Additive 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] IIIA-CSIC,
Campus de la UAB,
08193 Bellaterra, Spain
jherranz@iiia.csic.es

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. Previous solutions in the literature worked for the case $t = 2$. Our solution is not only theoretical: we show that some existing (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.

1 Introduction

Nowadays, there are many digital situations where users want to obtain some information which involves private information of other users, in such a way that the final and desired information is obtained, but nothing about the private inputs of the users is leaked. A well-known example is (Symmetric) Private Information Retrieval (PIR) [1], where a user U wants to obtain the i-th entry of a database held by a different user S. User U wants to keep private the value of i, so S does not obtain any information about it. Furthermore, in the symmetric case, U must not obtain any information about the other entries of the database, different from the i-th one.

This kind of protocols receive the generic name of *Secure Function Evaluation* (SFE) [2]. Each user U_j holds a private input a_j, they engage an interactive protocol and, at the end, some (maybe all) of the users obtain $f(a_1, \ldots, a_n)$, for some function f which may be public or may be part of the secret input of some of the users. A very important particular case of SFE is that where f is a multivariate polynomial. We will focus on this particular case. When the polynomial to be evaluated is of degree $t = 1$ (i.e., $f(a_1, \ldots, a_n) = \sum_j c_j a_j$ for some values c_j), then the problem of SFE can be solved by using additively homomorphic encryption schemes. In such schemes, there exists some operation $\otimes$ defined in the set of ciphertexts, such that (informally) $\mathcal{E}(m_1) \otimes \mathcal{E}(m_2) = \mathcal{E}(m_1 + m_2)$, where $\mathcal{E}$ denotes the encryption function.

However, when the degree of f is larger (involving in particular products, e.g. $f(a_1, \ldots, a_4) = a_1a_2 + a_1a_3a_4$), standard homomorphic schemes are not enough. Intuitively, what we would need in this case is a cryptographic mechanism that, given encryptions of the private inputs, allows to compute encryptions of both sums and products of the inputs. In [3], such cryptographic mechanisms received the name of *algebraic homomorphic* encryption schemes. The existence of such schemes has been left as an open problem for many years (see [4, 5] for some related work). The most important contribution towards a solution to this problem was done by Boneh, Goh and Nissim [6]. They propose a new secure encryption scheme which allows to compute polynomials with degree at most $t = 2$. A proof of the importance of this result, and of the number of practical applications of SFE, is the huge number of papers which discuss, cite or are based on the mechanism proposed in [6].

Our Contribution

Our goal is to give more steps towards a solution of the generic problem, when the degree of the formula is bounded by some value t possibly bigger than 2. To do this, we define a theoretical cryptographic object, that we denote as *t-chained encryption scheme*, which is composed of different (pseudo-)homomorphic encryption schemes satisfying some conditions. This encryption scheme can evaluate any polynomial of degree t. 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. 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). We study in detail two possible particular instances for the case $t = 3$, to illustrate this problem. However, any future advance in the area of lattice-based (pseudo-)homomorphic schemes will immediately have an impact on the implementability of our solutions.

Organization of the Paper

We recall in Section 2 some basic concepts on (pseudo-)homomorphic encryption schemes. In Section 3 we describe the general idea and a simple example of our construction. The main part of our work is presented in Section 4, where the concept of t-chained encryption scheme is introduced, after having proved some technical results involving (pseudo)-homomorphisms, which can be of independent interest. In Section 5 we explain how to compute the sum and product of integer private inputs with a t-chained encryption scheme, and briefly present the applications of such computations. Then we give in Section 6 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. Some concluding remarks are given in Section 7.

2 Preliminaries

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 not explicitly include the randomness as an input of the encryption functions.

2.1 *L*-Pseudo-homomorphic Encryption Schemes

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

$$\mathcal{D}_{sk}\big(\mathcal{E}_{pk}(m_1) \otimes \mathcal{E}_{pk}(m_2)\big) \;=\; m_1 \oplus m_2,$$

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

This basic pseudo-homomorphic property $\mathcal{D}_{sk}\big(\mathcal{E}_{pk}(m_1) \otimes \mathcal{E}_{pk}(m_2)\big) = m_1 \oplus m_2$ does not imply $\mathcal{E}_{pk}(m_1) \otimes \mathcal{E}_{pk}(m_2) = \mathcal{E}_{pk}(m_1 \oplus m_2)$ but just $\mathcal{E}_{pk}(m_1) \otimes \mathcal{E}_{pk}(m_2) \in \mathcal{D}_{sk}^{-1}(m_1 \oplus m_2)$. This is important as often the function $\mathcal{E} : \mathcal{M} \to \mathcal{C}$ is not surjective. In order to avoid cumbersome notations, $\tilde{\mathcal{E}}_{pk}(x)$ will represent an element of $\mathcal{D}_{sk}^{-1}(x)$ just as $\mathcal{E}_{pk}(x)$ has been representing an element of $\{\mathcal{E}_{pk}(x, r) | r \in \mathcal{R}\}$. We thus have $\mathcal{E}_{pk}(m_1) \otimes \mathcal{E}_{pk}(m_2) = \tilde{\mathcal{E}}_{pk}(m_1 \oplus m_2)$. With these ideas in mind, one can consider the following definition.

