circuit privacy for free. Furthermore, `BGV` fulfils Definition 14 if the result of a ciphertext evaluation of a circuit is bootstrapped and modulus switched before being returned to the client. Since the level of a ciphertext does not give much information about the circuit and since bootstrapping is costly, Definition 15 can be considered the most natural one for leveled schemes.

We further showed that, even if the distribution of the ciphertexts does not leak information about the circuit, the noise may. Consequently, we have proposed a new definition of computational circuit privacy that takes the noise into account. We showed how the strategic use of modulus switching allows us to fulfil this definition. Finally, we showed how to construct proofs of well-formedness for keys and ciphertexts. If keys and ciphertexts are extended by these proofs when sent to the server, we achieve malicious computational circuit privacy.

References

AGHV22. Adi Akavia, Craig Gentry, Shai Halevi, and Margarita Vald. Achievable CCA2 relaxation for homomorphic encryption. In Eike Kiltz and Vinod Vaikuntanathan, editors, *TCC 2022: 20th Theory of Cryptography Conference, Part II*, volume 13748 of *Lecture Notes in Computer Science*, pages 70–99. Springer, Heidelberg, November 2022.

ALS20. Thomas Attema, Vadim Lyubashevsky, and Gregor Seiler. Practical product proofs for lattice commitments. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020, Part II*, volume 12171 of *Lecture Notes in Computer Science*, pages 470–499. Springer, Heidelberg, August 2020.

APS15. Martin R. Albrecht, Rachel Player, and Sam Scott. On the concrete hardness of learning with errors. *Journal of Mathematical Cryptology*, 9(3):169–203, 2015.

BDL+18.   Carsten Baum, Ivan Damgård, Vadim Lyubashevsky, Sabine Oechsner, and Chris Peikert. More efficient commitments from structured lattice assumptions. In Dario Catalano and Roberto De Prisco, editors, *SCN 18: 11th International Conference on Security in Communication Networks*, volume 11035 of *Lecture Notes in Computer Science*, pages 368–385. Springer, Heidelberg, September 2018.

BdMW16.   Florian Bourse, Rafaël del Pino, Michele Minelli, and Hoeteck Wee. FHE circuit privacy almost for free. In Matthew Robshaw and Jonathan Katz, editors, *Advances in Cryptology – CRYPTO 2016, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pages 62–89. Springer, Heidelberg, August 2016.

BGV12.   Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (Leveled) fully homomorphic encryption without bootstrapping. In Shafi Goldwasser, editor, *ITCS 2012: 3rd Innovations in Theoretical Computer Science*, pages 309–325. Association for Computing Machinery, January 2012.

Bra12.   Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical GapSVP. In Reihaneh Safavi-Naini and Ran Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, volume 7417 of *Lecture Notes in Computer Science*, pages 868–886. Springer, Heidelberg, August 2012.

CGGI16.   Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In Jung Hee Cheon and Tsuyoshi Takagi, editors, *Advances in Cryptology – ASIACRYPT 2016, Part I*, volume 10031 of *Lecture Notes in Computer Science*, pages 3–33. Springer, Heidelberg, December 2016.

CGGI17.   Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachène. Faster packed homomorphic operations and efficient circuit bootstrapping for TFHE. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 377–408. Springer, Heidelberg, December 2017.

CKKS17.   Jung Hee Cheon, Andrey Kim, Miran Kim, and Yong Soo Song. Homomorphic encryption for arithmetic of approximate numbers. In Tsuyoshi Takagi and Thomas Peyrin, editors, *Advances in Cryptology – ASIACRYPT 2017, Part I*, volume 10624 of *Lecture Notes in Computer Science*, pages 409–437. Springer, Heidelberg, December 2017.

CNP23a.   Anamaria Costache, Lea Nürnberger, and Rachel Player. Optimisations and tradeoffs for helib. In Mike Rosulek, editor, *Topics in Cryptology – CT-RSA 2023*, pages 29–53, Cham, 2023. Springer International Publishing.

CNP23b.   Anamaria Costache, Lea Nürnberger, and Rachel Player. Optimisations and trade-offs for HElib. Cryptology ePrint Archive, Paper 2023/104, 2023.

CS16.   Ana Costache and Nigel P. Smart. Which ring based somewhat homomorphic encryption scheme is best? In Kazue Sako, editor, *Topics in Cryptology – CT-RSA 2016*, volume 9610 of *Lecture Notes in Computer Science*, pages 325–340. Springer, Heidelberg, February / March 2016.

DD22.   Nico Döttling and Jesko Dujmovic. Maliciously Circuit-Private FHE from Information-Theoretic Principles. In Dana Dachman-Soled, editor, *3rd Conference on Information-Theoretic Cryptography (ITC 2022)*, volume 230 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 4:1–4:21, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

DM14.   Léo Ducas and Daniele Micciancio. Improved short lattice signatures in the standard model. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 335–352. Springer, Heidelberg, August 2014.

DM15.   Léo Ducas and Daniele Micciancio. FHEW: Bootstrapping homomorphic encryption in less than a second. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 617–640. Springer, Heidelberg, April 2015.

DS16.   Léo Ducas and Damien Stehlé. Sanitization of FHE ciphertexts. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology – EUROCRYPT 2016, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 294–310. Springer, Heidelberg, May 2016.

FV12.   Junfeng Fan and Frederik Vercauteren. Somewhat practical fully homomorphic encryption. Cryptology ePrint Archive, Report 2012/144, 2012. `https://eprint.iacr.org/2012/144`.

Gen09. Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. `crypto.stanford.edu/craig`.

