Additively Homomorphic Encryption with d-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] Dept. Matemàtica Aplicada IV,
Universitat Politècnica de Catalunya,
C/ Jordi Girona, 1-3, 08034 Barcelona, Spain
jherranz@ma4.upc.edu

Abstract. The search for encryption schemes that allow to evaluate functions (or circuits) over encrypted data has attracted a lot of attention since the seminal work on this subject by Rivest, Adleman and Dertouzos in 1978.
In this work we define a theoretical object, chained encryption schemes, which allow an efficient evaluation of polynomials of degree d over encrypted data. Chained encryption schemes are generically constructed by concatenating cryptosystems with the appropriate homomorphic properties; such schemes are common in lattice-based cryptography. As a particular instantiation we propose a chained encryption scheme whose IND-CPA security is based on a worst-case/average-case reduction from uSVP.

Keywords: homomorphic encryption, secure function evaluation, lattices.

1 Introduction

Secure function evaluation (SFE) is 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. In (a simplified version of) SFE, a user Alice has a function f and a user Bob has some data x. Depending on the setting, one of the two users, or both of them, must obtain $f(x)$ without learning each other's input.

One solution for this problem uses the concept of *garbled circuit*, introduced by Yao in [36]. Alice receives from Bob a garbled version of x, and sends back a garbled version of f as well as some cryptographic material allowing Bob to evaluate this function on x. After the end of the protocol, Bob learns $f(x)$ and nothing else about f, and Alice learns nothing about x. This solution is based on the usage of encrypted truth tables for the garbled function and oblivious transfer for the garbled data. The main drawback is that the size of the evaluated

ciphertext is at least linear in $|f|$. Alternative solutions were therefore proposed, following a different paradigm (denoted as *computing over encrypted data*), to get size sublinear in $|f|$. Here Bob sends to Alice some information related to x (e.g. an encryption of x), Alice combines f and the data received from Bob, and sends a reply to Bob. From this reply, Bob is able to learn $f(x)$ and Alice learns nothing about x (not even $f(x)$). If moreover, Bob does not learn anything about f (besides $f(x)$) we say the protocol provides function privacy.

The garbled circuit approach provides generic protocols that work for virtually any function f, which may not be the case in the computing over encrypted data setting. On the other hand, the computing over encrypted data setting (on which this paper is focused) can lead to protocols with a much lower communication cost. Indeed, in the garbled circuit approach the communication includes an encrypted description of f and an encrypted description of x. In the computing over encrypted data setting only an encrypted description of x is sent and the reply sent to Bob by Alice can be very compact, perhaps independent of the size of f. More precisely, we will say that a secure evaluation is *efficient* for a family of functions $\mathcal{F}$ if for $f \in \mathcal{F}$ the size of the information exchanged by Alice and Bob is at most sublinear in the function size (and thus less than the size of the information exchanged in the garbled circuit approach).

A family of functions that are specially interesting is the one of multivariate polynomials with m monomials and degree d; that is $P(X_1, \ldots, X_v) = \sum_{\ell=1}^{m} P_\ell(X_1, \ldots, X_v)$, where $P_\ell(X_1, \ldots, X_v)$ are monomials of degree at most d. Many applications such as private information retrieval [21], or private searching on streaming data [25] are based on the secure evaluation of low-degree multivariate polynomials with a large number of monomials (varying in real world scenarios from thousands to billions and above). The only approach to obtain efficient (and secure) evaluations of multivariate polynomials has been until now the usage of homomorphic encryption schemes.

In order to provide such evaluations for degree d polynomials, these encryption schemes must allow to compute products of d plaintexts over encrypted data (possibly with a large expansion factor), and to sum a very large number m of these encrypted products with a small expansion factor (sublinear or logarithmic in m). In this paper we propose a generic construction to obtain such properties and we instantiate this construction with a well-known lattice-based cryptosystem. The security of this particular instance is based on a worst-case/average-case reduction from uSVP (see [24] for more details on hard problems related to lattices), which has been proved as hard as other standard problems like GapSVP or the Bounded Distance Decoding (BDD) problem in [22]. Other instantiations can be found in [15] and [2], using respectively a cryptosystem with security based in the worst-case hardness of LWE, and a cryptosytem with security based in the average-case hardness of particular instances of BDD [1].