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

If $\mathcal{E}_{pk}(m_1) \otimes \mathcal{E}_{pk}(m_2) = \mathcal{E}_{pk}(m_1 \oplus m_2)$, then we can iteratively apply this result to deduce that $\mathcal{E}_{pk}(m_1) \otimes \ldots \otimes \mathcal{E}_{pk}(m_k) = \mathcal{E}_{pk}(m_1 \oplus \ldots \oplus m_k)$ for any k. We will say that such encryption schemes are ∞-pseudo-homomorphic (or simply homomorphic). Note that if $\mathcal{E}$ is surjective and 2-pseudo-homomorphic then $\mathcal{E}_{pk}(m_1) \otimes \mathcal{E}_{pk}(m_2) = \mathcal{E}_{pk}(m_1 \oplus m_2)$ and thus $\mathcal{E}$ is homomorphic.

2.2 Semantic Security

We recall the standard notion of security for public key encryption schemes in terms of *indistinguishability*, or *semantic security*. We consider chosen-plaintext attacks (CPA), because homomorphic schemes can never achieve security against chosen-ciphertext attacks. To define security, we use the following game that an attacker $\mathcal{A}$ plays against a challenger:

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

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

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

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

From this definition, it is quite obvious that the role of the randomness r is crucial to ensure (semantic) security of a public key encryption scheme. In effect, a deterministic scheme can never be semantically secure, because an attacker $\mathcal{A}$ could always encrypt m_0 and m_1 by using pk, and then compare the resulting ciphertexts with the challenge one c^*, to decide the value of the bit b.

3 Overview of Our Solution

3.1 Basic Idea

Finding a ring homomorphic encryption scheme has been unsuccessful for many years. Very informally, what was sought is some encryption function $\mathcal{E} : \mathbb{Z}_{s_1} \to \mathbb{Z}_{s_2}$ such that:

$$
\begin{aligned}
\mathcal{E}(a_1) + \mathcal{E}(a_2) &= \mathcal{E}(a_1 + a_2) \\
\mathcal{E}(a_1) \cdot \mathcal{E}(a_2) &= \mathcal{E}(a_1 \cdot a_2).
\end{aligned}
$$

Some cryptosystems ensure naturally that $\mathcal{E}(a_1) \cdot \mathcal{E}(a_2) = \mathcal{E}(a_1 \cdot a_2)$, as for example El Gamal's [7]. On the other hand, finding a cryptosystem such that $\mathcal{E}(a_1) + \mathcal{E}(a_2) = \mathcal{E}(a_1 + a_2)$ was not trivial. Since adding plaintexts through ciphertext manipulation was needed by many applications, an alternative was found. Indeed, an increasing family of cryptosystems, among which Paillier's [8] is probably the most well-known, had encryption functions such that $\mathcal{E}(a_1) \cdot \mathcal{E}(a_2) = \mathcal{E}(a_1 + a_2)$. Finally, recent lattice-based cryptosystems such that $\mathcal{E}(a_1) + \mathcal{E}(a_2) = \mathcal{E}(a_1 + a_2)$ have been proposed [9].

However, for a long time no cryptosystem has allowed at the same time to sum up and multiply plaintexts through two different ciphertext operations. A very important step forward was done by Boneh, Goh and Nissim in [6], where they proposed a scheme such that $\mathcal{E}(a_1) \cdot \mathcal{E}(a_2) = \mathcal{E}(a_1 + a_2)$ and such that there is a bilinear map $e(\cdot,\cdot)$ verifying $e(\mathcal{E}(a_1), \mathcal{E}(a_2)) = e(\mathcal{E}(a_1 \cdot a_2), 1)$. Noting $\mathcal{E}'(x) = e(\mathcal{E}(x), 1)$ they obtained an encryption function $\mathcal{E}'$ such that:

$$\begin{aligned} &\mathcal{E}'(a_1) \cdot \mathcal{E}'(a_2) = \mathcal{E}'(a_1 + a_2) \text{ (since } e(\cdot,\cdot) \text{ is bilinear) and} \\ &\mathcal{E}(a_1) \otimes \mathcal{E}(a_2) = \mathcal{E}'(a_1 \cdot a_2), \text{ where } y_1 \otimes y_2 := e(y_1, y_2). \end{aligned}$$

Therefore, starting from encryptions $\mathcal{E}(a_1)$ and $\mathcal{E}(a_2)$, it is possible to obtain encryptions (according to $\mathcal{E}'$) of both $a_1 + a_2$ and $a_1 \cdot a_2$.