GSW13. Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In Ran Canetti and Juan A. Garay, editors, *Advances in Cryptology – CRYPTO 2013, Part I*, volume 8042 of *Lecture Notes in Computer Science*, pages 75–92. Springer, Heidelberg, August 2013.

HS20.   Shai Halevi and Victor Shoup. Design and implementation of HElib: a homomorphic encryption library. Cryptology ePrint Archive, Report 2020/1481, 2020. `https://eprint.iacr.org/2020/1481`.

Klu22. Kamil Kluczniak. Circuit privacy for FHEW/TFHE-style fully homomorphic encryption in practice. Cryptology ePrint Archive, Report 2022/1459, 2022. `https://eprint.iacr.org/2022/1459`.

KS23.   Kamil Kluczniak and Giacomo Santato. On circuit private, multikey and threshold approximate homomorphic encryption. Cryptology ePrint Archive, Paper 2023/301, 2023.

LM06.   Vadim Lyubashevsky and Daniele Micciancio. Generalized compact Knapsacks are collision resistant. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP 2006: 33rd International Colloquium on Automata, Languages and Programming, Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 144–155. Springer, Heidelberg, July 2006.

LN17.   Vadim Lyubashevsky and Gregory Neven. One-shot verifiable encryption from lattices. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017, Part I*, volume 10210 of *Lecture Notes in Computer Science*, pages 293–323. Springer, Heidelberg, April / May 2017.

LNP22. Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Maxime Plançon. Lattice-based zero-knowledge proofs and applications: Shorter, simpler, and more general. In Yevgeniy Dodis and Thomas Shrimpton, editors, *Advances in Cryptology – CRYPTO 2022, Part II*, volume 13508 of *Lecture Notes in Computer Science*, pages 71–101. Springer, Heidelberg, August 2022.

LNS21. Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. Shorter lattice-based zero-knowledge proofs via one-time commitments. In Juan

31

Garay, editor, *PKC 2021: 24th International Conference on Theory and Practice of Public Key Cryptography, Part I*, volume 12710 of *Lecture Notes in Computer Science*, pages 215–241. Springer, Heidelberg, May 2021.

LPR10. Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 1–23. Springer, Heidelberg, May / June 2010.

Lyu09. Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In Mitsuru Matsui, editor, *Advances in Cryptology – ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 598–616. Springer, Heidelberg, December 2009.

Lyu12. Vadim Lyubashevsky. Lattice signatures without trapdoors. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 738–755. Springer, Heidelberg, April 2012.

MP19. Sean Murphy and Rachel Player. A central limit framework for ring-LWE decryption. Cryptology ePrint Archive, Report 2019/452, 2019. `https://eprint.iacr.org/2019/452`.

OPP14. Rafail Ostrovsky, Anat Paskin-Cherniavsky, and Beni Paskin-Cherniavsky. Maliciously circuit-private FHE. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology – CRYPTO 2014, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pages 536–553. Springer, Heidelberg, August 2014.

Supplemental Material

A   Definitions of Circuit Privacy

Since Gentry first introduced the concept of circuit privacy in 2009 [Gen09] many different definitions have been proposed. In this section, we give an overview of existing notions and prove relations among them.

A.1   Defining Semi-Honest Security

We will give here the definitions for semi-honest circuit privacy.

Definition 17 ([Gen09]). *We say that a homomorphic encryption scheme $\mathcal{E}$ is (statistically) circuit private for circuits in $\mathcal{C}_{\mathcal{E}}$, if for any key set (sk, pk, evk) output by $KeyGen(1^{\lambda})$, for any function f on plaintexts that has a corresponding circuit $C_f \in \mathcal{C}_{\mathcal{E}}$ and any fixed ciphertexts $ct_1, \ldots, ct_k$ that are the encryption by $Enc_{\mathcal{E}}(\cdot, pk)$ of plaintexts $m_1, \ldots, m_k$ respectively, the following holds true for the distributions over the random coins in $Enc_{\mathcal{E}}(\cdot, pk)$ and $Eval_{\mathcal{E}}(\cdot, C_f, evk)$:*

$$\Delta(Enc_{\mathcal{E}}(f(m_1, \ldots, m_k), pk), Eval_{\mathcal{E}}(ct_1, \ldots, ct_k, C_f, evk)) \leq negl(\lambda).$$

Definition 18 ([Gen09]). *We say that a leveled homomorphic encryption scheme $\mathcal{E}$ is leveled (statistically) circuit private, if Definition 1 holds if the ciphertexts $ct = Enc_{\mathcal{E}}(f(m_1, \ldots, m_k), pk)$ and $ct' = Eval_{\mathcal{E}}(ct_1, \ldots, ct_k, C_f, evk)$ are at the same level.*

Definition 19 ([BdMW16]). *We say that a homomorphic encryption scheme $\mathcal{E}$ is circuit private if there exists a PPT algorithm SIM, such that for any function f on k plaintexts, that has a corresponding circuit $C_f \in \mathcal{C}_{\mathcal{E}}$ of length $L = poly(\lambda)$, any plaintexts $m_1, \ldots, m_k \in \{0, 1\}$, the following holds*

$$\Delta(SIM(1^{\lambda}, evk, f(m_1, \ldots, m_k), 1^L, ct_1, \ldots, ct_k),$$
$$(evk, Eval_{\mathcal{E}}(ct_1, \ldots, ct_k, C_f, evk), ct_1, \ldots, c_k, 1^{\lambda})) \leq negl(\lambda),$$

where $ct_i \leftarrow Enc(m_i, pk)$, $(evk, pk) \leftarrow KeyGen_{\mathcal{E}}(1^{\lambda})$.