Related Work. Since the introduction of the concept of homomorphic encryption, by Rivest, Adleman and Dertouzos in [30], many schemes with homomorphic properties have been proposed. Most of them allow only to compute

over encrypted data one of the operations, either the product (RSA [31], El Gamal [11]) or the sum of the plaintexts (Goldwasser-Micali [17] modulo 2 and Paillier [26] modulo a hard-to-factor composite integer).

These schemes lead to an efficient evaluation of multivariate monomials of any degree or multivariate polynomials of degree 1, but obtaining a scheme that provides efficient evaluation of multivariate polynomials of arbitrary degree is a much more complex problem. In order to evaluate a larger span of functions, some protocols have tried to use the homomorphic encryption schemes that allow to compute just one operation (sum or product) in a less direct way than just using the provided plaintext-ciphertext map. In particular, Sander, Young and Yung proposed a solution in [32], which allows to evaluate any constant fan-in boolean circuit in NC^1. The major drawback of their approach is that communication complexity is exponential in the depth of the circuit, which restricts their protocol to circuits of logarithmic depth. Ishai and Paskin show in [19] how to evaluate any branching program P through ciphertexts whose size depends polynomially on the length of P. Such branching programs include, by a result of Barrington [4], the circuits in NC^1. Unfortunately, in order to evaluate a multivariate polynomial with m monomials, we need an NC^1 circuit of depth in $O(\log m)$ or a branching program of length in $O(m)$ (see [23]). Thus, neither of these protocols are able to provide efficient evaluation of polynomials.

Finding an encryption scheme allowing an efficient direct computation over encrypted data of degree d multivariate polynomials for $d > 1$ has been an open issue for a long term. The first attempts that tried to provide a *fully homomorphic* encryption scheme (i.e. a scheme allowing to compute over encrypted data *both* sums and multiplications arbitrarily), failed to resist to the research community attacks: Fellows and Koblitz proposed Polly Cracker [12] which was broken in [34], Grigoriev and Ponomarenko proposed another public-key scheme [18] which was broken in [7]. For the case of symmetric cryptography, Domingo-Ferrer proposed two schemes [9, 10] which were broken in [6, 35]. Fortunately, as we already noted, in order to have efficient evaluations of degree d multivariate polynomials we just need the encryption scheme to compute products of d plaintexts over encrypted data (possibly with a large expansion factor), and to sum a very large number m of these encrypted products with a small expansion factor (sublinear or logarithmic in m). We will say that such a scheme is *d-multiplicative fully homomorphic*. If for any d a d-multiplicative fully homomorphic instance of the scheme can be produced (possibly with an exponential cost in d), we will say that the scheme is *constant-bounded fully homomorphic*. If moreover the computational costs of the different functions of the encryption scheme are at most polynomial in d, we will say it is *leveled fully homomorphic*.

Finding a non-trivial (i.e. for $d > 1$) d-multiplicative fully homomorphic encryption scheme has also been a long standing open problem. The first step forward was given in 2005, by Boneh, Goh and Nissim, who proposed [5] the first efficient 2-multiplicative fully homomorphic encryption scheme. Their scheme allows the SFE of polynomials of degree $d = 2$, as long as the output $P(a_1, \ldots, a_t)$ is a small number (the computational cost of decryption is polynomial on this

number). The size of the ciphertexts is independent of the number m of monomials in the polynomial, and the secure function evaluation protocol provides function privacy. In 2008, the authors proposed, in a preliminary version of this paper [2], a way to obtain efficient constant-bounded fully homomorphic encryption schemes (without function privacy).