This encryption function has many practical applications, even if it only allows to do one multiplication. Moreover, besides the practical interest of such a function, the scheme of Boneh, Goh and Nissim highlights a very important idea: we do not need to use the same encryption scheme all the time. In this paper, we propose a construction that results in an encryption function $\mathcal{E}$ derived from t different homomorphic encryption functions $\mathcal{E}_1, \ldots, \mathcal{E}_t$ such that:

$$\begin{aligned} \mathcal{E}(a_1) + \mathcal{E}(a_2) &= \mathcal{E}(a_1 + a_2), \text{ and} \\ \mathcal{E}_1(a_1) \otimes \ldots \otimes \mathcal{E}_t(a_t) &= \mathcal{E}(a_1 \cdot \ldots \cdot a_t), \text{ for a given operation } \otimes. \end{aligned}$$

which allows us to do t multiplications. This construction is valid for any t and we will provide practical constructions for $t = 3$.

3.2 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 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 equality being verified as the homomorphic property of $\mathcal{E}_2$ implies $k \cdot \mathcal{E}_2(x) = \mathcal{E}_2(k \cdot x)$. We thus have:

$$\begin{aligned} &\mathcal{E}(a_1) + \mathcal{E}(a_2) = \mathcal{E}(a_1 + a_2) \text{ and} \\ &\mathcal{E}_1(a_1) \cdot \mathcal{E}_2(a_2) = \mathcal{E}(a_1 \cdot a_2). \end{aligned}$$

Of course, this construction can be generalized if we obtain additively homomorphic 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 additively homomorphic encryption schemes that we could use to realize this 'ideal world' construction only provide an L-homomorphic property and not a real homomorphism. Moreover, in many cases we do not have $\mathcal{E} : \mathbb{Z}_s \rightarrow \mathbb{Z}_{s'}$, but $\mathcal{E} : \mathbb{Z}_s \rightarrow \mathbb{Z}_{s'}^{n'}$, or $\mathcal{E} : \mathbb{Z}_s^n \rightarrow \mathbb{Z}_{s'}^{n'}$.

In fact, we will use a cryptosystem such that $\mathcal{E} : \mathbb{Z}_s \rightarrow \mathbb{Z}_{s'}$ (s' being very large) in our practical constructions in Section 6. However, using it directly leads to a very inefficient construction. We will thus modify it to obtain an encryption function $\mathcal{E} : \mathbb{Z}_s \rightarrow \mathbb{Z}_{s''}^{n''}$ with a small s'' and a large n''.

Sections 4.1 and 4.2 show how to modify the existing encryption schemes to fit our needs. Then, in Section 4.3 we provide a general construction derived from the one we have just presented, and in Section 5 we prove the homomorphic properties of the cryptosystems obtained through this construction.

4 t-Chained Pseudo-homomorphic 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-pseudo-homomorphisms, and prove some of their properties through a small set of lemmas and propositions.

4.1 Extending Pseudo-homomorphic Encryption Schemes

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

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

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

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-pseudo-homomorphism from $(G_1, \oplus_1)$ to $(G_2, \oplus_2)$.

Pseudo-homomorphisms can be combined with pseudo-homomorphic 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-pseudo-homomorphic encryption scheme such that there is a computable L'-pseudo-homomorphism (ϕ, ϕ^*) (ϕ being public) from a given space to PKE's plaintext space, the associated encryption scheme $PKE' = (\mathcal{KG}, \mathcal{E} \circ \phi, \phi^* \circ \mathcal{D})$ is a $min(L, L')$-pseudo-homomorphic encryption scheme. If there exists a public computable L''-pseudo-homomorphism (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 $min(L, L'')$-pseudo-homomorphic encryption scheme. Moreover, in both cases, if PKE is IND-CPA, PKE' is IND-CPA too.*

Proof. (sketch) The fact that PKE' is $min(L, L')$ or $min(L, L'')$-pseudo-homomorphic is trivial. 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, ψ being public, distinguishing x_2, x_2 in PKE' implies distinguishing them also in PKE.

