Additively Homomorphic Encryption with t-Operand Multiplications

Carlos Aguilar Melchor[1], Philippe Gaborit[1], and Javier Herranz[2]

[1] XLIM-DMI, Université de Limoges,
123, av. Albert Thomas
87060 Limoges Cedex, France
{carlos.aguilar,philippe.gaborit}@xlim.fr
[2] DMA-IV, Universitat Politècnica de Catalunya,
C. Jorgi Girona, 1-3, Mòdul C-3,
08034 Barcelona, Spain
jherranz@ma4.upc.edu

Abstract. Homomorphic encryption schemes are an essential ingredient to design protocols where different users interact in order to obtain some information from the others, at the same time that each user keeps private some of his information. When the algebraic structure underlying these protocols is complicated, then standard homomorphic encryption schemes are not enough, because they do not allow to compute at the same time additions and products of plaintexts through the manipulation of ciphertexts.

In this work we define a theoretical object, t-chained encryption schemes, which can be used to compute additions and products of t integer values, by ciphertext manipulation. Efficient solutions have been previously proposed only for the case $t = 2$. Our solution is not only theoretical: we show that some existing IND-CPA secure (pseudo)homomorphic encryption schemes (some of them based on lattices) can be used to implement in practice the concept of t-chained encryption scheme.

Keywords: homomorphic encryption, lattices, secure function evaluation.

1 Introduction

In 1978 Rivest, Adleman and Dertouzos introduced *privacy homomorphisms* [1], which are public key encryption schemes that map an operation on the ciphertext space to an operation on the plaintext space. Such schemes allow therefore to modify plaintexts through ciphertext manipulation. Privacy homomorphisms are also known as *homomorphic (public-key) encryption schemes* and are usually classified by the operation they allow to compute on the plaintext space: *multiplicatively* homomorphic schemes, such as RSA [2] or El Gamal [3], allow to compute a product; and *additively* homomorphic schemes, such as Paillier [4], allow to compute a sum. An homomorphic public-key encryption scheme that allows to compute both sums and products of plaintexts through ciphertext manipulation is called an *algebraic* homomorphic public-key encryption scheme.

Additively homomorphic public-key encryption schemes have many applications such as computing with encrypted functions (or with encrypted data), multi-party computation, zero-knowledge proofs, oblivious transfer, commitment schemes, etc. A large survey of these applications (and an associated bibliography) is done by Rappe in [5]. Rappe also highlights that being able to arbitrarily compute sums and products of encrypted plaintexts (with a secure algebraic homomorphic scheme, for example) would result in a dramatic improvement on performance and functionality of many applications.

Unfortunately, only a few algebraic homomorphic public-key encryption schemes have been proposed in the literature and all of them have been broken. Fellows and Koblitz proposed Polly Cracker [6] which cannot be considered secure [7], and Grigoriev and Ponomarenko proposed a scheme [8] which was broken in [9]. As Fontaine and Galand note in their survey on homomorphic encryption [10], no satisfactory solution has been proposed so far, and given the conjecture by Boneh and Lipton that algebraic homomorphic encryption schemes cannot be secure [11], the question of their existence is still open (see also this and other interesting questions on the subject in [12]).

For the case of symmetric cryptography, Domingo-Ferrer proposed two schemes [13, 14] which were broken in [15, 16]. A secret-key algebraic homomorphic encryption scheme has been recently proposed by Armknecht and Sadeghi [17]. Their scheme is derived from a Private Information Retrieval (PIR) scheme by Kiayias and Yung [18] whose security was based on the Polynomial Reconstruction Problem. This problem was broken in [19] and [20], and Armknecht and Sadeghi's proposal is therefore based on a variation that resists to these attacks. The main issue with the encryption scheme they propose is that security against an attacker with access to r ciphertexts is only ensured if ciphertext size is in $O(e^r)$. This limits considerably the number of applications. In any case, the work in [17] is very interesting from a theoretical point of view and may lead to code-based homomorphic schemes with very useful properties.

In the scenario of asymmetric cryptography, the most important contribution towards the construction of an algebraic homomorphic scheme was done by Boneh, Goh and Nissim. In [21] they propose a new secure encryption scheme which is additively homomorphic and, at the same time, allows to compute products of plaintexts by ciphertext manipulation. However, this proposal has one fundamental limitation: only products of plaintexts pairs can be computed and the resulting ciphertexts can be used to compute sums but not products any more. The span of applications of algebraic homomorphic encryption is so large that, despite this limitation, the scheme in [21] has lead to a huge number of papers which discuss, cite or are based on the mechanism proposed therein.

Our Contribution. Our motivating goal was to get rid of the limitation of Boneh, Goh and Nissim's encryption scheme, allowing the computation of plaintext products by ciphertext manipulation even if one of the ciphertexts is the result of another product computation. To do this, we define a theoretical cryptographic object, that we denote as *t-chained encryption scheme*, which is constructed through the combination of multiple (pseudo)homomorphic encryption schemes satisfying some conditions. If the combined encryption schemes are IND-CPA secure (see a formal definition of this concept in Appendix A), then the resulting t-chained encryption scheme is also IND-CPA secure. We show how to compute products with t-chained schemes so that each ciphertext can be used iteratively in up to $t-1$ product computations before it can only be used to compute sums.

After that, we review some existing (pseudo)homomorphic schemes that can be used as components to realize in practice this theoretical concept of t-chained encryption schemes. We study in detail two possible particular instances to illustrate this construction. Both instances are IND-CPA secure: security of the first one is reduced to well-known problems such as uSVP (with a worst-case/average-case reduction) and the subgroup decision problem; security of the second one is reduced to a more adhoc problem, the Differential Knapsack Vector Problem. Some of the employed encryption schemes must be necessarily based on lattices, which has an effect on the efficiency of the resulting t-chained encryption schemes (in particular, regarding the size of the ciphertexts). However, any future advance in the area of lattice-based (pseudo)homomorphic schemes will immediately have an impact on the efficiency of our solutions.

Application to Secure Function Evaluation. Some important applications of homomorphic encryption schemes can be seen as particular cases of a primitive: *secure function evaluation* (SFE). In (a simplified version of) SFE, a user Alice has a function f and a user Bob some data x. Bob must obtain $f(x)$ without Alice learning anything about x (not even $f(x)$) through a protocol with the minimum possible interaction. If moreover, Bob does not learn anything about f (besides $f(x)$) we say the evaluation is symmetrically secure. Due to length constraints we just consider in this paper Bob's security, and describe a construction that does not ensure symmetric security. In Appendix D we provide a few more details on this point and highlight how the construction can be modified to provide symmetric security if the underlying encryption schemes have the correct properties.

An additively homomorphic encryption scheme allows to securely evaluate multivariate polynomials $F(x_1, \ldots, x_n)$ of degree $t = 1$: Bob, who holds secret inputs $a_1, \ldots, a_n$, generates a key pair for an additively homomorphic scheme, encrypts each input, and sends to Alice the resulting ciphertexts, along with the public key. Alice can use the additively homomorphic properties of the scheme to compute an encryption of $F(a_1, \ldots, a_n)$, which is sent back to Bob. Finally, Bob decrypts and obtains the desired value. If the encryption scheme is IND-CPA secure Bob's security is ensured.

It is often necessary to evaluate higher degree polynomials either to improve performance or to unlock new applications (see [5]), and additively homomorphic encryption schemes do not provide a straight solution to these situations. A fully algebraic homomorphic public-key encryption scheme would solve this problem, but as we have noted before, there is not any such scheme right now.

The encryption scheme proposed by Boneh, Goh and Nissim allows to evaluate polynomials of degree $t = 2$, as long as the output $F(a_1, \ldots, a_n)$ is a small number (the computational cost of decryption is linear on this number). For this reason they propose to use it for 2-DNF formula evaluation, which can be derived from polynomial evaluation with a decryption cost in $O(1)$, as long as ciphertext size is at least in $O(\log L)$, being L the number of sums done.

Sander, Young and Yung propose an alternative approach [22] based on an uncommon usage of additively homomorphic encryption schemes, which allows to compute any boolean formula. The major drawback of their approach is that a given ciphertext's size is exponential in the number of operations it has undergone and thus the scheme can only be used to evaluate functions with logarithmic depth. Many applications need linear depth functions to be evaluated and thus, despite the generality of the solution provided by Sander, Young and Yung, the number of applications based on this scheme in the literature is similar to the case of Boneh, Goh and Nissim's scheme.

In this paper we focus on SFE to highlight our construction's practicality. We will see that t-chained schemes, introduced in this work, can be used to evaluate any polynomial of degree t ensuring Bob's security as long as the underlying encryption schemes are IND-CPA secure.

Organization of the Paper. In Section 2 we describe the general idea and a simple example of our construction. We recall in Section 3.1 some basic concepts on (pseudo)homomorphic encryption schemes. The main part of our work is presented in Section 3. We first recall some basic concepts on (pseudo)homomorphic encryption schemes, and we prove some technical results involving (pseudo)homomorphisms, which can be of independent interest. After that, the concept of t-chained encryption scheme is introduced. In Section 4 we explain how to use t-chained encryption schemes to securely evaluate a t-degree polynomial; we discuss the security of the protocol and we briefly present some of its applications in Appendix C. We finally give, in Section 5, some examples of t-chained encryption schemes (for $t = 2$ and $t = 3$) that can be obtained by using some existing (pseudo)homomorphic encryption schemes.

2 Overview of Our Solution

2.1 Basic Idea

From a theoretical point of view, an algebraic encryption scheme would have an encryption function $\mathcal{E}$ which is a ring homomorphism between $(\mathbb{Z}_s, +, \cdot)$ and some ring $(R, +_R, \cdot_R)$.[3] Some encryption functions, as for example ElGamal's [3], ensure naturally a group homomorphism between $(\mathbb{Z}_s^*, \cdot)$ and the group of associated ciphertexts. Other encryption functions, as Paillier's [4], ensure a group homomorphism between $(\mathbb{Z}_s, +)$ and the group of ciphertexts. As noted in the introduction, all the cryptosystems with an encryption function that is a ring homomorphism between $(\mathbb{Z}_s, +, \cdot)$ and some ring have been broken.