In STOC 2009, Gentry proposed an elegant solution [13] for the (efficient) leveled fully homomorphic encryption problem, in two steps. First, he proposed an efficient constant-bounded fully homomorphic encryption scheme based on the hardness of a new problem, the Ideal Coset Problem, which is close to a decisional Closest Vector Problem (which is in turn an instance of the Bounded Distance Decoding problem, see [24]). Second, he proposed an efficient leveled fully homomorphic variant of this scheme, based on the Ideal Coset Problem and a second new problem, the SplitKey Distinguishing Problem which seems to be related to the Sparse Subset Sum Problem (in fact the scheme can be modified to be fully homomorphic if circular security is assumed, see [13] for details). In lattice-based encryption schemes the randomness distribution usually evolves as homomorphic operations are done until the ciphertext becomes impossible to decrypt, which places a limit on the number of operations that can be done. The groundbreaking idea of Gentry is the proposal of a scheme that can "refresh" this randomness to its initial state (more exactly close to the initial state), by the homomorphic evaluation of its own decryption circuit, without revealing the plaintext. In his PhD dissertation [14], Gentry recently presented a *quantum* reduction from the security of his leveled fully homomorphic scheme to the worst case of the Shortest Independent Vector Problem (SIVP, see [24]) on *ideal lattices in a given ring R*. Improvements and variations of Gentry's schemes have appeared very recently [33, 8].

Finally, in [15], Gentry, Halevi and Vaikuntanathan have proposed a new efficient 2-multiplicative encryption scheme (GHV for short) which improves the proposal of Boneh, Goh, and Nissim in various ways. First, it is based on a worst-case/average-case *classical* reduction from LWE (again, see [24]). Moreover, it does not have restrictions in the size of the output. And finally, it can also be used with our construction to obtain a constant-bounded homomorphic encryption scheme.

Our Contribution. This paper is a major write up of [2]. With respect to the related work as a whole, our main contribution is to provide a *generic* construction of efficient constant-bounded fully homomorphic encryption schemes. This construction can be instantiated using different encryption schemes as a base. In particular, the encryption schemes of the fruitful field of lattice-based cryptography seem specially well adapted, but other fields such as code-based cryptography are promising too. The recent instantiation with GHV (proposed in [15]), highlights the generic aspect of our contribution.

With respect to the proposal of Gentry [13], the main contribution comes from the fact that the construction relies on the same security assumptions as encryption schemes with simple-to-achieve homomorphic properties. Thus, we

benefit from the strong reductions available in simple lattice-based encryption schemes, instead of the assumptions needed to get (somewhat) fully homomorphic encryption schemes. In particular, this leads to assumptions on the *classical* (by opposition to *quantum*) worst-case hardness of standard problems, which moreover are done over pretty general lattices, namely integer lattices, instead of ideal lattices over a given ring.

Finally, with respect to the generic proposals of Yao [36] and Sander et al. [32], considering the secure evaluation of polynomials, the main advantage comes from the bandwidth efficiency for low-degree polynomials with a large number of monomials. Indeed, our construction can be used for the secure evaluation of a polynomial with v variables, M monomials and degree d. In Table 1, we compare the bandwidth required by these generic proposals with different instantiations of our construction. The number of monomials is supposed to be bounded by a polynomial in the security parameter $\kappa^{r/2}$ for a given r, and $\mathrm{poly}(\kappa)$ generically denotes a polynomial function of the security parameter κ.

Approach	**Required Bandwidth**
Yao's garbled circuits [36]	$M \cdot d \cdot \mathrm{poly}(\kappa)$
Sander-Young-Yung [32]	$(M \cdot d)^2 \cdot \mathrm{poly}(\kappa)$
$\tilde{O}(\kappa^{1.5+r})$-uSVP instantiation	$\tilde{O}(M^{4d/r}) \cdot \mathrm{poly}(\kappa)^d$
$\tilde{O}(\kappa^{3.5+3r})$-LWE instantiation [15]	$\mathrm{poly}(\log M)^d \cdot \mathrm{poly}(\kappa)^d$
Optimal bound (for our construction)	$\log M \cdot \mathrm{poly}(\kappa)^d$

Table 1. Comparison of the bandwidth requirements of different solutions.

Note that in the uSVP instantiation r can be chosen arbitrarily large to reduce the bandwidth usage, but at the cost of a stronger security assumption. In the LWE instantiation, this is pointless as bandwidth usage does not depend on r and it is enough to set r such that $\kappa^{r/2} = O(M)$.

Last row of Table 1 would be ideally achieved by combining our construction with an additively homomorphic encryption scheme, supporting M additions, where the expansion factor between plaintexts and ciphertexts does not depend on M. The scheme by Gentry [13] could be a candidate for such an encryption scheme but, to the best of the authors knowledge, there is no such scheme with a classical reduction (nor based on integer lattices).