4.2 Twisting Additive Pseudo-homomorphic Encryption Schemes

In many applications, homomorphic encryption schemes are used to sum up plaintext integers. If for a given L-pseudo-homomorphic 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*. These schemes can be easily modified using Proposition 1 and simple L-pseudo-homomorphisms.

When a plaintext or ciphertext space is $(\mathbb{Z}_s, +)^n$ we will call s its order and n its dimension.

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-pseudo-homomorphism 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 $\phi : (\mathbb{Z}_s, +)^n \to (\mathbb{Z}_{s'}, +)^{n'}$ by $\phi((x_1, \ldots, x_n)) = (y_1, \ldots, y_{n'})$ with $y_{(i-1)\cdot\lceil(\log_2 s)/k\rceil+j}$ being the $j-th$ k-bit block of x_i. The sum of L images of ϕ results in elements with coordinates at most equal to $(2^k - 1) \cdot L$ and thus is strictly smaller than s'. Defining $\phi^*(y_1, \ldots, y_{n'}) = (x_1, \ldots, x_n)$ with $x_i = \sum_{j=1}^{\lceil(\log_2 s)/k\rceil} 2^{(j-1)\cdot k} \cdot y_{(i-1)\cdot\lceil(\log_2 s)/k\rceil+j} \bmod s$, we have that (ϕ, ϕ^*) form an L-pseudo-homomorphism from $(\mathbb{Z}_s, +)^n$ to $(\mathbb{Z}_{s'}, +)^{n'}$.

Corollary 1. *Let PKE be a ciphertext additive L-pseudo-homomorphic 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 pseudo-homomorphic properties and indistinguishability. In other words, it is possible to lower the ciphertext space order by increasing its dimension. 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 $(\mathbb{Z}_2, +)^{m\cdot\log_2 s}$ (vectors of bits). To preserve the L-pseudo-homomorphic 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 transformation explained in Corollary 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 lemma gives a tool to do the opposite operation: increasing the ciphertext space order of a ciphertext additive scheme. Furthermore, the same argument can be applied to the plaintext, as well, which means that the plaintext space order can be lowered. This will be very useful in order to match the ciphertext space order of one scheme and the plaintext space order or the following scheme, in the chains of schemes that we will introduce later.

Lemma 2. *For any $s_1, s_2, n \in \mathbb{Z}^+$ there is a computable L-pseudo-homomorphism 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-pseudo-homomorphism 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-pseudo-homomorphism from $(\mathbb{Z}_{s_1}, +)^n$ to $(\mathbb{Z}_{s_2}, +)^n$.

Now that we have shown how to change the order of the ciphertext and plaintext spaces, let us explain how to change the dimension of the plaintext space of a plaintext additive encryption scheme. Combining this result with the previous ones, we will be able to modify the schemes in a given chain in such a way that the ciphertext space of a scheme is exactly equal to the plaintext space of the following scheme in the chain.

Lemma 3. *A plaintext additive L-pseudo-homomorphic scheme PKE with plaintext space $(\mathbb{Z}_s,+)^n$ can be transformed into a plaintext additive L-pseudo-homomorphic 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 $\mathcal{E}'' : (\mathbb{Z}_s,+)^n \times (\mathbb{Z}_s,+)^n \to (\mathcal{C},\otimes) \times (\mathcal{C},\otimes)$ with $\mathcal{E}''((x_1,x_2)) = (\mathcal{E}(x_1),\mathcal{E}(x_2))$. Similarly, we define $\mathcal{D}'' : (\mathcal{C},\otimes) \times (\mathcal{C},\otimes) \to (\mathbb{Z}_s,+)^n \times (\mathbb{Z}_s,+)^n$ 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-pseudo-homomorphic 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 ∞-pseudo-homomorphism from $(\mathbb{Z}_s,+)^\ell$ to $(\mathbb{Z}_s,+)^n$.*

*Proof. T*rivial.

These two lemmas prove that it is possible to change the dimension of the plaintext space of a plaintext additive pseudo-homomorphic encryption scheme, without changing the order.

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

4.3 *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 pseudo-homomorphic encryption scheme PKE_2 in order to encrypt the ciphertexts of a plaintext and ciphertext additive pseudo-homomorphic encryption scheme PKE_1, in such a way that the imbrication of both schemes leads to a new plaintext additive pseudo-homomorphic encryption scheme called 2-chained. Chained schemes are used in the following section for crypto-computing multiplications.