The proposal of Boneh, Goh and Nissim is very interesting from a theoretical point of view. Indeed, their cryptosystem provides two group homomorphic encryption functions $\mathcal{E}_1$ and $\mathcal{E}_2$ such that

$$\begin{tikzcd}
\mathcal{E}_1 : (\mathbb{Z}_s, +) \arrow[r] & (G_1, \times) \arrow[d, "e(\cdot,\cdot)"] \\
\mathcal{E}_2 : (\mathbb{Z}_s, +) \arrow[r] & (G_2, \times)
\end{tikzcd}$$

[3] In fact, the homomorphism must be between $(\mathbb{Z}_s, +, \cdot) \times (\mathit{\Omega}, +_{\mathit{\Omega}}, \cdot_{\mathit{\Omega}})$ and some ring $(R, +_R, \cdot_R)$ as the encryption scheme is probabilistic, but the randomization is often hidden to simplify notations.

where $\mathcal{E}_2(x) = e(\mathcal{E}_1(x), 1)$ and $e(\cdot,\cdot)$ is a bilinear map such that $e(\mathcal{E}_1(a_1), \mathcal{E}_1(a_2)) = e(\mathcal{E}_1(a_1 \cdot a_2), 1) = \mathcal{E}_2(a_1 \cdot a_2)$. Starting from encryptions $\mathcal{E}_1(a_1)$ and $\mathcal{E}_1(a_2)$, it is thus possible to obtain encryptions (according to $\mathcal{E}_2$) of both a_1+a_2 and $a_1 \cdot a_2$. The bilinear map $e(\cdot,\cdot)$ is a Weil pairing over an elliptic curve. Unfortunately, once in G_2 it is not possible to use a second pairing in order to compute more products. Our motivating goal was to obtain a chain of homomorphic encryption functions and bilinear pairings such that for any t we have:

$$\begin{tikzcd}
\mathcal{E}_1 : (\mathbb{Z}_s, +) \arrow[r] & (G_1, \times) \arrow[d, "e_1(\cdot,\cdot)"] \\
\mathcal{E}_2 : (\mathbb{Z}_s, +) \arrow[r] & (G_2, \times) \arrow[d, "e_2(\cdot,\cdot)"] \\
\cdots\cdots\cdots & {} \arrow[d, "e_{t-1}(\cdot,\cdot)"] \\
\mathcal{E}_t : (\mathbb{Z}_s, +) \arrow[r] & (G_t, \times)
\end{tikzcd}$$

With such a structure (and the right properties on the bilinear functions), the inductive limit of the disjoint union $R_t = \amalg_{i=1}^{t} G_i$ is isomorphic to a ring. Even if such a structure does not represent a ring and an associated ring homomorphism, it provides at least a series of groups and group homomorphisms which have a ring and a ring homomorphism as a limit. Unfortunately, our construction is slightly weaker than the aforementioned one, and such a ring cannot be defined. Nevertheless, from a practical point of view we are still able to compute as with such a structure.

2.2 A Simple Construction

A real construction of a scheme allowing to do t multiplications is a little bit complex and is described in the following sections, but we will present here an 'ideal world' construction as its simplicity allows to grab the principles our construction is based on. Suppose that, for any $s \in \mathbb{Z}^+$, we can obtain a cryptosystem with an encryption function $\mathcal{E} : \mathbb{Z}_s \to \mathbb{Z}_{s'}$, such that $\mathcal{E}(a_1) + \mathcal{E}(a_2) = \mathcal{E}(a_1 + a_2)$. Starting from a value $s_1 \in \mathbb{Z}^+$, it is then possible to:

- obtain $\mathcal{E}_1 : \mathbb{Z}_{s_1} \to \mathbb{Z}_{s_2}$ such that $\mathcal{E}_1(a_1) + \mathcal{E}_1(a_2) = \mathcal{E}_1(a_1 + a_2)$;
- obtain $\mathcal{E}_2 : \mathbb{Z}_{s_2} \to \mathbb{Z}_{s_3}$ such that $\mathcal{E}_2(a_1) + \mathcal{E}_2(a_2) = \mathcal{E}_2(a_1 + a_2)$.

Because of the chosen parameters we have $\mathcal{E}_2(\mathcal{E}_1(a_1)) + \mathcal{E}_2(\mathcal{E}_1(a_2)) = \mathcal{E}_2(\mathcal{E}_1(a_1) + \mathcal{E}_1(a_2)) = \mathcal{E}_2(\mathcal{E}_1(a_1 + a_2))$. Denoting $\mathcal{E} = \mathcal{E}_2 \circ \mathcal{E}_1$, we have $\mathcal{E}(a_1) + \mathcal{E}(a_2) = \mathcal{E}(a_1 + a_2)$. Moreover,

$$\mathcal{E}_1(a_1) \cdot \mathcal{E}_2(a_2) = \mathcal{E}_2(a_2 \cdot \mathcal{E}_1(a_1)) = \mathcal{E}_2(\mathcal{E}_1(a_1 \cdot a_2)) = \mathcal{E}(a_1 \cdot a_2),$$

the first two equalities being verified as the homomorphic property of $\mathcal{E}_2$ (resp. $\mathcal{E}_1$) implies $k \cdot \mathcal{E}_2(x) = \mathcal{E}_2(k \cdot x)$ (resp. $k \cdot \mathcal{E}_1(x) = \mathcal{E}_1(k \cdot x)$). We thus have:

$$\mathcal{E}(a_1) + \mathcal{E}(a_2) = \mathcal{E}(a_1 + a_2) \text{ and } \mathcal{E}_1(a_1) \cdot \mathcal{E}_2(a_2) = \mathcal{E}(a_1 \cdot a_2)$$

Of course, this construction can be generalized if we obtain additive schemes $\mathcal{E}_1, \ldots, \mathcal{E}_t$ such that

$$(\mathbb{Z}_{s_1}, +) \xrightarrow{\mathcal{E}_1} (\mathbb{Z}_{s_2}, +) \xrightarrow{\mathcal{E}_2} (\mathbb{Z}_{s_3}, +) \xrightarrow{\mathcal{E}_3} \ldots \xrightarrow{\mathcal{E}_t} (\mathbb{Z}_{s_{t+1}}, +).$$

Unfortunately, the existing additive schemes that we could use to realize this 'ideal world' construction only provide a pseudohomomorphic property (properly defined in the next section) and not a real homomorphism. Moreover, in many cases we do not have $\mathcal{E} : \mathbb{Z}_s \to \mathbb{Z}_{s'}$, but $\mathcal{E} : \mathbb{Z}_s \to \mathbb{Z}_{s'}^{n'}$, or $\mathcal{E} : \mathbb{Z}_s^{n} \to \mathbb{Z}_{s'}^{n'}$. In fact, we will use a cryptosystem such that $\mathcal{E} : \mathbb{Z}_s \to \mathbb{Z}_{s'}$ (s' being very large) in our practical constructions in Section 5. However, using it directly leads to a very inefficient construction. We will thus modify it to obtain an encryption function $\mathcal{E} : \mathbb{Z}_s \to \mathbb{Z}_{s''}^{n''}$ with a small s'' and a large n''.

In Section 3.1, we provide basic definitions and in Sections 3.2 and 3.3 we show how to modify the existing encryption schemes to fit our needs. Then, in Section 3.4 we provide a general construction derived from the intuitive one that we have just presented.

3 t-Chained Pseudohomomorphic Encryption Schemes

In this section we define t-chained encryption schemes, the basic tool of our protocols. In order to do this we first define L-pseudohomomorphisms, and prove some of their properties through a small set of lemmas and propositions. The (sketched) proofs are given in Appendix B.1, except when they deal with security issues in which case they have been included in the main text (when not trivial).

3.1 Preliminaries: (Pseudo)Homomorphic Encryption

A public key encryption scheme $PKE = (\mathcal{KG}, \mathcal{E}, \mathcal{D})$ consists of three probabilistic and polynomial time algorithms. The key generation algorithm $\mathcal{KG}$ takes as input a security parameter (for example, the desired length for the secret key) and outputs a pair (sk, pk) of secret and public keys. The encryption algorithm takes as input a plaintext m corresponding to some set of plaintexts $\mathcal{M}$, some randomness $r \in \mathcal{R}$ and a public key pk, and outputs a ciphertext $c = \mathcal{E}_{pk}(m, r) \in \mathcal{C}$, where $\mathcal{C}$ is the ciphertexts' space. Finally, the decryption algorithm takes as input a ciphertext and a secret key, and gives a plaintext $m = \mathcal{D}_{sk}(c)$ as output. In the rest of the paper, for simplicity of the notation, we will only include explicitly the randomness as an input of the encryption functions and the public key as their index when necessary.

We say that $PKE = (\mathcal{KG}, \mathcal{E}, \mathcal{D})$ is *pseudohomomorphic* if $\mathcal{M}$ and $\mathcal{C}$ have both a group structure (with operations $\oplus_{\mathcal{M}}$ and $\oplus_{\mathcal{C}}$, respectively; we will write $(\mathcal{M}, \oplus_{\mathcal{M}})$ and $(\mathcal{C}, \oplus_{\mathcal{C}})$), and the property

$$\mathcal{D}\big(\mathcal{E}(m_1) \oplus_{\mathcal{C}} \mathcal{E}(m_2)\big) \;=\; m_1 \oplus_{\mathcal{M}} m_2$$

holds for any $m_1, m_2 \in \mathcal{M}$.

Again the index of the operation will only be included explicitly when necessary and often we will just write $\mathcal{D}\big(\mathcal{E}(m_1) \oplus \mathcal{E}(m_2)\big) \;=\; m_1 \oplus m_2$. This basic pseudohomomorphic property does not imply $\mathcal{E}(m_1) \oplus \mathcal{E}(m_2) = \mathcal{E}(m_1 \oplus m_2)$, but just $\mathcal{E}(m_1) \oplus \mathcal{E}(m_2) \in \mathcal{D}^{-1}(m_1 \oplus m_2)$. This is important as often the function $\mathcal{E} : \mathcal{M} \to \mathcal{C}$ is not surjective. In order to avoid cumbersome notations, $\tilde{\mathcal{E}}(x)$ will represent an element of $\mathcal{D}^{-1}(x)$ just as $\mathcal{E}(x)$ represents an element of $\{\mathcal{E}(x, r) | r \in \mathcal{R}\}$. We thus have $\mathcal{E}(m_1) \oplus \mathcal{E}(m_2) = \tilde{\mathcal{E}}(m_1 \oplus m_2)$. With these ideas in mind, one can consider the following definition.

Definition 1. *A public key encryption scheme which satisfies $\mathcal{E}(m_1) \oplus \ldots \oplus \mathcal{E}(m_k) = \tilde{\mathcal{E}}(m_1 \oplus \ldots \oplus m_k)$ for all $k \leq L$ and all k-tuple $(m_1, \ldots, m_k) \in \mathcal{M}^k$ is said to be L-pseudohomomorphic.*

If $\mathcal{E}(m_1) \oplus \mathcal{E}(m_2) = \mathcal{E}(m_1 \oplus m_2)$, then we can iteratively apply this result to deduce that $\mathcal{E}(m_1) \oplus \ldots \oplus \mathcal{E}(m_k) = \mathcal{E}(m_1 \oplus \ldots \oplus m_k)$ for any k. We will say that such encryption schemes are ∞-pseudohomomorphic (or simply homomorphic). Note that if $\mathcal{E}$ is surjective for any element $y \in \mathcal{D}^{-1}(x)$ there is an $r \in \mathcal{R}$ such that $y = \mathcal{E}(x, r)$. Thus if $\mathcal{E}$ is surjective and 2-pseudohomomorphic then $\mathcal{E}(m_1) \oplus \mathcal{E}(m_2) = \mathcal{E}(m_1 \oplus m_2)$ and thus $\mathcal{E}$ is (fully) homomorphic.