For a given degree d and a growing number of monomials our construction beats asymptotically the other approaches. For variable d, the comparison depends on whether the polynomials are sparse or not, and on the number v of variables. If the polynomials are very sparse, our solution will not be efficient. On the other hand if the polynomials are dense (i.e. we have $M \simeq v^d$), our construction will beat the other approaches if and only if the number of variables is

larger than the number of bits in a ciphertext of the encryption scheme used to instantiate our construction. Classical applications of secure polynomial evaluation, such as private information retrieval [21] or private searching on streaming data [25], result in dense polynomials with a large number of variables and should thus benefit from this construction.

As opposed to the other solutions in this table, the construction that we introduce in this work does not provide function privacy. An alternative construction providing such a property is possible, but due to space restrictions it is left to the long version of this paper. Some details about this alternative construction are provided in Section 5.2.

2 Basic Idea

Let $\mathsf{PKC} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme such that the addition of ciphertexts over the integers maps to an addition on the plaintext space and such that 0 decrypts to 0. Let $a, b \in \{0,1\}$ and $\alpha \in \mathsf{Enc}(pk, a), \beta \in \mathsf{Enc}(pk, b)$, with $(\alpha^{(1)}, \ldots, \alpha^{(t)})$ the bit-representation of α. We define the reconstruction function $R((\alpha^{(1)}, \ldots, \alpha^{(t)})) = \sum_i 2^{i-1}\alpha^{(i)} = \alpha$.

The basic idea is to build the compound ciphertext $\alpha \otimes \beta \stackrel{\mathsf{def}}{=} (\alpha^{(1)}\beta, \ldots, \alpha^{(t)}\beta)$ which encrypts redundantly a and b. Consider the following decryption algorithm: first, decrypt each coordinate with Dec; then reconstruct the inner ciphertext with R, and decrypt it again with Dec.

What is interesting is that each coordinate of the compound ciphertext is either 0 (which decrypts to 0) or β. If $b = 0$, all the coordinates will decrypt to 0 and the resulting null-vector will also decrypt to $ab = 0$ (as $b = 0$) whatever the value of a is. On the other hand, if $b = 1$ all the coordinates in which we have β will decrypt to 1, and we will thus get back $(\alpha^{(1)}, \ldots, \alpha^{(t)})$ which decrypts to $ab = a$ (as $b = 1$).

Toy Example

$\alpha = 110 \quad \beta = 101 \quad \gamma = \alpha \otimes \beta = (101, 101, 000)$
If $\beta \in \mathsf{Enc}(pk, 0)$

$$\gamma \stackrel{\text{1st decryption}}{\longrightarrow} (0,0,0) \stackrel{\text{reconstruction}}{\longrightarrow} 000 \stackrel{\text{2nd decryption}}{\longrightarrow} 0$$

If $\beta \in \mathsf{Enc}(pk, 1)$

$$\gamma \stackrel{\text{1st decryption}}{\longrightarrow} (1,1,0) \stackrel{\text{reconstruction}}{\longrightarrow} 110 \stackrel{\text{2nd decryption}}{\longrightarrow} a$$

Thus, these compound ciphertexts encrypt a product, but not very efficiently (the data is redundant). However, as PKC provides an homomorphic operation, we can add many of these compound ciphertexts, and the result will decrypt to

the sum of the products. This allows us to evaluate degree 2 polynomials over encrypted data using a single vector of t coordinates, which will save bandwidth if the number of added monomials is over t.

Finally, we can note that this construction can be easily generalized if α is a vector of integers (we split each integer in bits and reconstruct them separately on the decryption phase), which allows us to iterate the construction and evaluate polynomials of degree d, at the price of an expansion factor for the length of the ciphertexts which is exponential in d.

3 Chaining Encryption Schemes

3.1 (n, t)-Chainable Schemes

In this subsection we provide a definition of chainable encryption schemes. This definition allows us to present the properties needed to chain schemes (or to compute with them) as well as to have a short naming convention that highlights a given scheme's performance parameters. For integer values $a < b$, we denote as $[a, b]$ the set $\{a, a+1, \ldots, b-1, b\}$.