Definition 20 ([Klu22]). *Let $\mathcal{E}$ be a homomorphic encryption scheme with set of admissible circuit $\mathcal{C}_{\mathcal{E}}$. Let f be a function that has a corresponding polynomial-sized circuit $C_f \in \mathcal{C}_{\mathcal{E}}$. $\mathcal{E}$ is said to be circuit private, if, for any k plaintexts $m_1, \ldots, m_k$, there exists a PPT simulator SIM such that*

$$\Delta(SIM(evk, f(m_1, \ldots, m_k)), Eval(ct_1, \ldots, ct_k, C_f, evk)) \leq negl(\lambda),$$

where $ct_i \leftarrow Enc(m_i, pk)$, and $evk, pk \leftarrow SetUp(1^{\lambda})$.

Definition 21 ([DD22]). *We say an FHE scheme $\mathcal{E}$ is semi honestly circuit private, if for all λ, and for all $pk, evk, sk \leftarrow KeyGen(1^\lambda)$, messages $m_1, \ldots, m_k$, and functions f, f' such that $f(m) = f'(m)$, that have corresponding circuits $C_f, C_{f'} \in \mathcal{C}_\mathcal{E}$*

$$\Delta(Eval(Enc(m_1, pk), \ldots, Enc(m_k, pk), C_f, evk),$$
$$Eval(Enc(m_1, pk), \ldots, Enc(pk, m_k), C_{f'}, evk) \leq negl(\lambda).$$

Definition 22 ([KS23]). *Let $\mathcal{E}$ be a homomorphic encryption scheme for circuits $C \in \mathcal{C}_\mathcal{E}$. Define the experiment $Exp_b^{IND\text{-}CP}(\mathcal{A})$, where $b \in \{0, 1\}$ is a bit and $\mathcal{A}$ is an adversary. The experiment is defined as follows*

$Exp_b^{IND\text{-}CP}(\mathcal{A})\text{:}$

$$r, r_1, \ldots, r_k \xleftarrow{\$} \mathcal{U},$$
$$(sk, pk, evk) \leftarrow KeyGen(\lambda, r),$$
$$m_1, \ldots, m_k, C_{f_0}, C_{f_1}, st \leftarrow \mathcal{A}(\lambda, r_1, \ldots, r_k),$$
$$[ct_i \leftarrow Enc(m_i, r_i, pk)]_{i=1}^k,$$
$$ct \leftarrow Eval(ct_1, \ldots, ct_k, C_{f_b}, evk),$$
$$b' \leftarrow \mathcal{A}(st, ct),$$
$$\textbf{return } b',$$

where $C_{f_0}, C_{f_1} \in \mathcal{C}_\mathcal{E}$ and $f_0(m_1, \ldots, m_k) = f_1(m_1, \ldots, m_k)$. The scheme $\mathcal{E}$ is said to be $\lambda-$bit IND-CP secure if, for any adversary $\mathcal{A}$ we have that $\lambda \leq \log_2 \frac{T(\mathcal{A})}{adv^\mathcal{A}}$.

A.2 Maliciously secure definitions

Definition 23 ([OPP14]). *Let $\mathcal{E}$ denote a homomorphic encryption scheme that returns correct results for all circuits $C \in \mathcal{C}_{\mathcal{E},U}$ represented by a representation model U. We say $\mathcal{E}$ is (maliciously) circuit private if there exist an unbounded algorithm $SIM(1^k, pk^\star, evk^\star, ct_1^\star, \ldots, ct_k^\star, b)$ and an unbounded deterministic algorithm $EXT(1^k, pk^\star, b)$, such that for all k, and all $pk^\star, evk^\star, ct_1^\star, \ldots, ct_k^\star$ and all functions f that have a corresponding circuit $C_f \in \mathcal{C}_{\mathcal{E},U}$ the following holds*

- $m^\star = EXT(1^k, pk^\star, ct_1^\star \ldots, ct_k^\star).$
- $\Delta(SIM(1^k, pk^\star, ct_1^\star, \ldots, ct_k^\star, U(f, m^\star)), Eval(ct_1^\star, \ldots, ct_k^\star, C_f, evk^\star)) \leq negl(\lambda).$

In particular, for circuits $C_f \in \mathcal{C}_{\mathcal{E},U}$ the output distribution of $Eval$ (including length) depends only on k. For leveled schemes, SIM and EXT also take a depth parameter 1^d. We say a scheme is semi-honest circuit-private, if the above holds, where $pk^\star, evk^\star, ct^\star$ belong to the set of well-formed public-key ciphertexts pairs.

Definition 24 ([DD22]). *We say an HE scheme $\mathcal{E}$ for circuits $C \in \mathcal{C}_\mathcal{E}$ is maliciously, statistically circuit private if there exists an unbounded simulator*

SIM with one-time oracle access to the function f and its corresponding circuit $C_f \in \mathcal{C}_\mathcal{E}$ such that for all λ, and for all public keys $pk^\star, evk^\star$ and ciphertexts $ct_1^\star, \ldots, ct_k^\star$

$$\Delta(SIM(1^\lambda, pk^\star, evk^\star, ct_1^\star, \ldots, ct_k^\star), Eval(ct_1^\star, \ldots, ct_k^\star, C_f, evk^\star)) \leq negl(\lambda).$$