With respect to security, it is well-known that homomorphic schemes can never achieve security against chosen-ciphertext attacks. For this reason, we will consider *indistinguishability* under chosen-plaintext attacks (CPA). Roughly speaking, an encryption scheme is IND-CPA secure if, knowing only the public key, it is computationally hard to distinguish between two encryptions of two different plaintexts. A more formal definition of this security property can be found in Appendix A.

3.2 Extending Pseudohomomorphic Encryption Schemes

Following the idea of Definition 1 one can define a pseudohomomorphic relation between two groups by:

Definition 2. *Let $(G_1, \oplus_1)$ and $(G_2, \oplus_2)$ be two groups and*

$$\phi : (G_1, \oplus_1) \to (G_2, \oplus_2) \quad \phi^* : (G_2, \oplus_2) \to (G_1, \oplus_1)$$

two computable functions such that for all $k \leq L$ and all k-tuple $(g_1, \ldots, g_k) \in G_1^k$ we have $\phi^*(\phi(g_1) \oplus_2 \ldots \oplus_2 \phi(g_k)) = g_1 \oplus_1 \ldots \oplus_1 g_k$.
We say that (ϕ, ϕ^*) forms a computable L-pseudohomomorphism from $(G_1, \oplus_1)$ to $(G_2, \oplus_2)$.

Pseudohomomorphisms can be combined with pseudohomomorphic encryption schemes in order to change their plaintext or ciphertext space without changing the security properties of the cryptosystem. This is stated in the next proposition.

Proposition 1. *If $PKE = (\mathcal{KG}, \mathcal{E}, \mathcal{D})$ is an L-pseudohomomorphic encryption scheme such that there is a computable L'-pseudohomomorphism (ϕ, ϕ^*) (ϕ being public) from a space $(G_1, \oplus_1)$ to PKE's plaintext space $(\mathcal{M}, \oplus_{\mathcal{M}})$, then the associated encryption scheme $PKE' = (\mathcal{KG}, \mathcal{E} \circ \phi, \phi^* \circ \mathcal{D})$ is a $min(L, L')$-pseudohomomorphic encryption scheme. If there exists a computable L''-pseudohomomorphism (ψ, ψ^*) (ψ being public) from PKE's ciphertext space to another space, $PKE' = (\mathcal{KG}, \psi \circ \mathcal{E}, \mathcal{D} \circ \psi^*)$ is a $min(L, L'')$-pseudohomomorphic encryption scheme. Moreover, in both cases, if PKE is IND-CPA secure, PKE' is IND-CPA secure too.*

Proof. (sketch) Suppose that PKE' is a plaintext space extension. For any $k \leq min(L, L')$ and any $g_1, \ldots, g_k \in G_1$,

$$\begin{aligned}
\mathcal{D}'(\mathcal{E}'(g_1) \oplus_{\mathcal{C}} \cdots \oplus_{\mathcal{C}} \mathcal{E}'(g_k)) &= \phi^*(\mathcal{D}(\ \ \mathcal{E}(\phi(g_1)) \oplus_{\mathcal{C}} \cdots \oplus_{\mathcal{C}} \mathcal{E}(\phi(g_k))\ )) && \text{by definition of } \mathcal{E}' \text{ and } \mathcal{D}', \\
&= \phi^*(\phi(g_1) \oplus_{\mathcal{M}} \cdots \oplus_{\mathcal{M}} \phi(g_k)) && \text{as } k \leq L, \\
&= g_1 \oplus_1 \cdots \oplus_1 g_k && \text{as } k \leq L'.
\end{aligned}$$

PKE' is thus $min(L, L')$-pseudohomomorphic. The same proof can be done if PKE' is a ciphertext extension. IND-CPA for PKE' in the first case is ensured as distinguishing two plaintexts x_1, x_2 in PKE' implies distinguishing $\phi(x_1), \phi(x_2)$ in PKE and ϕ must be injective. In the second case, being ϕ public, distinguishing x_1, x_2 in PKE' implies distinguishing them also in PKE. □

3.3 Twisting Additive Pseudohomomorphic Encryption Schemes

If for a given L-pseudohomomorphic encryption scheme, its plaintext space is $(\mathbb{Z}_s, +)^n$ for some positive integers $s, n \in \mathbb{Z}^+$, we say that the scheme is *plaintext additive*. If its ciphertext space is $(\mathbb{Z}_{s'}, +)^{n'}$ for some positive integers $s', n' \in \mathbb{Z}^+$, we say that the scheme is *ciphertext additive*. When a plaintext or ciphertext space is $(\mathbb{Z}_s, +)^n$ we will call s its *order* and n its *dimension*. These schemes can be easily modified using Proposition 1 and simple L-pseudohomomorphisms.

Lemma 1. *Let $(\mathbb{Z}_s, +)^n$ be a group for $s, n \in \mathbb{Z}^+$. For any $k, L, s' \in \mathbb{Z}^+$ such that $(2^k - 1) \cdot\ L <\ s' < s$ there is a computable L-pseudohomomorphism from $(\mathbb{Z}_s, +)^n$ to $(\mathbb{Z}_{s'}, +)^{n'}$ where $n' = n \cdot \lceil (\log_2 s)/k \rceil$.*

The proof (available in Appendix B) uses a simple construction that we will associate to this lemma. The main idea is to split each coordinate of an element in $(\mathbb{Z}_s, +)^n$ in k-bit blocks, and to form a vector of dimension n', each k-bit block being the binary representation of a coordinate. Computing with such a vector in $(\mathbb{Z}_{s'}, +)^{n'}$ for a large enough s' ensures that the process is reversible.

Corollary 1. *Let PKE be a ciphertext additive L-pseudohomomorphic encryption scheme with ciphertext space $(\mathbb{Z}_s, +)^n$. For any $k \in \mathbb{Z}^+$, it is possible to lower the order of the ciphertext space s to any value s' such that $(2^k - 1) \cdot L < s' < s$ by increasing the dimension n to $n' = n \cdot \lceil (\log_2 s)/k \rceil$. This transformation preserves indistinguishability.*

It is thus possible to split ciphertexts in order to have many small elements instead of one large element while preserving the pseudohomomorphic and indistinguishability properties. The extreme case happens when one considers $k = 1$, i.e. the initial ciphertexts of PKE, which are elements in $(\mathbb{Z}_s, +)^n$, are transformed into elements of $(\mathbb{Z}_2, +)^{n \cdot \lceil \log_2 s \rceil}$ (vectors of bits). To preserve the L-pseudohomomorphic properties, however, we will see these bits as elements of $\mathbb{Z}_{s'}$, for some s' between L and s, as stated in the previous lemma and corollary.

For an encryption function $\mathcal{E}$ with ciphertext space $(\mathbb{Z}_s, +)^n$, let us denote as $\parallel \mathcal{E} \parallel_\infty$ the maximum value that can be in a component of $\mathcal{E}(x)$, for all possible plaintexts x. Of course, we have $\parallel \mathcal{E} \parallel_\infty \leq s$, but in general this value can be much smaller. For example, if we apply the construction of Lemma 1 to an encryption function $\mathcal{E}$, then we obtain $\parallel \mathcal{E} \parallel_\infty \leq 2^k - 1$, and in the particular case where $k = 1$, we will have $\parallel \mathcal{E} \parallel_\infty \leq 1$.

The following lemmas give tools to increase the ciphertext space order of a ciphertext additive scheme, and to modify the plaintext space dimension of a plaintext additive scheme.

Lemma 2. *For any $s_1, s_2, n \in \mathbb{Z}^+$ there is a computable L-pseudohomomorphism from $(\mathbb{Z}_{s_1}, +)^n$ to $(\mathbb{Z}_{s_2}, +)^n$ for $L = \lfloor s_2/s_1 \rfloor$.*

Lemma 3. *A plaintext additive L-pseudohomomorphic scheme PKE with plaintext space $(\mathbb{Z}_s, +)^n$ can be transformed into a plaintext additive L-pseudohomomorphic scheme PKE' with plaintext space $(\mathbb{Z}_s, +)^{kn}$, for any $k \in \mathbb{Z}^+$. This transformation preserves indistinguishability.*

Lemma 4. *For any $s, n \in \mathbb{Z}^+$ and any $\ell < n$ we define $\pi^{-1} : (\mathbb{Z}_s, +)^\ell \to (\mathbb{Z}_s, +)^n$ by $\pi^{-1}((x_1, \cdots, x_\ell)) = (x_1, \cdots, x_\ell, 0, \cdots, 0)$, where π is the standard projection. (π^{-1}, π) is a computable ∞-pseudohomomorphism from $(\mathbb{Z}_s, +)^\ell$ to $(\mathbb{Z}_s, +)^n$.*

Corollary 2. *A plaintext additive L-pseudohomomorphic scheme PKE with plaintext space $(\mathbb{Z}_s, +)^n$ can be transformed into a plaintext additive L-pseudohomomorphic scheme PKE' with plaintext space $(\mathbb{Z}_s, +)^\ell$, for any $\ell \in \mathbb{Z}^+$. This transformation preserves indistinguishability.*

Summing up, it is possible to increase or reduce the ciphertext order of a ciphertext additive scheme, possibly changing the ciphertext dimension, and to increase or reduce the plaintext dimension of a plaintext additive scheme, without changing its order. Moreover, *each of these operations preserves indistinguishability.* Combining these results, we will be able to modify the schemes in a given chain in such a way that the order and dimension of the ciphertext space of a scheme are equal to the order and dimension of the plaintext space of the following scheme in the chain.

3.4 t-Chained Schemes

For simplicity, we start with the case of chains with $t = 2$ schemes. We propose a way to adapt a plaintext additive pseudohomomorphic encryption scheme PKE_2 in order to encrypt the ciphertexts of a plaintext and ciphertext additive pseudohomomorphic encryption scheme PKE_1, in such a way that the imbrication of both schemes leads to a new plaintext additive pseudohomomorphic encryption scheme called 2-chained.

Definition 3. *Let $PKE_1 = (\mathcal{KG}_1, \mathcal{E}_1, \mathcal{D}_1)$ be a plaintext and ciphertext additive L_1-pseudohomomorphic encryption scheme with associated plaintext and ciphertext spaces $(\mathbb{Z}_{s_1}, +)$ and $(\mathbb{Z}_{s_2}, +)^{n_2}$, and let $PKE_2 = (\mathcal{KG}_2, \mathcal{E}_2, \mathcal{D}_2)$ be a plaintext additive L_2-pseudohomomorphic encryption scheme with associated plaintext and ciphertext spaces $(\mathbb{Z}_{s_2}, +)$ and $(\mathcal{C}_2, \oplus_2)$.*

Set $PKE'_2 = (\mathcal{KG}'_2, \mathcal{E}'_2, \mathcal{D}'_2)$ as the plaintext additive L_2-pseudohomomorphic encryption scheme with plaintext space $(\mathbb{Z}_{s_2}, +)^{n_2}$ and ciphertext space $(\mathcal{C}_2, \oplus_2)^{n_2}$ derived from PKE_2 (using Lemma 3), such that $\mathcal{E}'_2((x_1, \ldots, x_{n_2})) = (\mathcal{E}_2(x_1), \ldots, \mathcal{E}_2(x_{n_2}))$.