Definition 3. *Let $PKE_1 = (\mathcal{KG}_1,\mathcal{E}_1,\mathcal{D}_1)$ be a plaintext and ciphertext additive L_1-pseudo-homomorphic 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-pseudo-homomorphic 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-pseudo-homomorphic 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: $\mathcal{KG} = \mathcal{KG}_1 \times \mathcal{KG}'_2$; $\mathcal{E} = \mathcal{E}'_2 \circ \mathcal{E}_1$; $\mathcal{D} = \mathcal{D}_1 \circ \mathcal{D}'_2$.

Note that 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. This can be achieved by using Corollary 1 and Lemma 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 pseudo-homomorphic properties of such a scheme.

Proposition 2. *A 2-chained encryption scheme PKE is a plaintext additive L-pseudo-homomorphic 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-pseudo-homomorphism $(\mathcal{E}_1, \mathcal{D}_1)$ or as an extension of PKE'_1 with the L_2-pseudo-homomorphism $(\mathcal{E}'_2, \mathcal{D}'_2)$. In any case, Proposition 1 proves that the resulting 2-chained scheme is a plaintext additive $min(L_1, L_2)$-pseudo-homomorphic encryption scheme, and IND-CPA secure if any of PKE_1 or PKE_2 are IND-CPA secure.

The lemmas of Section 4.2 show that in order to obtain 2-chained schemes, the only thing we need is to be able to create plaintext and ciphertext additive pseudo-homomorphic 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-pseudo-homomorphic encryption schemes such that the plaintext space order can be chosen arbitrarily large, it is possible to construct a 2-chained L-pseudo-homomorphic encryption scheme.*

Proof. (sketch) Suppose that, for any positive integer s, we can take a ciphertext and plaintext additive L-pseudo-homomorphic encryption scheme PKE such that $\mathcal{E} : (\mathbb{Z}_s, +)^n \rightarrow (\mathbb{Z}_{s'}, +)^{n'}$, n, s' and n' possibly being functions 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-pseudo-homomorphic 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-pseudo-homomorphic scheme PKE''_1 (as $\lfloor s_2/s_1 \rfloor > L$).

PKE''_1 (resp. PKE'_2) is L-pseudo-homomorphic 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-pseudo-homomorphic 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 pseudo-homomorphic schemes $PKE_1, PKE'_2, \ldots, PKE'_{t-1}$ and of one plaintext pseudo-homomorphic scheme PKE'_t (all of them resulting from properly twisting some initial schemes, as explained in Section 4.2). The encryption diagram of a t-chained scheme would be

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

Propositions 2 and 3 are trivially generalized. This scheme is thus L-pseudo-homomorphic, with $L = min(L_1, \ldots, L_t)$ if PKE_i is L_i-pseudo-homomorphic.

In [9], Kawachi, Tanaka and Xagawa propose a set of lattice-based ciphertext and plaintext additive pseudo-homomorphic encryption schemes, derived from [10–13], in which L and the plaintext space order can be set to any value in $\mathbb{Z}^+$. These cryptosystems can thus be used to implement t-chained encryptions schemes in practice.

5 Crypto-Computing with t-Chained Schemes

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

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

5.1 Crypto-Computing 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-pseudo-homomorphic with $L = min(L_1, L_2)$. Indeed, if $a_1, a_2 \in \mathbb{Z}_{s_1}$, we can consider the ciphertexts $C_i = \mathcal{E}(a_i) = \mathcal{E}_2'(\mathcal{E}_1(a_i)) \in (\mathcal{C}_2, \otimes_2)^{n_2}$, for $i = 1, 2$. Then, if $L \geq 2$, anyone can operate these ciphertexts to obtain $C = C_1 \otimes_2 C_2 = \tilde{\mathcal{E}}(a_1 + a_2)$. 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 crypto-computation 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 $c_2 = \mathcal{E}_2(a_2) \in (\mathcal{C}_2, \otimes_2)$. We write $c_1 = (\mathcal{E}_1^{(1)}(a_1), \ldots, \mathcal{E}_1^{(n_2)}(a_1))$, where $\mathcal{E}_1^{(l)}(a_1) \in \mathbb{Z}_{s_2}$, for $l = 1, \ldots, n_2$. Obviously, $\mathcal{E}_1^{(l)} : (\mathbb{Z}_{s_1}, +)^{n_1} \to (\mathbb{Z}_{s_2}, +)$ is a L_1-pseudo-homomorphism. We compute:

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

where the first operation has to be interpreted as 'applying $\otimes_2$ 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 4.2. 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.