Definition 25 ([DD22]). *Let $\mathcal{E}$ be a homomorphic encryption scheme for circuits $C \in \mathcal{C}_\mathcal{E}$. Let $\Phi : \mathcal{C}_\mathcal{E} \to \{0,1\}^\star$ be a leakage function. We say $\mathcal{E}$ is Φ (maliciously) circuit private, if there exists an unbounded simulator SIM with one-time oracle access to f and its corresponding circuit $C_f \in \mathcal{C}_\mathcal{E}$ such that for all λ, public keys $pk^\star, evk^\star$ ciphertexts $ct_1^\star, \ldots, ct_k^\star$, and PPT adversaries $\mathcal{A}$,*

$$Pr[\mathcal{A}(SIM^f(1^\lambda, pk^\star, ct_1^\star, \ldots, ct_k^\star, \Phi(C_f)))] - Pr[\mathcal{A}(Eval(ct_1^\star, \ldots, ct_k^\star, C_f, evk^\star))] \leq negl(\lambda).$$

A.3 Hybrid secure definitions

Definition 26 ([AGHV22]). *A homomorphic scheme $\mathcal{E}$ for circuits $C \in \mathcal{C}_\mathcal{E}$ is circuit-private$^+$, if the following holds with probability $\geq 1 - negl(\lambda)$ over the choice of $(pk, evk, sk) \leftarrow KeyGen(1^\lambda)$. For every function f that has a corresponding circuit $C_f \in \mathcal{C}_\mathcal{E}$ over k inputs and ciphertexts $ct_1^\star, \ldots, ct_k^\star$ in the ciphertext space of $\mathcal{E}$ the following distributions are statistically close*

$$\Delta(Enc_\mathcal{E}(f(Dec_\mathcal{E}(ct_1^\star, sk), \ldots, Dec_\mathcal{E}(ct_k^\star, sk)), pk), Eval_\mathcal{E}(ct_1^\star, \ldots, ct_k^\star, C_f, evk)) \leq negl(\lambda),$$

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

A.4 Relations between the definitions

We now proceed to prove the implications between the definitions outlined above. We begin by proving relations between the semi-honest definitions, with the exception of Definition 22. This is the only definition based on an indistinguishability game and therefore a special case. It additionally assumes the adversary to be in possession of the secret key. Definitions 23, 24 and 25 also assume the adversary to have the secret key. These definitions are however simulation-based.

A simulation-based definition tells us how much information can be obtained from observing the output of the Eval algorithm on a ciphertext ct. The less input the simulator gets, the stronger the definition is. Therefore, knowledge of the secret key actually weakens the simulation-based definitions. This is in contrast to the indistinguishability-based definitions. In this case, the more information the adversary has access to, the stronger her capabilities are, and therefore the stronger the definition is. Therefore, assuming knowledge of the secret key strengthens Definition 22, while it weakens Definitions 23, 24 and 25.

The knowledge of the secret key in Definition 22 essentially allows the adversary to guess the underlying circuit by not only looking at the ciphertext

distribution but also at the noise which, as has been shown in [KS23] and is discussed in Section 3.2, is much more instructive.

Proving an implication between Definition 17 and 22 does not appear to be feasible in general. In order to do so, one would have to argue about the distribution of the noise, based on the distribution of the ciphertext components. Since these two distributions are not independent, one must argue about the concrete distributions, which cannot then be done in generality. This can however be done for specific schemes, and we discuss the implication between those two definitions in the specific case of `BGV` in Section 3.2. In the following Lemma a Definition number with the superscript sh indicates that this is originally a malicious definition, but we here consider it in a semi-honest setting to have a broader comparison between the definitions.

Lemma 5. *The following relations between the semi-honest definitions hold for IND-CPA secure schemes based on the RLWE problem*

$$26 \Rightarrow 17 \Rightarrow 18 \Rightarrow 20 \Rightarrow 19 \Rightarrow 21 \Rightarrow 24^{sh} \Rightarrow 23^{sh}$$

$$\Downarrow$$

$$25^{sh}$$

The proof that Definition 17 implies Definition 19 corrects a minor mistake in [BdMW16], where it is claimed that Definition 19 is stronger than Definition 17.

Proof. We begin by noticing that

$$26 \Rightarrow 17.$$

The definitions are equivalent, but for the fact that Definition 26 also allows for maliciously generated ciphertexts. It is therefore stronger.

Furthermore,

$$17 \Rightarrow 18.$$

If an adversary cannot distinguish between the distribution of the simulation and the distribution of the output of the evaluation without the requirement that both are at the same level, she will in particular not be able to distinguish between those two if they are at the same level. We furthermore notice that

$$20 \Rightarrow 19.$$

The definitions are equivalent, with the exception that in 19 the simulator receives as additional information the length of the circuit as well as the exact ciphertexts on which the simulation is being run. The same argument as above holds: if the simulator is able to output a distribution that is indistinguishable

36

from the output of the evaluation without having any additional information (level, length of circuit), it will still be able to do so in the possession of the additional information.

We are left with showing what the relation between Definitions 20 and 18 is, as well as showing the relation between the two and the semi-honest settings of Definitions 23 and 25.