We define the 2-chained encryption scheme derived from PKE_1 and PKE_2 as $PKE = (\mathcal{KG}, \mathcal{E}, \mathcal{D})$ with: $\mathcal{KG} = \mathcal{KG}_1 \times \mathcal{KG}'_2$; $\mathcal{E} = \mathcal{E}'_2 \circ \mathcal{E}_1$; $\mathcal{D} = \mathcal{D}_1 \circ \mathcal{D}'_2$.

The resulting 2-chained encryption scheme PKE has plaintext space $(\mathbb{Z}_{s_1}, +)$ and ciphertext space $(\mathcal{C}_2, \oplus_2)^{n_2}$. The following proposition describes the security and pseudohomomorphic properties of such a scheme.

Proposition 2. *A 2-chained encryption scheme PKE is a plaintext additive L-pseudohomomorphic encryption scheme with $L = min(L_1, L_2)$. If one of the encryption schemes used to create the 2-chained scheme is IND-CPA secure, then PKE is also IND-CPA secure.*

Proof. (sketch) Lemma 3 proves that if PKE_2 is IND-CPA secure then PKE_2' is IND-CPA secure too. The 2-chained scheme can be seen as an extension of PKE_2' with the L_1-pseudohomomorphism $(\mathcal{E}_1, \mathcal{D}_1)$ or as an extension of PKE_1' with the L_2-pseudohomomorphism $(\mathcal{E}_2', \mathcal{D}_2')$. In any case, Proposition 1 proves that the resulting 2-chained scheme is a plaintext additive $min(L_1, L_2)$-pseudohomomorphic encryption scheme, and IND-CPA secure if any of PKE_1 or PKE_2 are IND-CPA secure.

Note that in Definition 3 we are implicitly assuming that the order s_2 of the ciphertext space of PKE_1 is the same as the order of the plaintext space of PKE_2. The lemmas of Section 3.3 show that in order to obtain 2-chained schemes, the only thing we need is to be able to create plaintext and ciphertext additive pseudohomomorphic encryption schemes with large enough plaintext order. This is specified by the following proposition.

Proposition 3. *For any L, if there is a family of plaintext and ciphertext additive L-pseudohomomorphic encryption schemes such that the plaintext space order can be chosen arbitrarily large, it is possible to construct a 2-chained L-pseudohomomorphic encryption scheme.*

Proof. (sketch) Suppose that, for any positive integer s, we can take a ciphertext and plaintext additive L-pseudohomomorphic encryption scheme PKE such that $\mathcal{E} : (\mathbb{Z}_s, +)^n \rightarrow (\mathbb{Z}_{s'}, +)^{n'}$, where n, s', n' may depend on s.

Set PKE_1 as such a scheme for a given s_1. We denote as $(\mathbb{Z}_{s_1}, +)^{n_1}$ and $(\mathbb{Z}_{s_1'}, +)^{n_1'}$ the plaintext and ciphertext spaces of this cryptosystem. Set PKE_2 as a second scheme with plaintext and ciphertext spaces $(\mathbb{Z}_{s_2}, +)^{n_2}$ and $(\mathbb{Z}_{s_2'}, +)^{n_2'}$, such that the plaintext order s_2 satisfies $\lfloor s_2/s_1 \rfloor > L$.

Using Lemma 4 we lower the plaintext dimension of these schemes to one and obtain two L-pseudohomomorphic schemes PKE_1' and PKE_2'. Using Lemma 2 we increase the ciphertext space order of PKE_1' to s_2 and obtain an L-pseudohomomorphic scheme PKE_1'' (as $\lfloor s_2/s_1 \rfloor > L$).

PKE_1'' (resp. PKE_2') is L-pseudohomomorphic and has plaintext and ciphertext spaces $(\mathbb{Z}_{s_1}, +)$ and $(\mathbb{Z}_{s_2}, +)^{n_1'}$ (resp. $(\mathbb{Z}_{s_2}, +)$ and $(\mathbb{Z}_{s_2'}, +)^{n_2'}$). These schemes satisfy the properties required in Definition 3 and can therefore be used to construct a 2-chained L-pseudohomomorphic encryption scheme.

Generalization. If PKE_2 is plaintext and ciphertext additive, the 2-chained scheme constructed above is implicitly plaintext and ciphertext additive. Note that in this case, we can use it for further imbrications. We thus define a *t-chained scheme*, as the consecutive imbrication of $t-1$ plaintext and ciphertext pseudohomomorphic schemes $PKE_1, PKE_2', \ldots, PKE_{t-1}'$ and of one plaintext pseudohomomorphic scheme PKE_t' (all of them resulting from properly twisting some initial schemes, as explained in Section 3.3). The encryption diagram of a t-chained scheme would be

$$(\mathbb{Z}_{s_1}, +) \xrightarrow{\mathcal{E}_1} (\mathbb{Z}_{s_2}, +)^{n_2} \xrightarrow{\mathcal{E}_2'} (\mathbb{Z}_{s_3}, +)^{n_3} \xrightarrow{\mathcal{E}_3'} \ldots \xrightarrow{\mathcal{E}_{t-1}'} (\mathbb{Z}_{s_t}, +)^{n_t} \xrightarrow{\mathcal{E}_t'} (\mathcal{C}_t, \oplus_t)^{n_t}.$$

Propositions 2 and 3 are trivially generalized. This scheme is thus L-pseudohomomorphic, with $L = min(L_1, \ldots, L_t)$ if PKE_i is L_i-pseudohomomorphic. And it is IND-CPA secure if some PKE_i is IND-CPA secure.

In [23], Kawachi, Tanaka and Xagawa propose a set of lattice-based ciphertext and plaintext additive pseudohomomorphic encryption schemes, derived from [24–27], that are proved to be IND-CPA secure under standard assumptions. In these schemes, L and the plaintext space order can be set to any value in $\mathbb{Z}^+$ and thus they can be used to implement t-chained encryptions schemes.

4 Secure Function Evaluation with t-Chained Schemes

Maybe the most important consequence of the existence of t-chained encryption schemes is that they can be used for computing over ciphertexts. Namely a t-chained encryption scheme $PKE = (\mathcal{KG}, \mathcal{E}, \mathcal{D})$ resulting from the schemes $PKE_1, \ldots, PKE_t$ can be used by anyone to:

(i) Compute ciphertexts $\mathcal{E}_1(a), \dots, \mathcal{E}_t(a), \mathcal{E}(a)$ of any of the encryption schemes $PKE_1, \dots, PKE_t, PKE$.
(ii) Given L ciphertexts $\mathcal{E}(a_1), \dots, \mathcal{E}(a_L)$, anyone can publicly compute an element C, such that $\mathcal{D}(C) = a_1 + \cdots + a_L$.
(iii) Given a set of t ciphertexts $\mathcal{E}_1(a_1), \dots, \mathcal{E}_t(a_t)$, anyone can publicly compute an element C, such that $\mathcal{D}(C) = a_1 \cdot \ldots \cdot a_t$.

In particular, a t-chained encryption scheme can thus be used for secure evaluation of multivariate polynomials (of total degree t). In order to show how this is done, and for simplicity of the explanation, let us consider first the case of a 2-chained scheme. Then we will informally present the general case.

4.1 Secure Evaluation of the Sum and Product of Two Inputs

Let PKE_1, PKE_2, PKE_2', and PKE denote the cryptosystems introduced in Definition 3.

Sum of inputs. Proposition 2 states that PKE is L-pseudohomomorphic with $L = min(L_1, L_2)$. Indeed, if $a_1, a_2 \in \mathbb{Z}_{s_1}$, we can consider the ciphertexts $C_i = \mathcal{E}(a_i) = \mathcal{E}_2'(\mathcal{E}_1(a_i)) \in (\mathcal{C}_2, \oplus_2)^{n_2}$, for $i = 1, 2$. Then, if $L \geq 2$, anyone can operate these ciphertexts to obtain $C = C_1 \oplus_2 C_2 = \tilde{\mathcal{E}}(a_1 + a_2)$. Recall that we use notation $\tilde{\mathcal{E}}(x)$ to represent an element of $\mathcal{D}^{-1}(x)$. The owner of $sk = (sk_1, sk_2)$ can decrypt C, by applying $\mathcal{D} = \mathcal{D}_1 \circ \mathcal{D}_2$, to obtain $a_1 + a_2 \bmod s_1$ as desired.

Product of inputs. Regarding secure evaluation of the product, given two values $a_1 \in \mathbb{Z}_{s_1}$ and $a_2 \in \mathbb{Z}_{s_2}$, we can consider the ciphertexts $c_1 = \mathcal{E}_1(a_1) \in (\mathbb{Z}_{s_2}, +)^{n_2}$ and $c_2 = \mathcal{E}_2(a_2) \in (\mathcal{C}_2, \oplus_2)$. We write $c_1 = (\mathcal{E}_1^{(1)}(a_1), \dots, \mathcal{E}_1^{(n_2)}(a_1))$, where $\mathcal{E}_1^{(l)}(a_1) \in \mathbb{Z}_{s_2}$, for $l = 1, \dots, n_2$. Obviously, $\mathcal{E}_1^{(l)} : (\mathbb{Z}_{s_1}, +)^{n_1} \to (\mathbb{Z}_{s_2}, +)$ is a L_1-pseudohomomorphism. We compute:

$$\begin{aligned}(\mathcal{E}_1^{(1)}(a_1)\mathcal{E}_2(a_2), \dots, \mathcal{E}_1^{(n_2)}(a_1)\mathcal{E}_2(a_2)) &= (\tilde{\mathcal{E}}_2(\mathcal{E}_1^{(1)}(a_1)a_2)), \dots, \tilde{\mathcal{E}}_2(\mathcal{E}_1^{(n_2)}(a_1)a_2)) \\ &= (\tilde{\mathcal{E}}_2(\tilde{\mathcal{E}}_1{}^{(1)}(a_1a_2)), \dots, \tilde{\mathcal{E}}_2(\tilde{\mathcal{E}}_1^{(n_2)}(a_1a_2))) \\ &= \tilde{\mathcal{E}}_2'(\tilde{\mathcal{E}}_1(a_1a_2)) = \tilde{\mathcal{E}}(a_1a_2),\end{aligned}$$

where the first operation has to be interpreted as 'applying $\oplus_2$ to $\mathcal{E}_2(a_2)$ a number $\mathcal{E}_1^{(l)}(a_1)$ of times'. The first equality is true if $L_2 \geq \parallel \mathcal{E}_1 \parallel_\infty$, where $\parallel \ \parallel_\infty$ is the norm defined in Section 3.3. It is important to be precise as this value can be much smaller than the space order s_2. Indeed, in our practical examples we will use Lemma 1 with $k = 1$ and thus we will have $\parallel \mathcal{E}_1 \parallel_\infty = 1$, whereas $s_2 >> 1$. The second equality in the above array is true if $L_1 \geq a_2$, and the two last equalities hold because of the definitions of PKE_2' and PKE.