5.2 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 4.3). Therefore, this ciphertext can be represented as $(\tilde{\mathcal{E}}_2'^{(1)}(\tilde{\mathcal{E}}_1(a_1a_2)), \ldots, \tilde{\mathcal{E}}_2'^{(n_3)}(\tilde{\mathcal{E}}_1(a_1a_2)))$. The same procedure as before (i.e. multiplying each component of this vector with $\mathcal{E}_3(a_3)$) can be applied to obtain:

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

Note that the number of operations involving ciphertexts of the different encryption functions in the chain increases when 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 $C_{1,1}, \ldots, C_{n,t}$ of the values $a_1, \ldots, a_n$, and anyone can compute an encryption $\widetilde{C}$ of the value $F(a_1, \ldots, a_n) \bmod s_1$. We represent such a polynomial by $F(a_1, \ldots, a_n) = \sum_{i=1}^{m} c_i M_i$, M_i being a monomial of degree at most t. In order to evaluate the polynomial:

1. Each monomial M_i is computed by multiplying the associated ciphertexts;
   If the monomial has a degree $t' < t$ it is then multiplied by $t - t'$ encryption of 1;
2. The monomial is added to itself c_i 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 $L_1 > m \cdot c \cdot a^{t-1}$ and $L_i > m \cdot c \cdot a^{t-i} \| \mathcal{E}_{i-1} \|_\infty$, for $i = 2, \ldots, t$, noting c and a the maximum possible values for $c_1, \ldots, c_m$ and $a_1, \ldots, a_n$ respectively.

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

5.3 Applications

Here we list some particular cases of private 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 [6]. Therefore, our solution can be applied in all the cases where the Boneh-Goh-Nissim's scheme has been applied, and maybe adding some new functionalities, since our solution allows crypto-computation of formulas of any degree t (not only $t = 2$).

Private searching. A clear example is the topic of private searching on streaming data, introduced in [14]. There, the authors propose a solution, based on the Boneh-Goh-Nissim's 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 [6] how to apply their cryptographic scheme for evaluating a 2-DNF formula. Our schemes could in principle be used to privately 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 [6], 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 [15, 16] or extended private information retrieval [17]. Finally, the work [18] 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.

6 Specific Realizations

We will divide homomorphic encryption schemes in two families: factorization and discrete logarithm based homomorphic encryption schemes, on the one hand; and lattice-based homomorphic encryption schemes, on the other hand.

6.1 Factorization and Discrete Logarithm Based Schemes

In most of these schemes the plaintext space is $(Z_N, +)$ for some positive integer N. They are thus plaintext additive. The ciphertext space $(\mathcal{C}, \otimes)$ is usually $(Z_N, \times)^*$ or $((Z_N, \times)^*)^2$ for some N such that the discrete logarithm in these ciphertext spaces is hard to compute. The existence of a computable pseudo-homomorphism ψ between one of these spaces and $(Z_s, +)^n$, for some s and n, is not possible as computing the discrete logarithm of g^x in $(\mathcal{C}, \otimes)$ can be done by computing $\psi(g^x)/\psi(g)$ in $(Z_s, +)^n$. Thus, the factorization and discrete logarithm schemes cannot be ciphertext and plaintext additive. Therefore, they cannot be used in the first positions of our t-chained schemes, but they can be used as the last scheme of the chain.

Many cryptosystems could be used but Paillier's [8] and Boneh, Goh and Nissim's [6] are specially well fitted for instantiation of t-chained schemes. Paillier's encryption function is an epimorphism $\mathcal{E}_{\mathbf{Paillier}} : (\mathbb{Z}_N, +) \rightarrow (\mathbb{Z}_{N^2}, \times)^*$ and thus the scheme is ∞-pseudo-homomorphic. Boneh, Goh and Nissim's encryption function is also an epimorphism $\mathcal{E}_{\mathbf{BGN}} : (\mathbb{Z}_N, +) \rightarrow (\mathbb{G}, \times)$, $\mathbb{G}$ being a bilinear group of order N associated to a bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$. Both schemes ensure IND-CPA security as long as N is a hard-to-factor modulus, which implies that $N > 2^{1024}$ for current security standards.

Boneh, Goh and Nissim's encryption scheme has a very interesting property. Two encrypted messages can be multiplied once using the bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_1$. The resulting ciphertexts are not in the same group than the original ones and the decryption function for the product, $\mathcal{D}'_{\mathbf{BGN}} : (\mathbb{G}_1, \times) \rightarrow (\mathbb{Z}_N, +)$, is slightly different for these ciphertexts. But the 'new' scheme remains homomorphic allowing to do sums of the plaintexts associated to the ciphertexts in $\mathbb{G}_1$. The encryption scheme of Boneh, Goh and Nissim has a major drawback, though. The decryption of $\mathcal{E}_{\mathbf{BGN}}(x)$ needs $O(\sqrt{x})$ group operations in $\mathbb{G}$ (or $\mathbb{G}_1$). In order to be able to decrypt, plaintext size must be moderate.