If Definition 18 holds, then we know that

$$\Delta(\texttt{Enc}_{\mathcal{E}}^{\ell}(f(m_1,\ldots,m_k),\texttt{pk}),\texttt{Eval}_{\mathcal{E}}^{\ell}(C_f,\texttt{Enc}(m_1,\texttt{pk}),\ldots,\texttt{Enc}(m_k,\texttt{pk}),\texttt{evk})) \leq \text{negl}(\lambda),$$

for each level $\ell \in \{0,\ldots,L\}$. Now, the simulator as defined in 20 does not have access to the public key. If it did, it could simply output a fresh encryption of any message at the correct level, which by Definition 18 would then be indistinguishable from a ciphertext that is the output of $\texttt{Eval}()$. Since the simulator does not have access to the public key, it can draw $\texttt{ct}_{\text{sim}}[0], \texttt{ct}_{\text{sim}}[1] \xleftarrow{\$} \mathcal{R}_{Q_\ell}$ at random and output $\texttt{ct}_{\text{sim}} = (\texttt{ct}_{\text{sim}}[0] + f(m_1,\ldots,m_k), \texttt{ct}_{\text{sim}}[1])$ as the simulation of a ciphertext. This tuple is still distributed uniformly at random in $\mathcal{R}_{Q_\ell} \times \mathcal{R}_{Q_\ell}$. Adding $f(m_1,\ldots,m_k)$ only shifts the mean of the distribution, as $f(m_1,\ldots,m_k)$ is deterministic. Since all elements in $\mathcal{R}_{Q_\ell}$ are taken modulo Q_ℓ, this shift is not detectable, so the distribution of the tuple remains uniform. Since the RLWE problem guarantees that an RLWE sample cannot be distinguished from a uniformly random element from the same ring and a fresh encryption is an RLWE sample, the distribution of this simulated encryption of $f(m_1,\ldots,m_k)$ cannot be distinguished from the distribution of a real ciphertext encrypting the same value. We therefore have

$$\Delta(\texttt{ct}_{\text{sim}}, \texttt{Enc}_{\mathcal{E}}^{\ell}(f(m_1,\ldots,m_k))) \leq \text{negl}(\lambda).$$

By Definition 18, the distribution of a fresh encryption is indistinguishable from the distribution of the result of an evaluation of the ciphertexts at the same level. So, the simulator's output remains indistinguishable. We therefore have that

$$18 \Rightarrow 20 \Rightarrow 19.$$

Next, we show that

$$19 \Rightarrow 21.$$

If Definition 19 holds, then we have for two functions $f_i, i \in \{1,2\}$ and a PPT simulator $\texttt{SIM}$

$$\Delta(\texttt{SIM}(1^{\lambda},\texttt{evk}, f_i(m_1,\ldots,m_k), 1^L, \texttt{ct}_1,\ldots,\texttt{ct}_k),$$
$$(\texttt{evk}, \texttt{Eval}_{\mathcal{E}}(C_{f_i}, \texttt{ct}_1,\ldots,\texttt{ct}_k,\texttt{evk}), \texttt{ct}_1,\ldots,\texttt{ct}_k, 1^{\lambda})) \leq negl(\lambda). \quad (2)$$

Since Definition 21 requires $f(m_1,\ldots,m_k) = f_2(m_1,\ldots,m_k)$ we get

$$\Delta(\texttt{SIM}(1^{\lambda}, \texttt{evk}, f_1(m_1,\ldots,m_k), 1^L, \texttt{ct}_1,\ldots,\texttt{ct}_k),$$

$$(\mathtt{evk}, \mathtt{Eval}_{\mathcal{E}}(C_{f_2}, \mathtt{ct}_1, \ldots, \mathtt{ct}_k, \mathtt{evk}), \mathtt{ct}_1, \ldots, \mathtt{ct}_k, 1^\lambda)) \leq negl(\lambda).$$

Then, by Equation 2 we have that

$$\Delta((\mathtt{evk}, \mathtt{Eval}_{\mathcal{E}}(C_{f_1}, \mathtt{ct}_1, \ldots, \mathtt{ct}_k, \mathtt{evk}), \mathtt{ct}_1, \ldots, \mathtt{ct}_k, 1^\lambda),$$
$$(\mathtt{evk}, \mathtt{Eval}_{\mathcal{E}}(C_{f_2}, \mathtt{ct}_1, \ldots, \mathtt{ct}_k, \mathtt{evk}), \mathtt{ct}_1, \ldots, \mathtt{ct}_k, 1^\lambda)) \leq negl(\lambda).$$

Assuming that Definition 21 holds, we can construct a simulator that gives the output as required in Definition 24[sh] in the following way. Let $\mathcal{F}$ be the set of all functions such that $C_f \in \mathcal{C}_{\mathcal{E}}$. Since this is information about the scheme itself, it can be assumed to be public. We then construct the simulator as follows.

For each $f' \in \mathcal{F}$, it calculates $\mathtt{Eval}(C_{f'}, \mathtt{ct}_1, \ldots, \mathtt{ct}_k, \mathtt{evk})$, where $\mathtt{ct}_i = \mathtt{Encode}(\mathtt{pk}, m_i)$. Then $\Delta(\mathtt{Eval}(C_{f'}, \mathtt{ct}_1, \ldots, \mathtt{ct}_k, \mathtt{evk}), \mathtt{Eval}(C_f, \mathtt{ct}_1, \ldots, \mathtt{ct}_k, \mathtt{evk}))$, and outputs $\mathtt{Eval}(C_{f'}, \mathtt{ct}_1, \ldots, \mathtt{ct}_k, \mathtt{evk})$ if this statistical distance is negligible. Since the simulator only needs to be unbounded, we know that it will eventually reach this point, either because it finds f' different from f such that $f'(m) = f(m)$ or else because it finds f. This further illustrates why an unbounded simulator weakens the definition, since it may output a correct solution only by guessing the correct circuit. In this case, such a scheme would satisfy the circuit privacy definition of 24.

The fact that Definition 24 implies Definition 23[sh] is easily seen. Definition 23[sh] restricts to only a particular representation of a circuit, and allows the simulator to additionally see an evaluation of f on simulated plaintexts. Therefore, the simulator receives more information and the definition is weaker. Definition 24[sh] also implies Definition 25[sh], since again the simulator receives the additional information from the leakage function Φ on f. The claim follows.