4.2 General Case

Although we have explained the case $t = 2$ for simplicity, the computing techniques can be easily generalized for greater values of t. We skip the details due to the cumbersome notation. As an informal example with $t = 3$, if we are interested in the product $a_1a_2a_3$, we have to provide a ciphertext $\mathcal{E}_3(a_3)$. Once the first multiplication has been done, we have $\tilde{\mathcal{E}}_2'(\tilde{\mathcal{E}}_1(a_1a_2))$, which belongs to $(\mathbb{Z}_{s_3}, +)^{n_3}$ (using the notations of the generalization at the end of Section 3.4). Therefore, this ciphertext can be represented as $(\tilde{\mathcal{E}}_2'^{(1)}(\tilde{\mathcal{E}}_1(a_1a_2)), \dots, \tilde{\mathcal{E}}_2'^{(n_3)}(\tilde{\mathcal{E}}_1(a_1a_2)))$. The same procedure as before (i.e. 'multiplying' each component of this vector with $\mathcal{E}_3(a_3)$) can be applied to obtain:

$$\begin{aligned}\big(\tilde{\mathcal{E}}_2'^{(1)}(\tilde{\mathcal{E}}_1(a_1a_2))\mathcal{E}_3(a_3), \, \dots \, , \tilde{\mathcal{E}}_2'^{(n_3)}(\tilde{\mathcal{E}}_1(a_1a_2))\mathcal{E}_3(a_3)\big) = \\ &= \big(\tilde{\mathcal{E}}_3(\tilde{\mathcal{E}}_2'^{(1)}(\tilde{\mathcal{E}}_1(a_1a_2))a_3), \dots, \tilde{\mathcal{E}}_3(\tilde{\mathcal{E}}_2'^{(n_3)}(\tilde{\mathcal{E}}_1(a_1a_2))a_3)\big) \\ &= \big(\tilde{\mathcal{E}}_3(\tilde{\mathcal{E}}_2'^{(1)}(\tilde{\mathcal{E}}_1(a_1a_2)a_3)), \dots, \tilde{\mathcal{E}}_3(\tilde{\mathcal{E}}_2'^{(n_3)}(\tilde{\mathcal{E}}_1(a_1a_2)a_3))\big) \\ &= \big(\tilde{\mathcal{E}}_3(\tilde{\mathcal{E}}_2'^{(1)}(\tilde{\mathcal{E}}_1(a_1a_2a_3))), \dots, \tilde{\mathcal{E}}_3(\tilde{\mathcal{E}}_2'^{(n_3)}(\tilde{\mathcal{E}}_1(a_1a_2a_3)))\big) \\ &= \tilde{\mathcal{E}}_3'(\tilde{\mathcal{E}}_2'(\tilde{\mathcal{E}}_1(a_1a_2a_3))) = \tilde{\mathcal{E}}(a_1a_2a_3).\end{aligned}$$

Note that the number of operations involving ciphertexts of the different encryption functions in the chain increases with t. In this case, we must have $L_1 > a_2 a_3$, $L_2 >\| \mathcal{E}_1 \|_\infty a_3$, and $L_3 >\| \mathcal{E}_2 \|_\infty$. In the general case of t-chained schemes, in order to compute the ciphertext $\tilde{\mathcal{E}}(a_1 \cdots a_t)$, we must have $L_1 > \prod_{j=2}^{t} a_j$ and $L_i >\| \mathcal{E}_{i-1} \|_\infty \prod_{j=i+1}^{t} a_j$, for $i = 2, \ldots, t$.

Summing up, the presented t-chained encryption schemes can be used to evaluate t-degree multivariate polynomials on ciphertexts. Namely, if $F(x_1, \ldots, x_n)$ is a n-variate polynomial over $\mathbb{Z}_s$ of total degree t, and with m monomials, one can consider encryptions $C_{i,j} = \mathcal{E}_i(a_j)$, for $i = 1, \ldots, t$ and $j = 1, \ldots, n$, and anyone can compute an encryption $\tilde{C}$ of the value $F(a_1, \ldots, a_n) \bmod s_1$. We represent such a polynomial by $F(a_1, \ldots, a_n) = \sum_{r=1}^{m} \lambda_r M_r$, M_r being a monomial of degree at most t. In order to evaluate the polynomial:

1. Each monomial M_r is computed by multiplying the associated ciphertexts;
   If the monomial has degree $t' < t$ it is then multiplied by $t - t'$ encryption of 1;
2. The monomial M_r is added to itself λ_r times;
3. The results of each monomial computation are summed up.

The constraints associated to the values $L_1, \ldots, L_t$ are easy to derive from the ones we presented for sum and product computation. Namely, we must have $L_1 > m \cdot \lambda \cdot a^{t-1}$ and $L_i > m \cdot \lambda \cdot a^{t-i} \| \mathcal{E}_{i-1} \|_\infty$, for $i = 2, \ldots, t$, noting λ and a the maximum possible values for $\lambda_1, \ldots, \lambda_m$ and $a_1, \ldots, a_n$ respectively.

Note that for boolean polynomials we have $a = \lambda = 1$ and thus if we manage to have $\| \mathcal{E}_{i-1} \|_\infty = 1$ the only constraint is $L_i > m$ for $i \in \{1, \ldots, t\}$.

Some applications of SFE in which having $t > 2$ is relevant are presented in Appendix C

4.3 Security

As stated by Proposition 2, a t-chained scheme PKE (with underlying schemes $PKE_1, \ldots, PKE_t$), is IND-CPA secure as long as one of the underlying schemes is IND-CPA secure. This means that the ciphertexts of any plaintext pair m_1, m_2 resulting from the encryption with $\mathcal{E} = \mathcal{E}'_t \circ \cdots \circ \mathcal{E}_1$ are indistinguishable.

When requiring a secure evaluation of a polynomial, a user (Bob) holding a secret input a_j does not know which operations are going to be done with this input. He should thus encrypt this value with all the underlying encryption schemes of the chain, producing $C_{i,j} = \mathcal{E}_i(a_j)$ for $i = 1, \ldots, t$, so that the value may be used at any point of a product or sum evaluation. Therefore, when considering to use a t-chained scheme for secure function evaluation we are not interested in the IND-CPA security of the t-chained scheme PKE, but on the IND-CPA security of the scheme PKE_G defined by $\mathcal{E}_G(x) = (\mathcal{E}_1(x), \ldots, \mathcal{E}_t(x))$.

Proposition 4. *If t encryption schemes $\mathcal{E}_1, \ldots, \mathcal{E}_t$ are all ε-indistinguishable under CPA attacks, then the global encryption scheme $\mathcal{E}_G(x) = (\mathcal{E}_1(x), \ldots, \mathcal{E}_t(x))$ is $t\varepsilon$-indistinguishable under CPA attacks.*

The proof of this Proposition can be found in Appendix B.2. The final conclusion is that our protocol for Secure Function Evaluation offers security for Bob, despite the publication of all the ciphertexts $C_{i,j} = \mathcal{E}_i(a_j)$, for $i = 1, \ldots, t$ and $j = 1, \ldots, n$, as long as all the encryption schemes $PKE_1, \ldots, PKE_t$ are IND-CPA secure. Some details on the privacy of the function (*i.e.* symmetric security) are given in Appendix D.

5 Specific Realizations

The encryption scheme of Boneh, Goh and Nissim allows to compute products of two operands. In this section we will provide two schemes that allow to compute products of three operands.

5.1 2-Chained Encryption Schemes for 3-DNF Secure Evaluation

As noted at the end of Section 3.4, the lattice-based additively pseudohomomorphic schemes that have been proposed by Kawachi, Tanaka and Xagawa in [23], derived from [24–27], can be used to construct a t-chained scheme for any value of t, in particular for $t = 3$. Our idea here is to achieve the same functionality as in a 3-chained scheme (e.g. private evaluation of multi-variate polynomials with degree up to 3) by constructing a 2-chained scheme which uses as PKE_2 the scheme of Boneh-Goh-Nissim. It is possible to set $t > 2$ by chaining multiple times lattice-based schemes and once at the end Boneh-Goh-Nissim. However, ciphertext size grows quickly, and therefore we present here just the case $t = 2$. This scheme allows to securely evaluate, in the sense of SFE, 3-DNF boolean formulas with billions of disjunctions (and thus with billions of depth levels) which could not be done in practice with any of the previously existing SFE protocols. For a definition of DNF formulas, and of how they are related to boolean polynomials, see Appendix C.

Regarding PKE_1, we consider one of the schemes proposed in [23]. It is a variant of [25] (noted hereafter mR04), whose IND-CPA security is based on a worse-case reduction to $\tilde{O}(n^{1.5+r}) - uSVP$ for a given parameter r. In mR04, the encryption and decryption functions form an L-pseudohomomorphism $(\mathcal{E}, \mathcal{D})$ from $(\mathbb{Z}_p, +)$ to $(\mathbb{Z}_N, +)$ with $N = 2^{8n^2}$, being n a security parameter such that $L \times p < n^r$. The encryption system has a decryption error probability which for practical parameters is negligible, and thus will not be considered in our simple practical constructions.

Choosing $p > L$, we can apply Lemma 1 to this scheme, with $k = 1$, to obtain an L-pseudohomomorphic encryption scheme $PKE_{1\alpha}$ with plaintext space $(\mathbb{Z}_p, +)$ and ciphertext space $(\mathbb{Z}_p, +)^{8n^2}$, such that $\| \mathcal{E}_{1\alpha} \|_\infty = 1$. We take a hard-to-factor modulus $N_P > p \times L$ and we consider the pseudohomomorphism from $(\mathbb{Z}_p, +)$ to $(\mathbb{Z}_{N_P}, +)$ described in Lemma 2. In this way, we obtain PKE_1, an L-pseudohomomorphic encryption scheme with plaintext space $(\mathbb{Z}_p, +)$ and ciphertext space $(\mathbb{Z}_{N_P}, +)^{8n^2}$ with $\| \mathcal{E}_1 \|_\infty = 1$.

Let PKE_2 be an instance of Boneh-Goh-Nissim's encryption scheme for the modulus N_P. PKE_2 is an ∞-pseudohomomorphic encryption scheme with plaintext space $(\mathbb{Z}_{N_P}, +)$ and ciphertext space $(\mathbb{G}, \times)$. Applying the construction given in Definition 3, we obtain a 2-chained L-pseudohomomorphic encryption scheme PKE with plaintext space $(\mathbb{Z}_p, +)$ and ciphertext space $(\mathbb{G}, \times)^{8n^2}$. The following results are directly inferred from Proposition 2 and Proposition 4.