Definition 1. *A scheme* $\mathsf{PKC} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is said* (n, t)*-chainable if the key generation algorithm* KeyGen *takes as input a security parameter* κ *and a positive integer* m*, and for any value of these parameters, there are two positive integers* n, t *(which may be functions of* κ *and* m*), such that for any keypair* $(pk, sk) \in \mathsf{KeyGen}(1^\kappa, m)$ *the following holds:*

- *The plaintext space* $\mathcal{P}$ *is a subset of* $\mathbb{Z}$*, and includes* $[0, m]$*;*
- *The ciphertext space* $\mathcal{C}$ *is a subset of* $\mathbb{Z}^n$ *and includes* 0^n*, moreover* 0^n *is in the support of the output of* $\mathsf{Enc}(pk, 0)$*;*
- *Bounded size: for any plaintext* $x \in \mathcal{P}$ *and any ciphertext* $c \in \mathsf{Enc}(pk, x)$*, all the entries of* c *are smaller than* 2^t *(i.e.,* $\mathsf{Enc}(pk, x) \subset [0, 2^t - 1]^n$*);*
- m*-limited homomorphism via integer addition: for any* $\ell \leq m$*,* $a_1, \ldots, a_\ell \in \{0, 1\}$ *and any* $c_1, \ldots, c_\ell$ *with* $c_i \in \mathsf{Enc}(pk, a_i)$*, the integer vector* $c = \sum_i c_i$ *is decrypted via* Dec *to the integer* $a = \sum_i a_i$ *(which is in* $[0, m]$*).*

Lattice-based schemes with homomorphic properties are usually suitable (sometimes with a small transformation) for this definition. Note, however, that we do not set any constraint on the ciphertext size $n \times t$ or its relation to m and thus, that not all the schemes that fit into this definition will be able to provide efficient (sub-linear in m) evaluations of polynomials. These issues will be dealt with in Sections 4 and 5.

3.2 Chaining Schemes

In this subsection we present an algorithm that chains two encryption schemes $\mathsf{PKC}_1, \mathsf{PKC}_2$ that are respectively (n_1, t_1)-chainable and (n_2, t_2)-chainable, into a scheme $\mathsf{PKC} = \mathsf{chain}(\mathsf{PKC}_1, \mathsf{PKC}_2)$, that is $(n_2 n_1 t_1, t_2)$-chainable. This scheme

has a worse ciphertext/plaintext expansion ratio than the two chained schemes, but is interesting because given $\alpha \in \mathsf{Enc}_1(pk_1, a_1)$ and $\beta \in \mathsf{Enc}_2(pk_2, a_2)$ we are able to generate an element of $\mathsf{Enc}(pk, a_1 a_2)$ (see Section 4).

Chaining Algorithm: $\mathsf{PKC} = \mathsf{chain}(\mathsf{PKC}_1, \mathsf{PKC}_2)$

Input:
- An (n_1, t_1)-chainable scheme $\mathsf{PKC}_1 = (\mathsf{KeyGen}_1, \mathsf{Enc}_1, \mathsf{Dec}_1)$
- An (n_2, t_2)-chainable scheme $\mathsf{PKC}_2 = (\mathsf{KeyGen}_2, \mathsf{Enc}_2, \mathsf{Dec}_2)$

Output:
- An $(n_1 t_1 n_2, t_2)$-chainable scheme $\mathsf{PKC} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$

Consider the intermediate encryption scheme PKC'_1:

$\mathsf{KeyGen}'_1(1^\kappa, m)$:
1 Return $(pk_1, sk_1) \leftarrow \mathsf{KeyGen}_1(1^\kappa, m)$

$\mathsf{Enc}'_1(pk_1, a)$:
1 Sample $\alpha = (\alpha^{(1)}, \ldots, \alpha^{(n_1)})$ from $\mathsf{Enc}_1(pk_1, a)$
2 Return $\alpha' = (\underbrace{\alpha'^{(1)}, \ldots, \alpha'^{(t_1)}}_{\text{bits of } \alpha^{(1)}}, \ldots, \underbrace{\alpha'^{((n_1-1)t_1+1)}, \ldots, \alpha'^{(n_1 t_1)}}_{\text{bits of } \alpha^{(n_1)}})$