□

We continue by proving relations between the malicious Definitions 23 - 25. The proof is very similar to the semi-honest case.

Lemma 6. *The following implications hold*

$$24 \Rightarrow 23$$
$$\Downarrow$$
$$25$$

Proof. Definitions 24 and 25 are equivalent, but for the fact that in Definition 25 the simulator additionally obtains information from a leakage function on the circuit C_f. The implication is therefore evident: if the simulator manages to output a correct ciphertext without the information from the leakage function, it will also be able to do so with this information.

The implication between Definition 24 and 23 can be seen in a similar way. In Definition 23, the simulator gets as an additional information an evaluation of a fixed representation of the circuit on a simulated message (a correct decryption may not be available due to the ciphertexts being potentially maliciously

formed). Therefore, if the simulator in Definition 24 was able to output a valid ciphertext without having seen the plaintext evaluation of the circuit, it will be able to do the same upon seeing it. □

The relation between Definitions 23 and 25 depends on the concrete instantiation of the leakage function. If the leakage function leaks the plaintext evaluation of the circuit and potentially additional information, then Definition 23 implies Definition 25. If the leakage function leaks different information, then the relation between the two definitions is unclear.

B   Proofs of Lemmas 1 - 4

Proof of Lemma 1

Proof. Let $\mathtt{ct}'_0$ and $\mathtt{ct}'_1$ be two ciphertexts whose levels have already be matched. That is, if $i = j, \mathtt{ct}'_0 = \mathtt{ct}_0, \mathtt{ct}'_1 = \mathtt{ct}_1$, if $i > j, \mathtt{ct}'_0 = \mathtt{ModSwitch}((\mathtt{ct}_0, Q_i), Q_j), \mathtt{ct}'_1 = \mathtt{ct}_1$ or otherwise $\mathtt{ct}'_0 = \mathtt{ct}_0, \mathtt{ct}'_1 = \mathtt{ModSwitch}((\mathtt{ct}_1, Q_j), Q_i)$.

Then we can denote the noise $\nu_i(\mathtt{ct})$ of $\mathtt{ct}_{\mathrm{add}}$ as

$$\nu_{\min(i,j)}(\mathtt{ct}_{\mathrm{add}}) = \nu_{\min(i,j)}(\mathtt{ct}'_0) + \nu_{\min(i,j)}(\mathtt{ct}'_1).$$

We can include the level adjustment into this expression of the noise as follows

$$
\begin{aligned}
\nu_{\min(i,j)}(\mathtt{ct}_{\mathrm{add}}) =& (1 - \chi_{\{j+1,\ldots,L\}}(i))\nu_i(\mathtt{ct}_0) + \chi_{\{j+1,\ldots,L\}}(i)\nu_j(\mathtt{ModSwitch}((\mathtt{ct}_0, Q_i), Q_j)) \\
&+ \chi_{\{j,\ldots,L\}}(i)\nu_j(\mathtt{ct}_1) + (1 - \chi_{\{j,\ldots,L\}}(i))\nu_i(\mathtt{ModSwitch}((\mathtt{ct}_1, Q_j), Q_i)).
\end{aligned}
$$

□

Proof of Lemma 2

Proof. Let $\mathtt{ct}'_0, \mathtt{ct}'_1$ be the ciphertexts after the level adjustment. The noise of the ciphertext $\mathtt{ct}_{\text{pre-mult}}$ can therefore be given with respect to $\mathtt{ct}'_0$ and $\mathtt{ct}'_1$ as

$$
\begin{aligned}
\nu_i(\mathtt{ct}_{\text{pre-mult}}) =& \mathtt{ct}'_0[0]\mathtt{ct}'_1[0] + (\mathtt{ct}'_0[1]\mathtt{ct}'_1[0] + \mathtt{ct}'_0[0]\mathtt{ct}'_1[1])s + \mathtt{ct}'_0[1]\mathtt{ct}'1[1]s^2 - m_0 m_1 \\
=& (\mathtt{ct}'_0[0] + \mathtt{ct}'_0[1]s - m_0)(\mathtt{ct}'_1[0] + \mathtt{ct}'_1[1]s - m_1) \\
&+ m_1(\mathtt{ct}'_0[0] + \mathtt{ct}'_0[1]s - m_0) + m_0(\mathtt{ct}'_1[0] + \mathtt{ct}'_1[1]s - m_1) \\
=& \nu_i(\mathtt{ct}'_0)\nu_i(\mathtt{ct}'_1) + m_1\nu_i(\mathtt{ct}'_0) + m_0\nu_i(\mathtt{ct}'_1).
\end{aligned}
$$

As for addition, the level adjustment outputs the following. If $i > j$,

$$
\begin{aligned}
\mathtt{ct}'_0 &= \mathtt{ModSwitch}((\mathtt{ct}_0, Q_i), Q_j) \\
\mathtt{ct}'_1 &= \mathtt{ct}_1.
\end{aligned}
$$