Theorem 1. *The 2-chained encryption scheme PKE is IND-CPA secure assuming either that the $\tilde{O}(n^{1.5+r})-$ $uSVP$ problem is hard or that the subgroup decision problem is hard in $\mathbb{G}$. Moreover, it is possible to use it for Secure Function Evaluation of 3-DNF boolean functions with up to L disjunctions if both problems are assumed to be hard.*

The resulting 2-chained encryption scheme allows to start from ciphertexts $c_1 = \mathcal{E}_1(a_1)$, $c_2 = \mathcal{E}_2(a_2)$, for integer values a_1, a_2, and to obtain a ciphertext $C_{12} = \mathcal{E}_2(\mathcal{E}_1(a_1 a_2)) \in \mathbb{G}$. Now, for a third integer a_3, we can consider the ciphertext $c_3 = \mathcal{E}_2(a_3) \in \mathbb{G}$. Using the bilinear map of the scheme by Boneh-Goh-Nissim, we can compute a ciphertext $C = e(C_{12}, c_3)$. This ciphertext C is an encryption of $a_1 a_2 a_3$, according to a different cryptosystem related to PKE_2, which can be decrypted by the owner of the secret key of the scheme PKE_2.

Note that Boneh-Goh-Nissim ciphertexts can only be decrypted if the associated plaintexts are small. We have $\| \mathcal{E}_1 \|_\infty = 1$ and thus plaintexts will be small if the integers a_i are small too. This makes this t-chained encryption scheme specially adapted for boolean formula evaluation. Using the technique proposed by Boneh, Goh and Nissim for DNF formula evaluation, we will set $p = L + 1$. Thus, as ciphertext size is in $O(n^2)$, for an optimal choice of n (such that $L \times p \simeq n^r$) we have a ciphertext size in $O(L^{4/r})$. The asymptotic ciphertext growth in the formula depth is polynomial, whereas with the technique of Sander, Young and Yung [22] it is exponential.

The underlying security problem of $mR04$, $\tilde{O}(n^{1.5+r}) - uSVP$, has been solved for $n \simeq 10$ and $r \simeq 10$ but is believed to be unfeasible for $r = 10$ and $n \simeq 100$ [28]. Thus, a concrete instantiation of $mR04$ could be: $n = 100$, $r = 10$, $p = 10^9 + 1$ and $L = 10^9$, which verify $p \times L < n^r$ and $L < p$. We can set a 1024-bit modulus for Boneh-Goh-Nissim which corresponds to current standards on factorization and ensures $N_P > L \times p$. With these parameters, our 2-chained encryption scheme would be 10^9-pseudohomomorphic, with plaintext space $(\mathbb{Z}_{10^9+1}, +)$ and ciphertext space $(\mathbb{G}, \times)^{80000}$, an element of $\mathbb{G}$ being represented by 2048 bits. Ciphertexts will therefore be a few megabytes long and the scheme will be usable to evaluate 3-DNF boolean formulas with depth up to 10^9. Again, the protocol of Sander, Young and Yung[22] also allows to evaluate such formulas, but ciphertext size will be in $O(e^{10^9})$ for such depths.

5.2 t-Chained Encryption Schemes for General Use

In [29], Aguilar, Castagnos and Gaborit propose a lattice-based homomorphic encryption scheme which is ciphertext and plaintext additive. The IND-CPA security of this scheme is based on a non-standard security assumption, the hardness of a problem they named the Differential Knapsack Vector Problem, which as its name states is a variation of the Knapsack Problem against which traditional lattice-reduction attacks seem ineffective. In this scheme, noted hereafter $ACG08$, the encryption and decryption functions form an L-pseudohomomorphism $(\mathcal{E}, \mathcal{D})$ from $(\mathbb{Z}_p, +)^N$ to $(\mathbb{Z}_q, +)^{2N}$ with $q = K \times p^2 \times L$, being K and N two security parameters.

For any $i \in \{1, \ldots, t\}$ we set $PKE_{i\alpha}$ as an instance of $ACG08$ from $(\mathbb{Z}_{p_i}, +)^N$ to $(\mathbb{Z}_{q_i}, +)^{2N}$ such that $p_1 = p$ and $p_i = q_{i-1}$ for $i > 1$. Using Lemma 4 on each of these schemes we obtain PKE_i with plaintext space $(\mathbb{Z}_{p_i}, +)$ and ciphertext space $(\mathbb{Z}_{q_i}, +)^{2N}$. Applying the construction given in Definition 3 iteratively, we obtain a t-chained L-pseudohomomorphic encryption scheme PKE with plaintext space $(\mathbb{Z}_p, +)$ and ciphertext space $(\mathbb{Z}_{q_t}, +)^{N_t}$ with $q_t = (K \times L \times p)^{2t-1} \times p$ and $N_t = (2N)^t$. Again, the following theorem is a direct result of Propositions 2 and 4.

Theorem 2. *The t-chained encryption scheme PKE is IND-CPA secure assuming that the Differential Knapsack Vector Problem is hard. Moreover, with the same assumption, it is possible to use it for Secure Function Evaluation of t-degree multivariate polynomials over $\mathbb{Z}_p$ with up to L monomials.*

Ciphertext size is $((2t-1)\log(K \times L \times p) + \log p) \times (2N)^t$, which is still exponential in t but just logarithmic in L. With this t-chained scheme there is no issue on decryption and thus it can be used for SFE of t-degree polynomials over an arbitrary ring $\mathbb{Z}_p$ or on any of the other applications of homomorphic schemes.

Based on the security parameters proposed in [29], we can set $N = 50$ and $K = 2^{20}$. For $t = 3$ we have a ciphertext size of roughly ten megabytes, for values of p and L close to one. Since the growth of the ciphertext size is logarithmic in these parameters, it remains pretty much the same for larger values of p, L. For example, for $p = 2^{32}, L = 2^{32}$ ciphertext size is just multiplied by a factor 4 (and not 64 as we already have $K = 2^{20}$) and for $p = 2^{100}, L = 2^{100}$ it is multiplied by a factor 10.

6 Conclusion

The construction we have presented in this paper provides a general way to obtain additively homomorphic schemes that allow to compute products of t operands. We have presented two specific implementations, but many other alternatives can be considered: for example a 3-chained encryption scheme could have been obtained from chaining twice $mR04$ instances with Paillier's encryption scheme which would have given an alternative 3-chained scheme for general use based on uSVP and the subgroup decision problem.

The performance differences between the proposed schemes illustrates the fact that the efficiency of our generic construction depends much on the underlying encryption schemes. In fact, there is no reason for the growth to be exponential in t, even if the underlying encryption schemes are randomized (and thus have an expansion factor in $]1; \infty[$). Indeed, in [30] Lipmaa uses Damgård-Jurik [31], an additively homomorphic encryption scheme, to iteratively encrypt data with a function $\mathcal{E}_1 \circ \cdots \circ \mathcal{E}_t$ such that $\mathcal{E}_i$ has an expansion factor $(i+1)/i > 1$. The global scheme has therefore an expansion factor $\prod_{i=1}^{t}(i+1)/i = t + 1$, which grows linearly. Unfortunately, Damgård-Jurik is not ciphertext additive and cannot therefore be used in our construction. But Lipmaa's technique shows that linear growth in t is possible if the underlying schemes enjoy some suitable properties.

Finally, we would like to highlight that due to space limitations we have focused on SFE, which is very illustrative of our schemes' practicality. However, an homomorphic encryption scheme has many applications beyond SFE (see [5]). We hope our research will motivate the analysis of such applications.

References

1. Rivest, R.L., Adleman, L., Dertouzos, M.L.: On Data Banks and Privacy Homomorphisms. In: On Data Banks and Privacy Homomorphisms. Academic Press (1978) 169–180

2. Rivest, R., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public key cryptosystems. Communications of the ACM **21**(2) (1978) 120–126
3. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory **31**(4) (1985) 469–472
4. Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: 18th Annual Eurocrypt Conference (EUROCRYPT'99), Prague, Czech Republic. Volume 1592 of Lecture Notes in Computer Science., Springer (1999) 223–238
5. Rappe, D.K.: Homomorphic cryptosystems and their applications. Cryptology ePrint Archive, Report 2006/001 (2006) http://eprint.iacr.org/.
6. Fellows, M., Koblitz, N.: Combinatorial cryptosystems galore! In: Finite fields: theory, applications, and algorithms (Las Vegas, NV, 1993). Volume 168 of Contemp. Math., Amer. Math. Soc. (1994) 51–61
7. Steinwandt, R., Geiselmann, W.: Cryptanalysis of Polly Cracker. IEEE Transactions on Information Theory **48**(11) (2002) 2990–2991
8. Grigoriev, D., Ponomarenko, I.: Homomorphic public-key cryptosystems and encrypting boolean circuits. Applicable Algebra in Engineering, Communication and Computing 17(3), 239-255. (2006) **17** (2006) 239–255
9. Choi, S.J., Blackburn, S.R., Wild, P.R.: Cryptanalysis of a homomorphic public-key cryptosystem over a finite group. J. Math. Cryptography **1** (2007) 351–358
10. Fontaine, C., Galand, F.: A survey of homomorphic encryption for nonspecialists. EURASIP J. Inf. Secur. **2007**(1) (2007) 1–15
11. Boneh, D., Lipton, R.J.: Algorithms for black-box fields and their application to cryptography (extended abstract). In Koblitz, N., ed.: Advances in Cryptology—CRYPTO '96. Volume 1109 of Lecture Notes in Computer Science., Springer-Verlag (1996) 283–297
12. Feigenbaum, J., Merritt, M.: Open questions, talk abstracts, and summary of discussions. In: DIMACS Series in Discrete Mathematics and Theoretical Computer Science. Volume 2. (1991) 1–45
13. Domingo-Ferrer, J.: A new privacy homomorphism and applications. Information Processing Letters **60**(5) (1996) 277–282
14. Domingo-Ferrer, J.: A provably secure additive and multiplicative privacy homomorphism. In Chan, A.H., Gligor, V.D., eds.: Information Security, 5th International Conference, ISC 2002 Sao Paulo, Brazil, September 30 - October 2, 2002, Proceedings. Volume 2433 of Lecture Notes in Computer Science., Springer (2002) 471–483
15. Cheon, J.H., Kim, W.H., Nam, H.S.: Known-plaintext cryptanalysis of the domingo-ferrer algebraic privacy homomorphism scheme. Inf. Process. Lett **97**(3) (2006) 118–123
16. Wagner, D.: Cryptanalysis of an algebraic privacy homomorphism. In: Information Security, 6th International Conference, ISC 2003, Bristol, UK, October 1-3, 2003, Proceedings. Volume 2851 of Lecture Notes in Computer Science., Springer (2003) 234–239
17. Armknecht, F., Sadeghi, A.R.: A new approach for algebraically homomorphic encryption. Cryptology ePrint Archive, Report 2008/422 (2008) http://eprint.iacr.org/.
18. Kiayias, A., Yung, M.: Secure Games with Polynomial Expressions. In: ICALP: Annual International Colloquium on Automata, Languages and Programming. (2001)
19. Bleichenbacher, D., Kiayias, A., Yung, M.: Decoding of Interleaved Reed Solomon Codes over Noisy Data. In Baeten, J.C.M., Lenstra, J.K., Parrow, J., Woeginger, G.J., eds.: Automata, Languages and Programming, 30th International Colloquium, ICALP 2003, Eindhoven, The Netherlands, June 30 - July 4, 2003. Proceedings. Volume 2719 of Lecture Notes in Computer Science., Springer (2003) 97–108
20. Coppersmith, D., Sudan, M.: Reconstructing curves in three (and higher) dimensional space from noisy data. In: Proceedings of the 35th Annual ACM Symposium on Theory of Computing, STOC'2003 (San Diego, California, USA, June 9-11, 2003), New York, ACM Press (2003) 136–142
21. Boneh, D., Goh, E.J., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In Kilian, J., ed.: Theory of Cryptography, Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12, 2005, Proceedings. Volume 3378 of Lecture Notes in Computer Science., Springer (2005) 325–341
22. Sander, T., Young, A., Yung, M.: Non-interactive CryptoComputing for NC^1. In: Proceedings of the 40th Symposium on Foundations of Computer Science (FOCS), New York, NY, USA, IEEE Computer Society Press (1999) 554–567
23. Kawachi, A., Tanaka, K., Xagawa, K.: Multi-bit cryptosystems based on lattice problems. In Okamoto, T., Wang, X., eds.: Public Key Cryptography - PKC 2007, 10th International Conference on Practice and Theory in Public-Key Cryptography, Beijing, China, April 16-20, 2007, Proceedings. Volume 4450 of Lecture Notes in Computer Science., Springer (2007) 315–329