$\mathsf{Dec}'_1(sk_1, \alpha')$:
1 Compute $\alpha = \mathcal{R}_1(\alpha') \stackrel{\mathsf{def}}{=} (\sum_{j=1}^{t_1} 2^{j-1}\alpha'^{(j)}, \ldots, \sum_{j=1}^{t_1} 2^{j-1}\alpha'^{((n_1-1)t_1+j)})$
2 Return $a \leftarrow \mathsf{Dec}_1(sk_1, \alpha)$

Return a description of the final encryption scheme PKC:

$\mathsf{KeyGen}(1^\kappa, m)$:
1 Set $(pk_1, sk_1) \leftarrow \mathsf{KeyGen}_1(1^\kappa, m)$, $(pk_2, sk_2) \leftarrow \mathsf{KeyGen}_2(1^\kappa, m)$
2 Return $((pk_1, pk_2), (sk_1, sk_2))$

$\mathsf{Enc}((pk_1, pk_2), a)$:
1 Set $\alpha' = (\alpha'^{(1)}, \ldots, \alpha'^{(n_1 t_1)}) \leftarrow \mathsf{Enc}'_1(pk_1, a)$
2 For each $j \in [1, n_1 t_1]$ set $\beta_j \leftarrow \mathsf{Enc}_2(pk_2, \alpha'^{(j)})$
3 Return $\gamma = (\beta_1, \ldots, \beta_{n_1 t_1})$

$\mathsf{Dec}((sk_1, sk_2), \gamma)$:
1 For each $j \in [1, n_1 t_1]$, set $\alpha'^{(j)} \leftarrow \mathsf{Dec}_2(sk_2, \gamma^{(j)})$
2 Return $a \leftarrow \mathsf{Dec}'_1(sk_1, (\alpha'^{(1)}, \ldots, \alpha'^{(n_1 t_1)}))$

Proposition 1. PKC *is* $(n_1 t_1 n_2, t_2)$*-chainable. Moreover, if the instance of* PKC_1 *associated to* (pk_1, sk_1) *and the instance of* PKC_2 *associated to* (pk_2, sk_2) *are* m*-limited homomorphisms, the instance of* PKC *associated to* $((pk_1, pk_2), (sk_1, sk_2))$ *is also an* m*-limited homomorphism.*

Proof. Clearly, $\mathcal{R}_1$ is linear and therefore the instance of PKC'_1 associated to (pk_1, sk_1) is an m-limited homomorphism via integer addition, as the instance of PKC_1 associated to the same keypair. For $i \in [1, m]$, let $a_i \in \{0, 1\}$ and $\gamma_i \leftarrow \mathsf{Enc}((pk_1, pk_2), a_i)$. We have

$$\sum_{i=1}^{m} \gamma_i = \left(\sum_{i=1}^{m} \beta_{i,1}, \ldots, \sum_{i=1}^{m} \beta_{i,n_1 t_1} \right)$$

with $\beta_{i,j} \leftarrow \mathsf{Enc}_2(pk_2, \alpha_i'^{(j)})$ and $\alpha'_i \leftarrow \mathsf{Enc}'_1(pk_1, a_i)$. Since the used instance of PKC_2 is an m-limited homomorphism via integer addition and each $\alpha_i'^{(j)}$ is in $\{0, 1\}$, applying Dec_2 to each coordinate, with secret key sk_2, we obtain

$$\left(\sum_{i=1}^{m} \alpha_i'^{(1)}, \ldots, \sum_{i=1}^{m} \alpha_i'^{(n_1 t_1)} \right) = \sum_{i=1}^{m} \alpha'_i$$

As the instance of PKC'_1 associated to (pk_1, sk_1) is also an m-limited homomorphism via integer addition, decrypting this vector with Dec'_1 and the secret key sk_1 we obtain $\sum_{i=1}^{m} a_i$, and thus the instance of PKC associated to $((pk_1, pk_2), (sk_1, sk_2))$ is an m-limited homomorphism via integer addition.