6.2 Lattice-Based Schemes

The lattice-based homomorphic schemes proposed by Goldreich, Goldwasser and Halevi [10], Regev [11, 12] and Ajtai [13] have very limited pseudo-homomorphic properties. In [9], Kawachi, Tanaka and Xagawa propose multi-bit variants of these schemes with extended pseudo-homomorphic properties. In [19], Peikert, Vaikuntanathan and Waters also propose a multi-bit variant of the scheme proposed by Regev in [12], but they do not evaluate the pseudo-homomorphic properties of their variant. Thus, even if this scheme is very interesting from a performance point of view, we will not use it for our constructions, letting the proofs and application of this scheme in t-chained schemes as future work.

Among the schemes proposed by Kawachi, Tanaka and Xagawa, two of them allow to form t-chained schemes with acceptable parameters. The most efficient one is the variant of Ajtai [13], whose security is based on an average-case reduction to the Diophantine Approximation problem (DA). The less efficient one is the variant of [11] (noted hereafter mR04), whose security is based on a worse-case reduction to $\tilde{O}(n^{1.5+r}) - uSVP$ for a given parameter r. Even if this scheme provides worse performance results than the variant of Ajtai, we will use it as an example as it allows us to prove that t-chained schemes, even if costly, are possible for the lattice schemes whose security is based on the weakest assumptions.

In mR04, the encryption and decryption functions form an L-pseudo-homomorphism $(\mathcal{E}, \mathcal{D})$ 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 underlying security problem, $\tilde{O}(n^{1.5+r}) - uSVP$, has been solved for $n \simeq 10$ and $r \simeq 10$ but is believed to be hard for $n \simeq 100$ [20]. Key size is in $O(n^4)$, and thus the scheme is considered to be unpractical for such parameters, as it is possible to use other cryptosystems with much smaller keys. However, in our case, we do not want to use this cryptosystem as a traditional public key schemes. In many situations key size is not such an issue as the keys do not need to be sent and are short-lived. Indeed, a user can store a one gigabyte key for the few seconds needed by the protocol and then free its memory by erasing the key. 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.

6.3 Examples of t-Chained Schemes

A 3-Chained Encryption Scheme Using Paillier. Constructing a 2-chained encryption scheme is of little interest as Boneh, Goh and Nissim already provide an efficient solution for the multiplications of 2 operands in [6]. We thus propose a 3-chained 10^4-pseudo-homomorphic encryption scheme. Such a scheme can be used for example, citing one of the previously described applications, to evaluate 3-DNF formulas with up to 10^4 disjunctions.

An instance of $mR04$ for parameters $n = 100$ and $r = 5$ with $p = 10^4 + 1$ and $L = 10^4$ verifies $p \times L < n^r$ and provides a 10^4-pseudo-homomorphic encryption scheme with plaintext space $(\mathbb{Z}_{10^4+1}, +)$ and ciphertext space $(\mathbb{Z}_{2^{80000}}, +)$. We apply Lemma 1 to this instance, with $k = 1$, to obtain an L-pseudo-homomorphic encryption scheme PKE_1 with plaintext space $(\mathbb{Z}_{10^4+1}, +)$ and ciphertext space $(\mathbb{Z}_{10^4+1}, +)^{80000}$, such that $\| \mathcal{E}_1 \|_\infty = 1$.

We define $PKE_{2\alpha}$ applying Lemma 1 with $k = 1000$ to the same initial scheme and obtain thus a 10^4-pseudo-homomorphic encryption scheme $PKE_{2\alpha}$ with plaintext space $(\mathbb{Z}_{10^4+1}, +)$ and ciphertext space $(\mathbb{Z}_{2^{1000}}, +)^{80}$. Using the pseudo-homomorphism from $(\mathbb{Z}_{2^{1000}}, +)$ to $(\mathbb{Z}_{2^{N_P}}, +)$ described in Lemma 2, N_P being a hard-to-factor 1024-bit modulus, we obtain a 10^4-pseudo-homomorphic encryption scheme PKE_2 with plaintext space $(\mathbb{Z}_{10^4+1}, +)$ and ciphertext space $(\mathbb{Z}_{N_P}, +)^{80}$.