24. Goldreich, O., Goldwasser, S., Halevi, S.: Eliminating decryption errors in the ajtai-dwork cryptosystem. In Kaliski, Jr., B.S., ed.: Advances in Cryptology – CRYPTO ' 97. Volume 1294 of Lecture Notes in Computer Science., International Association for Cryptologic Research, Springer-Verlag, Berlin Germany (1997) 105–111
25. Regev, O.: New lattice based cryptographic constructions. In: Proceedings of the 35th Annual ACM Symposium on Theory of Computing, STOC'2003 (San Diego, California, USA, June 9-11, 2003), New York, ACM Press (2003) 407–416
26. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005, ACM Press (2005) 84–93
27. Ajtai, M.: Representing hard lattices with O(n log n) bits. In Gabow, H.N., Fagin, R., eds.: Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005, ACM (2005) 94–103
28. Phong Q. Nguyen: Cryptanalysis of the Goldreich-Goldwasser-Halevi Cryptosystem from Crypto '97. In: Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings. Volume 1666 of Lecture Notes in Computer Science., Springer (1999) 288–304
29. Aguilar Melchor, C., Castagnos, G., Gaborit, P.: Lattice-based homomorphic encryption of vector spaces. In: The 2008 IEEE International Symposium on Information Theory (ISIT'08), Toronto, Ontario, Canada, IEEE Computer Society Press (2008) 1858–1862
30. Lipmaa, H.: An Oblivious Transfer Protocol with Log-Squared Communication. In: 8th Information Security Conference (ISC'05), Singapore. Volume 3650 of Lecture Notes in Computer Science., Springer (2005) 314–328
31. Damgård, I., Jurik, M.: A Length-Flexible Threshold Cryptosystem with Applications. In: ACISP 2003. (2003) 350–364
32. Ostrovsky, R., E. Skeith III, W.: Private searching on streaming data. J. Cryptology **20**(4) (2007) 397–430
33. Adida, B., Wikström, D.: How to shuffle in public. In: TCC'07. Volume 4392 of Lecture Notes in Computer Science., Springer (2007) 555–574
34. Adida, B., Wikström, D.: Offline/online mixing. In: ICALP'07. Volume 4596 of Lecture Notes in Computer Science., Springer (2007) 484–495
35. Bringer, J., Chabanne, H., Pointcheval, D., Tang, Q.: Extended private information retrieval and its application in biometrics authentications. In: CANS'07. Volume 4856 of Lecture Notes in Computer Science., Springer (2007) 175–193
36. Kursawe, K., Neven, G., Tuyls, P.: Private policy negotiation. In: Financial Cryptography Conference, FC'06. Volume 4107 of Lecture Notes in Computer Science., Springer (2006) 81–95

A IND-CPA Security

We recall here the standard notion of indistinguishability under chosen-plaintext attacks (CPA security), for an encryption scheme $PKE = (\mathcal{KG}, \mathcal{E}, \mathcal{D})$. We use the following game that an attacker $\mathcal{A}$ plays against a challenger:

$$\begin{array}{l} (pk, sk) \leftarrow \mathcal{KG}(\cdot) \\ (St, m_0, m_1) \leftarrow \mathcal{A}(\text{find}, pk) \\ b \leftarrow \{0,1\} \text{ at random} \\ c^* \leftarrow \mathcal{E}_{pk}(m_b) \\ b' \leftarrow \mathcal{A}(\text{guess}, c^*, St). \end{array}$$

The advantage of such an adversary $\mathcal{A}$ is defined as

$$\mathrm{Adv}(\mathcal{A}) = \left| \Pr[b' = b] - \frac{1}{2} \right|.$$

A public key encryption scheme is said to be ε-indistinguishable under CPA attacks if $\mathrm{Adv}(\mathcal{A}) < \varepsilon$ for any attacker $\mathcal{A}$ which runs in polynomial time.

B Some Proofs

We include here some of the proofs of the lemmas of Section 3.3, and also the proof of Proposition 4 of Section 4.3.

B.1 Proofs of Lemmas in Section 3.3

Lemma 1. *Let $(\mathbb{Z}_s,+)^n$ be a group for $s,n \in \mathbb{Z}^+$. For any $k, L, s' \in \mathbb{Z}^+$ such that $(2^k-1)\cdot\ L <\ s' < s$ there is a computable L-pseudohomomorphism from $(\mathbb{Z}_s,+)^n$ to $(\mathbb{Z}_{s'},+)^{n'}$ where $n' = n \cdot \lceil(\log_2 s)/k\rceil$.*

*Proof. (*sketch) Let $n' = n \cdot \lceil(\log_2 s)/k\rceil$, define

$$\begin{array}{llll} \phi: & (\mathbb{Z}_s,+)^n & \to & (\mathbb{Z},+)^{n'} \\ & (x_1,\ldots,x_n) & \to & (y_1,\ldots,y_{n'}) \end{array} \qquad \begin{array}{llll} \phi^*: & (\mathbb{Z},+)^{n'} & \to & (\mathbb{Z}_s,+)^n \\ & (y_1,\ldots,y_{n'}) & \to & (z_1,\ldots,z_n) \end{array}$$

with $y_{(i-1)\cdot\lceil(\log_2 s)/k\rceil+j}$ being the j-th k-bit block of x_i seen as an integer in $\mathbb{Z}$ and,
$z_i = \left(\sum_{j=1}^{\lceil(\log_2 s)/k\rceil} 2^{(j-1)\cdot k} \cdot y_{(i-1)\cdot\lceil(\log_2 s)/k\rceil+j}\right) \bmod s$.

The fact that $\phi^*(\phi(\mathbf{x})) = \mathbf{x}$ is trivial. Note that ϕ^* is linear and thus for any sum $\sum_\ell \phi(\mathbf{x}_\ell)$ of images of ϕ,

$$\phi^*(\sum_\ell \phi(\mathbf{x}_\ell)) = \sum_\ell \phi^*(\phi(\mathbf{x}_\ell)) = \sum_\ell \mathbf{x}_\ell$$

Define $\psi : (\mathbb{Z}_s,+)^n \to (\mathbb{Z}_{s'},+)^{n'}$ and $\psi^* : (\mathbb{Z}_{s'},+)^{n'} \to (\mathbb{Z}_s,+)^n$ by $\psi(\mathbf{x}) = \phi(\mathbf{x}) \bmod s'$ for $\mathbf{x} \in (\mathbb{Z}_s,+)^n$ and $\psi^*(\mathbf{y}) = \phi^*(\mathbf{y})$ for $\mathbf{y} \in (\mathbb{Z}_{s'},+)^{n'}$. For any sum $\sum_\ell \psi(\mathbf{x}_\ell)$ of images of ψ,

$$\psi^*(\sum_\ell \psi(\mathbf{x}_\ell)) = \phi^*(\sum_\ell \phi(\mathbf{x}_\ell) \bmod s')$$

Which is equal to $\sum_\ell \mathbf{x}_\ell$ as long as the coordinates of $\sum_\ell \phi(\mathbf{x}_\ell)$ are smaller than s'. The sum of $k' \leq L$ images of ϕ results in elements with coordinates at most equal to $(2^k-1)\cdot L < s'$. (ψ, ψ^*) is thus a computable L-pseudohomomorphism from $(\mathbb{Z}_s,+)^n$ to $(\mathbb{Z}_{s'},+)^{n'}$.

Lemma 2. *For any $s_1, s_2, n \in \mathbb{Z}^+$ there is a computable L-pseudohomomorphism from $(\mathbb{Z}_{s_1},+)^n$ to $(\mathbb{Z}_{s_2},+)^n$ for $L = \lfloor s_2/s_1 \rfloor$.*

Proof. (sketch) As an L-pseudohomomorphism only makes sense for $L > 0$ we suppose that $s_2 > s_1$. Define $\phi : (\mathbb{Z}_{s_1},+)^n \to (\mathbb{Z}_{s_2},+)^n$ by $\phi((x_1,\ldots,x_n)) = (x_1,\ldots,x_n)$, and $\phi^* : (\mathbb{Z}_{s_2},+)^n \to (\mathbb{Z}_{s_1},+)^n$ by $\phi^*((y_1,\ldots,y_n)) = (y_1 \bmod s_1, \ldots, y_n \bmod s_1)$. For any $k \leq L$, and $z_1,\ldots,z_k < s_1$, we have $\sum_{i=1}^k z_i < k \cdot s_1$ (in $\mathbb{Z}$) and thus $\sum_{i=1}^k z_i < s_2$. This fact directly implies that $\phi^*(\phi(\boldsymbol{x}_1) + \ldots, \phi(\boldsymbol{x}_k)) = \boldsymbol{x}_1 + \ldots + \boldsymbol{x}_k \bmod s_1$, for any k elements $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_k \in \mathbb{Z}_{s_1}^n$. Thus, (ϕ, ϕ^*) is an L-pseudohomomorphism from $(\mathbb{Z}_{s_1},+)^n$ to $(\mathbb{Z}_{s_2},+)^n$.

Lemma 3. *A plaintext additive L-pseudohomomorphic scheme PKE with plaintext space $(\mathbb{Z}_s,+)^n$ can be transformed into a plaintext additive L-pseudohomomorphic scheme PKE' with plaintext space $(\mathbb{Z}_s,+)^{kn}$, for any $k \in \mathbb{Z}^+$. This transformation preserves indistinguishability.*

Proof. (sketch) We can use the direct product to define $\mathcal{E}'' : (\mathbb{Z}_s,+)^n \times (\mathbb{Z}_s,+)^n \to (\mathcal{C},\oplus) \times (\mathcal{C},\oplus)$ with $\mathcal{E}''((x_1,x_2)) = (\mathcal{E}(x_1), \mathcal{E}(x_2))$. Similarly, we define $\mathcal{D}'' : (\mathcal{C},\oplus)\times(\mathcal{C},\oplus) \to (\mathbb{Z}_s,+)^n \times (\mathbb{Z}_s,+)^n$ by $\mathcal{D}''(y_1,y_2) = (D(y_1), D(y_2))$, and $\mathcal{KG}'' = \mathcal{KG}$. $PKE'' = (\mathcal{KG}'', \mathcal{E}'', \mathcal{D}'')$ is a plaintext additive L-pseudohomomorphic encryption scheme. If PKE is IND-CPA, then PKE'' is IND-CPA by a standard hybrid argument. Using this construction recursively we obtain PKE' for any k.