If $j > i$

$$\mathtt{ct}'_0 = \mathtt{ct}_0$$

39

$$\mathtt{ct}'_1 = \mathtt{ModSwitch}((\mathtt{ct}_1, Q_j), Q_i).$$

If $i = j$

$$\mathtt{ct}'0 = \mathtt{ct}_0$$
$$\mathtt{ct}'_1 = \mathtt{ct}_1.$$

Using the characteristic function, the critical quantity of the ciphertext can therefore be given as

$$\nu_{\min(i,j)}(\mathtt{ct}_{\text{pre-mult}})$$
$$= \big((1 - \chi_{\{j+1,\dots,L\}}(i))\nu_i(\mathtt{ct}_0) + \chi_{\{j+1,\dots,L\}}(i)\nu_j(\mathtt{ModSwitch}((\mathtt{ct}_0, Q_i), Q_j))\big)$$
$$\cdot \big(\chi_{\{j,\dots,L\}}(i)\nu_j(\mathtt{ct}_1) + (1 - \chi_{\{j,\dots,L\}}(i))\nu_i(\mathtt{ModSwitch}((\mathtt{ct}_1, Q_j), Q_i))\big)$$
$$+ m_0\big(\chi_{\{j,\dots,L\}}(i)\nu_j(\mathtt{ct}_1) + (1 - \chi_{\{j,\dots,L\}}(i))\nu_i(\mathtt{ModSwitch}((\mathtt{ct}_1, Q_j), Q_i))\big)$$
$$+ m_1\big((1 - \chi_{\{j+1,\dots,L\}}(i))\nu_i(\mathtt{ct}_0) + \chi_{\{j+1,\dots,L\}}(i)\nu_j(\mathtt{ModSwitch}((\mathtt{ct}_0, Q_i), Q_j))\big).$$

As can be seen in Appendix D of [CNP23b] we then have for the noise of the ciphertext $\mathtt{ct}$ that is output as the final result

$$\nu_{sp}(\mathtt{ct}) = \frac{Q_{sp}}{Q_{\min(i,j)}}\nu_{\min(i,j)}(\mathtt{ct}_{\text{pre-mult}}) + t\sum_{j=1}^{\ell} \mathtt{ct}_j^{\text{pre-mult}}[2]e_{2,j}.$$

Finally, after modulus switching, we obtain the following for the noise term

$$\nu_i(\mathtt{MULT}(\mathtt{ct}_0, \mathtt{ct}_1)) = \left\lfloor \frac{Q_{i-1}}{Q_{sp}}\mathtt{ct}[0] \right\rceil_t + \left\lfloor \frac{Q_{i-1}}{Q_{sp}}\mathtt{ct}[1] \right\rceil_t s - \left\lfloor \frac{Q_{i-1}}{Q_{sp}}m_0m_1 \right\rceil_t$$

$$= \frac{Q_{i-1}}{Q_{sp}}\frac{Q_{sp}}{Q_i}\nu_i(\mathtt{ct}_{\text{pre-mult}}) + \frac{Q_{i-1}}{Q_{sp}}\sum_{j=1}^{\ell} t\mathtt{ct}_j^{\text{pre-mult}}[2]e_{2,j}$$

$$+ \tau_0 + \tau_1 s + \tau_2.$$

We can approximate the second term by

$$||\frac{Q_{i-1}}{Q_i}\sum_{j=1}^{\ell} \mathtt{ct}_{\text{pre-mult}}^{(j)}[2]e_{2,j}||_\infty \leq \frac{Q_{i-1}}{Q_i}t\sum_{j=1}^{\ell} ||\mathtt{ct}_j^{\text{pre-mult}}[2]||_\infty ||e_{2,j}||_\infty$$

$$\leq \frac{Q_{i-1}}{Q_{sp}}t\ell D_{max}^\star B_e$$

$$= \frac{Q_{i-1}}{Q_i}\frac{t\ell D_{max}^\star B_e}{k}.$$

Due to Requirement 2, this term is negligible compared to the first term. We can therefore approximate the noise after a multiplication via

$$\nu_i(\texttt{MULT}(\texttt{ct}_0, \texttt{ct}_1)) \approx \frac{Q_{i-1}}{Q_i} \nu_i(\texttt{ct}_{\text{pre-mult}}) + \tau_0 + \tau_1 s + \tau_2.$$

We can approximate $\|\nu_i(\texttt{ct}_{\text{pre-mult}})\|_\infty$ as follows.

$$
\begin{aligned}
\|\nu_i(\texttt{ct}_{\text{pre-mult}})\|_\infty &= \|\nu_i(\texttt{ct}_0')\nu_i(\texttt{ct}_1') + m_1\nu_i(\texttt{ct}_0') + m_0\nu_i(\texttt{ct}_1')\|_\infty \\
&\le N\,\|\nu_i(\texttt{ct}_0')\|_\infty\,\|\nu_i(\texttt{ct}_1')\|_\infty + N\,\|m_1\|_\infty\,\|\nu_i(\texttt{ct}_0')\|_\infty \\
&\quad + N\,\|m_0\|_\infty\,\|\nu_i(\texttt{ct}_1')\|_\infty \\
&\le N\left(\sqrt{\frac{Q_i}{Q_{i-1}N} + t^2} - t\right)^2 + 2Nt\left(\sqrt{\frac{Q_i}{Q_{i-1}N} + t^2} - t\right) \\
&= \frac{Q_i}{Q_{i-1}},
\end{aligned}
$$

where the the fourth line holds since $\sqrt{\frac{Q_i}{Q_{i-1}N^3} + t^2} - t$ is one solution of $Nx^2 + 2Ntx - \frac{Q_i}{Q_{i-1}}$ and the third line holds due to Requirement 1. Indeed, this is where Requirement 1 originates from. $\qquad\square$