Finally, we define PKE_3 as an instance of Paillier's encryption scheme for the hard-to-factor modulus N_P. We will use Lemma 3 to transform it into a ∞-pseudo-homomorphic encryption scheme with the appropriate plaintext space (by increasing the dimension of this space).

Applying twice the construction given in Definition 3 we obtain a 3-chained 10^4-pseudo-homomorphic encryption scheme PKE with plaintext space $(\mathbb{Z}_{10^4+1}, +)$ and ciphertext space $((\mathbb{Z}_{N_P^2}, \times)^*)^{6400000}$. Note that the size of ciphertexts is $2048 \times 6400000 \simeq 13 \times 10^9$ bits, which is very large (maybe too large to be acceptable in many practical applications). However, sending such amounts of data with nowadays bandwidths is possible.

A 2-Chained Encryption Scheme Using Boneh-Goh-Nissim. We can achieve the same functionality as in the previous example (e.g. private evaluation of multi-variate polynomials with degree up to 3) by considering a 2-chained scheme which uses as PKE_2 the scheme of Boneh-Goh-Nissim.

As before, to construct PKE_1, we start from $mR04$ with parameters $n = 100$, $r = 5$, $p = 10^4 + 1$ and $L = 10^4$. We apply Lemma 1 with $k = 1$, to obtain a 10^4-pseudo-homomorphic encryption scheme $PKE_{1\alpha}$ with plaintext space $(\mathbb{Z}_{10^4+1}, +)$ and ciphertext space $(\mathbb{Z}_{10^4+1}, +)^{80000}$. Using the pseudo-homomorphism from $(\mathbb{Z}_{10^4+1}, +)$ to $(\mathbb{Z}_{N_P}, +)$ described in Lemma 2, being N_P a hard-to-factor 1024-bit modulus, we obtain PKE_1, a

10^4-pseudo-homomorphic encryption scheme with plaintext space $(\mathbb{Z}_{10^4+1}, +)$ and ciphertext space $(\mathbb{Z}_{N_P}, +)^{80000}$ with $\| \mathcal{E}_1 \|_\infty = 1$. Let PKE_2 be an instance of Boneh-Goh-Nissim's encryption scheme for the 1024-bit modulus N_P. PKE_2 is an ∞-pseudo-homomorphic encryption scheme with plaintext space $(\mathbb{Z}_{N_P}, +)$ and ciphertext space $(\mathbb{G}, \times)$.

Applying twice the construction given in Definition 3, we obtain a 2-chained 10^4-pseudo-homomorphic encryption scheme PKE with plaintext space $(\mathbb{Z}_{10^4+1}, +)$ and ciphertext space $(\mathbb{G}, \times)^{80000}$. Ciphertexts will therefore be a few megabytes long.

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 crypto-computations.

7 Conclusions

In this work we have presented a theoretical cryptographic object, that we denote as t-chained encryption schemes, which allow to compute over encrypted data in such a way that ciphertexts of sums and t multiplications of the encrypted initial inputs can be computed. This has a huge number of applications in protocols where some users want to evaluate a (possibly secret) function on their private inputs. For example, the work [6], which proposes a solution to this problem for the case $t = 2$, is very celebrated and cited because of its multiple applications.

Our solution theoretically works for any value of t. As an illustrative example, we have explained how to construct practical 3-chained encryption schemes starting from some well-known (pseudo-)homomorphic schemes, some of them necessarily involving lattices.

References

1. Chor, B., Kushilevitz, E., Goldreich, O., Sudan, M.: Private information retrieval. J. ACM **45**(6) (1998) 965–981
2. Yao, A.C.: How to generate and exchange secrets (extended abstract). In: 27th Annual Symposium on Foundations of Computer Science, Toronto, Ontario, Canada, IEEE (1986) 162–167
3. 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
4. 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
5. 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
6. 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
7. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory **31**(4) (1985) 469–472
8. 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

9. 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
10. 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
11. 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
12. 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
13. 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
14. Ostrovsky, R., III, W.E.S.: Private searching on streaming data. J. Cryptology **20**(4) (2007) 397–430
15. Adida, B., Wikström, D.: How to shuffle in public. In: TCC'07. Volume 4392 of Lecture Notes in Computer Science., Springer (2007) 555–574
16. Adida, B., Wikström, D.: Offline/online mixing. In: ICALP'07. Volume 4596 of Lecture Notes in Computer Science., Springer (2007) 484–495
17. 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
18. 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
19. Peikert, C., Vaikuntanathan, V., Waters, B.: A framework for efficient and composable oblivious transfer. In Wagner, D., ed.: Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings. Volume 5157 of Lecture Notes in Computer Science., Springer (2008) 554–571
20. 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