Lemma 4. *For any $s,n \in \mathbb{Z}^+$ and any $\ell < n$ we define $\pi^{-1} : (\mathbb{Z}_s,+)^\ell \to (\mathbb{Z}_s,+)^n$ by $\pi^{-1}((x_1,\cdots,x_\ell)) = (x_1,\cdots,x_\ell,0,\cdots,0)$, where π is the standard projection. (π^{-1},π) is a computable ∞-pseudohomomorphism from $(\mathbb{Z}_s,+)^\ell$ to $(\mathbb{Z}_s,+)^n$.*

*Proof. T*rivial.

Lemma 3 and 4 prove that it is possible to change the dimension of the plaintext space of a plaintext additive pseudohomomorphic encryption scheme, without changing the order, which proves the last corollary.

Corollary 2. *A plaintext additive L-pseudohomomorphic scheme PKE with plaintext space $(\mathbb{Z}_s,+)^n$ can be transformed into a plaintext additive L-pseudohomomorphic scheme PKE' with plaintext space $(\mathbb{Z}_s,+)^\ell$, for any $\ell \in \mathbb{Z}^+$. This transformation preserves indistinguishability.*

B.2 Proof of Proposition 4

Proposition 4. *If t encryption schemes $\mathcal{E}_1,\ldots,\mathcal{E}_t$ are all ε-indistinguishable under CPA attacks, then the global encryption scheme $\mathcal{E}_G(x) = (\mathcal{E}_1(x),\ldots,\mathcal{E}_t(x))$ is $t\varepsilon$-indistinguishable under CPA attacks.*

Proof. Let us start with the case $t = 2$, i.e. $\mathcal{E}_G = (\mathcal{E}_1,\mathcal{E}_2)$. Assume that there exists a CPA attack $\mathcal{A}$ against $\mathcal{E}_G$ with advantage ε'. Let us construct, then, an attacker $\mathcal{A}'$ whose goal is to attack either $\mathcal{E}_1$ or $\mathcal{E}_2$.

In the first stage, $\mathcal{A}'$ receives a public key pk_1 for $\mathcal{E}_1$ and a public key pk_2 for $\mathcal{E}_2$. Then, $\mathcal{A}'$ initializes $\mathcal{A}$ with the global public key $PK_G = (pk_1, pk_2)$. The hypothetical attacker $\mathcal{A}$ outputs two messages m_0, m_1. $\mathcal{A}'$ gives the same messages m_0, m_1 to the two challengers (one for $\mathcal{E}_1$ and the other one for $\mathcal{E}_2$). As a result, $\mathcal{A}'$ receives a ciphertext $\mathcal{E}_1(m_{b_1})$ from the first challenger and a ciphertext $\mathcal{E}_2(m_{b_2})$ from the second challenger, where $b_1, b_2 \in \{0,1\}$ are random bits.

$\mathcal{A}'$ sends then to $\mathcal{A}$ the challenge ciphertext $c^* = (\mathcal{E}_1(m_{b_1}), \mathcal{E}_2(m_{b_2}))$. At some point, $\mathcal{A}$ outputs his guess b'. Finally, the new attacker $\mathcal{A}'$ outputs the same guess $b'_1 = b'_2 = b'$ for the two games he is playing.

If $b_1 = b_2$ (which happens with probability $1/2$), then $\mathcal{A}$ outputs the correct bit $b' = b_1$ with probability $1/2 + \varepsilon'$. IF $b_1 \neq b_2$ (which happens again with probability $1/2$), let us denote by δ_i the probability that $\mathcal{A}$ outputs $b' = b_i$, for $i = 1, 2$. Let $\delta_j = \max\{\delta_1, \delta_2\} \geq 1/2$, where $j \in \{1, 2\}$. Then, we have that the probability that $\mathcal{A}'$ guesses the bit b_j is

$$\frac{1}{2} \cdot \left(\frac{1}{2} + \varepsilon'\right) + \frac{1}{2} \cdot \delta_j \;\geq\; \frac{1}{2} + \frac{\varepsilon'}{2}.$$

Summing up, $\mathcal{A}'$ would have an advantage $\varepsilon'/2$ in breaking the semantic security of the scheme $\mathcal{E}_j$. If $\varepsilon' \geq 2\varepsilon$, this would mean that the attacker $\mathcal{A}'$ would break the ε-indistinguishability of $\mathcal{E}_j$, which is a contradiction with the hypothesis of the proposition. Therefore, we conclude that $\mathcal{E}_G$ is 2ε-indistinguishable.

Applying this result in a recursive way, starting with $(\mathcal{E}_1,\mathcal{E}_2), (\mathcal{E}_3,\mathcal{E}_4), \ldots$, then $(\mathcal{E}_1,\ldots,\mathcal{E}_4), (\mathcal{E}_5,\ldots,\mathcal{E}_8), \ldots$, and so on, we obtain in $\log t$ steps the result stated in this proposition.

C Applications of SFE

Here we list some particular cases of secure function evaluation of polynomials where the t-chained schemes could be used. Actually, our schemes can be seen in some way as a generalization of the scheme by Boneh, Goh and Nissim [21]. Therefore, our solution can be applied in many of the cases where the Boneh-Goh-Nissim scheme has been applied, sometimes adding some new functionalities, since our solution allows the secure evaluation of polynomials of any degree t (not only $t = 2$).

Private searching. A clear example is the topic of private searching on streaming data, introduced in [32]. There, the authors propose a solution, based on the Boneh-Goh-Nissim cryptosystem, to filter messages from a stream. The programmer can choose a subset $\mathcal{S} \subset \mathcal{D} \times \mathcal{D}$, $\mathcal{D}$ being a public dictionary, and filter out the messages containing the couples of keywords in $\mathcal{S}$ (for example the couple `(secret,bomb)`), without revealing $\mathcal{S}$. Our schemes allow to solve this problem in the case of tuples of keywords in $\mathcal{D}^t$ for any t.

t-DNF formula evaluation. Boneh, Goh and Nissim explain in their paper [21] how to apply their cryptographic scheme for evaluating a 2-DNF formula. Our schemes could in principle be used to evaluate any t-DNF formula, for bigger values of t. A t-DNF formula is defined by:

$$\phi(x_1,\ldots,x_n) = \bigvee_{j=1}^{q} (\ell_{j,1} \wedge \ell_{j,2} \wedge \ldots \wedge \ell_{j,t}),$$

where $\ell_{j,i} \in \{x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n\}$. In order to evaluate this formula for a secret assignment $\boldsymbol{a} = (a_1, \ldots, a_n) \in \{0,1\}^n$, we define the polynomial $P(x_1, \ldots, x_n) = \sum_{j=1}^{q} \ell_{j,1} \cdot \ldots \cdot \ell_{j,t}$ over $\mathbb{Z}_s$ for $s > q$ and crypto-compute $P(\boldsymbol{a})$. As highlighted in [21], we have $\phi(\boldsymbol{a}) = 1 \Leftrightarrow P(\boldsymbol{a}) \neq 0$ which ensures that the function is correctly evaluated. Boneh Goh and Nissim use this technique to improve a set of cryptographic protocols among which a Private Information Retrieval scheme whose communication complexity drops from $n^{1/2}$ to $n^{1/3}$. Of course, using t-chained schemes, this complexity drops to $n^{1/(t+1)}$.

Other uses. Other cryptographic functionalities where both the scheme by Boneh, Goh and Nissim and our scheme can be applied include mixing and shuffling [33, 34] or extended private information retrieval [35]. Finally, the work [36] on private policy negotiation is an example where the scheme of Boneh, Goh and Nissim cannot be used, because it allows only one multiplication, but our schemes could be applied.

D Symmetrically Secure Function Evaluation

When using t-chained schemes for the secure evaluation of a t-degree multivariate polynomial $F(x_1, \ldots, x_n)$, the only privacy property that can be guaranteed in general is privacy of Bob's inputs, as we have discussed in Section 4.3. Recall that this privacy property is enough for many of the practical applications of secure function evaluation. On the other hand, the secure evaluation techniques explained in Sections 4.1 and 4.2 cannot always ensure the privacy of the function held by Alice . To illustrate this fact, suppose we try to securely evaluate a polynomial by using a t-chained scheme, where the last encryption scheme $\mathcal{E}$ in the chain is such that the distribution of the ciphertext $\mathcal{E}(m_1) + \ldots + \mathcal{E}(m_\ell)$ depends on the number ℓ of sums. In this case, when Bob receives the final encryption of $F(a_1, \ldots, a_n)$, he can obtain some information about the number of monomials in the polynomial, breaking in this way the intended privacy for Alice.

These problems can be overcome if we assume that all the encryption schemes in the t-chain enjoy a suitable *uniformity* property.

Definition 4. *A (pseudo)homomorphic public key encryption scheme $PKE = (\mathcal{KG}, \mathcal{E}, \mathcal{D})$, with randomness space $\mathcal{R}$ for encryption, is said to have the uniformity property if there exists a probability distribution $r \hookleftarrow \mathcal{R}$ such that*

$$\{\mathcal{E}(m_1, r_1) + \mathcal{E}(0, r) \mid r \hookleftarrow \mathcal{R}\} \text{ and } \{\mathcal{E}(m_2, r_2) + \mathcal{E}(0, r) \mid r \hookleftarrow \mathcal{R}\}$$

are computationally indistinguishable, for any messages m_1, m_2 and random values r_1, r_2.

The most well-known homomorphic encryption schemes (ElGamal, Paillier, Boneh-Goh-Nissim) enjoy this property, taking $r \hookleftarrow \mathcal{R}$ as the uniform random distribution. However, for the encryption schemes based on lattices, particularly those that we consider in Section 5, it is not known if this uniformity property can be enjoyed. Our intuition is that some of them will not, but that there can be other (secure) instantiations enjoying this property. Studying this property for particular lattice-based schemes remains as a challenging open problem.

In any case, assuming that we have a t-chain where all the encryption schemes enjoy the uniformity property, we can slightly modify the SFE techniques explained in Sections 4.1 and 4.2 to achieve privacy for the polynomial F (Alice's input). The idea is that Alice must replace $\mathcal{E}(a_1) + \ldots + \mathcal{E}(a_\ell)$ with $\mathcal{E}(a_1) + \ldots + \mathcal{E}(a_\ell) + \mathcal{E}(0, r)$, where $r \hookleftarrow \mathcal{R}$, each time she computes an encryption of a sum of plaintexts (in particular, each time she computes an encryption of a product of two plaintexts, as well). In this way, each ciphertext decrypted by Bob contains information on the associated plaintext only, and not on the operations it has undergone.