Proof of Lemma 3

Proof. We have

$$
\begin{aligned}
\nu_i(\texttt{ADDCONST}((\texttt{ct}, Q_i), \texttt{const})) =&\texttt{ADDCONST}((\texttt{ct}, Q_i), \texttt{const})[0] \\
&+ \texttt{ADDCONST}((\texttt{ct}, Q_i), \texttt{const})[1]s - m - \texttt{const} \\
=&\texttt{ct}[0] + \texttt{const} + \texttt{ct}[1]s - m - \texttt{const} \\
=&\nu_i(\texttt{ct}),
\end{aligned}
$$

and

$$
\begin{aligned}
\nu_i(\texttt{MULTCONST}((\texttt{ct}, Q_i), \texttt{const})) =&\texttt{MULTCONST}((\texttt{ct}, Q_i), \texttt{const})[0] \\
&+ \texttt{MULTCONST}((\texttt{ct}, Q_i), \texttt{const})[1]s - \texttt{const} \cdot m \\
=&\texttt{const} \cdot (\texttt{ct}[0] + \cdot\texttt{ct}[1]s - m) \\
=&\texttt{const} \cdot \nu_i(\texttt{ct}).
\end{aligned}
$$

$\qquad\square$

Proof of Lemma 4

Proof. The automorphism operation only permutes the slots, but does not change the values of the ciphertext. Hence, no new error is introduced. The key switching then results in the following noise according to Appendix D in [CNP23b].

$$\nu_i(\texttt{KeySwitch}(\sigma(\texttt{ct}), \texttt{sk}, \sigma(\texttt{sk}))) = \frac{Q_{sp}}{Q_i}\nu_i(\sigma(\texttt{ct})) + t\sum_{j=1}^{\ell}\sigma(\texttt{ct}_1^{(j)})e_{2,j}.$$

Therefore, after modulus switching, we have

$$
\begin{aligned}
\nu_i(\texttt{AUTOMORPHISM}((\texttt{ct}, Q_i))) = & \left\lfloor \frac{Q_{i-1}}{Q_{sp}} \texttt{KeySwitch}(\sigma(\texttt{ct}), \texttt{sk}, \sigma(\texttt{sk}))[0] \right\rceil \\
& + \left\lfloor \frac{Q_{i-1}}{Q_{sp}} \texttt{KeySwitch}(\sigma(\texttt{ct}), \texttt{sk}, \sigma(\texttt{sk}))[1] \right\rceil s \\
& - \left\lfloor \frac{Q_{i-1}}{Q_{sp}} \sigma(m) \right\rceil_t \\
= & \frac{Q_{i-1}}{Q_i} \nu_i(\sigma(\texttt{ct})) + \frac{Q_{i-1}}{Q_{sp}} t \sum_{j=1}^{\ell} \sigma(\texttt{ct}_1^{(j)}) e_{2,j} \\
& + \tau_0 + \tau_1 s + \tau_2 \\
\approx & \frac{Q_{i-1}}{Q_i} \nu_i(\sigma(\texttt{ct})) + \tau_0 + \tau_1 s + \tau_2,
\end{aligned}
$$

where the last line holds if Requirement 2 holds. Then, since $\nu_i(\sigma(\texttt{ct})) = \nu_i(\texttt{ct})$, if Requirement 1 holds, we have

$$
\nu_i(\texttt{AUTOMORPHISM}((\texttt{ct}, Q_i))) \approx \tau_0 + \tau_1 s + \tau_2.
$$

$\square$

C   Extension of Theorem 2 to Multiplication by Plaintext Polynomials

Corollary 1. *Let* `MULTCONST` *be such that it also accepts plaintext polynomials of degree bigger than* 0. `BGV'` *modified in this way is levelled noise circuit private, as long as* $t \leq \sqrt{\frac{Q_i}{Q_{i-1}} N^3} \frac{2}{\kappa ||C||_\infty}$, *where* C *is the polynomial product of all plaintexts with which the ciphertexts are multiplied.*

Proof. Let $\texttt{ptxt}_1, \ldots \texttt{ptxt}_\ell$ be plaintexts used in the evaluation of f, not necessarily all distinct. Let f be any mathematical function on ciphertexts $\texttt{ct}_1, \ldots, \texttt{ct}_k$, consisting of m additions and ℓ multiplications by plaintexts, in any order. Then we can bound the final value of f by

$$
||f(\texttt{ct}_1, \ldots, \texttt{ct}_k)||_\infty \leq \left|\left| C \sum_{j=1}^{m} \texttt{ct}_j \right|\right|_\infty,
$$

where $C = \prod_{u=1}^{\ell} \texttt{ptxt}_u$. This holds true since the value of $||f(\texttt{ct}_1, \ldots, \texttt{ct}_k)||_\infty$ will not wrap-around modulo Q_i since otherwise there would be a decryption error. Otherwise, this result is due to just pulling out all single multiplications at the start of the computation and upper-bounding the result in this way.

For additions, we have seen from the proof above that we can bound the critical quantity of any representation of a purely additive circuit consisting of m additions by $m\|\tau_0 + \tau_1 s + \tau_2\|_\infty$.

We therefore can bound the critical quantity of any representation of f by

$$
\begin{aligned}
\|\nu_i(f(\mathtt{ct}_1, \ldots, \mathtt{ct}_k))\|_\infty &\leq \|C\|_\infty \, m \, \|\tau_0 + \tau_1 s + \tau_2\|_\infty \\
&\leq \|C\|_\infty \, m \frac{th}{2} \\
&\leq \|C\|_\infty \, m \frac{tN}{2} \\
&\leq \|C\|_\infty \, mN \sqrt{\frac{Q_i}{Q_{i-1} N^3}} \frac{2}{2\kappa \|C\|_\infty} \\
&= \frac{m}{\kappa} \sqrt{Q_i} Q_{i-1} N.
\end{aligned}
$$

We therefore see that, for a correct choice of parameters, modulus switching will reduce the accumulated noise to the rounding noise, and that `BGV'` therefore continues to be circuit private. □