Finally, as $(0, \ldots, 0) \in \mathsf{Enc}((pk_1, pk_2), 0)$, and the ciphertexts are clearly vectors of $n_1 \cdot t_1 \cdot n_2$ scalars of t_2 bits each, we therefore have that PKC is $(n_1 t_1 n_2, t_2)$-chainable. □

Let us prove now that the chained scheme PKC resulting from PKC_1 and PKC_2 is IND-CPA secure if either of PKC_1, PKC_2 is IND-CPA secure. We recall first the standard notion of indistinguishability under chosen-plaintext attacks (IND-CPA security), for an encryption scheme $\mathsf{PKC} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$. We use the following game that an attacker $\mathcal{A}$ plays against a challenger:

$$\begin{array}{l} (pk, sk) \leftarrow \mathsf{KeyGen}(1^\kappa) \\ (St, a_0, a_1) \leftarrow \mathcal{A}(\text{find}, pk) \\ b \leftarrow \{0, 1\} \text{ at random} \\ c^* \leftarrow \mathsf{Enc}(pk, a_b) \\ b' \leftarrow \mathcal{A}(\text{guess}, c^*, St). \end{array}$$

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 enjoys IND-CPA security if $\mathrm{Adv}(\mathcal{A})$ is a negligible function of the security parameter κ, for any attacker $\mathcal{A}$ running in polynomial time (in κ).

Proposition 2 (IND-CPA Security). $\mathsf{PKC} = \mathsf{chain}(\mathsf{PKC}_1, \mathsf{PKC}_2)$ *is IND-CPA secure if either of* PKC_1, PKC_2 *is IND-CPA secure.*

Proof (Sketch.). Let us assume that there exists a CPA attacker $\mathcal{A}$ against PKC and let us prove, then, that neither of PKC_1, PKC_2 can be IND-CPA. Specifically, we can construct CPA attackers $\mathcal{A}_1, \mathcal{A}_2$ against the schemes PKC_1 and PKC_2.

For PKC_1, the attacker $\mathcal{A}_1$ is trivial as a random keypair of PKC_1 can be transformed in a random keypair of PKC by adding a random keypair of PKC_2. Moreover, the choice of the two plaintexts by $\mathcal{A}$ is maintained by $\mathcal{A}_1$, and the challenges from Enc_1 can be transformed into challenges following the distribution of Enc by splitting them into bits and encrypting them through Enc_2. Finally, as the plaintexts are the same, Attacker $\mathcal{A}_1$ will output the same guess as $\mathcal{A}$ will, and the success probability of both attackers will be exactly the same.

For PKC_2, the idea is similar, but we proceed in two steps. First, we define an attacker $\mathcal{A}'_2$ able to distinguish between the distributions associated to $n_1 t_1$ plaintexts. Namely, if $\mathcal{A}$ chooses two plaintexts a_0, a_1, $\mathcal{A}'_2$ chooses two sets of plaintexts $(\alpha_0^{(1)}, \ldots, \alpha_0^{(n_1 t_1)})$, $(\alpha_1^{(1)}, \ldots, \alpha_1^{(n_1 t_1)})$, for $\alpha_0 \leftarrow \mathsf{Enc}_1(pk_1, a_0)$ and $\alpha_1 \leftarrow \mathsf{Enc}_1(pk_1, a_1)$ which ensures that $\mathcal{A}$, and therefore $\mathcal{A}'_2$, is able to distinguish the challenges with an non-negligible advantage. Then, we use a standard hybrid argument to derive from $\mathcal{A}'_2$ an attacker $\mathcal{A}_2$ against PKC_2.

□

The output of the chaining algorithm being itself chainable we can iteratively construct a chain of d encryption schemes $\mathsf{PKC}_1, \ldots, \mathsf{PKC}_d$, if for any $i \in [1, d]$ PKC_i is (n_i, t_i)-chainable, and obtain an $(n_d \prod_{j=1}^{d-1} n_j t_j, t_d)$-chainable encryption scheme PKC, with $(pk_1, \ldots, pk_d)$ and $(sk_1, \ldots, sk_d)$ as public and secret keys. Note that $\mathsf{PKC}_1, \ldots, \mathsf{PKC}_d$ need not to be different schemes and that we can chain d times an (n, t)-chainable scheme PKC to itself. In this case we get an $(n(nt)^{d-1}, t)$-chainable scheme, which has the same public/secret keypair (pk, sk) as PKC.

4 Computing with Chained Schemes

4.1 Product and Polynomial Evaluation

Chained schemes being themselves chainable they provide a limited homomorphism via integer addition (by Definition 1). Thus, in order to compute sums of plaintexts over encrypted data with them we just need to add up the corresponding ciphertexts. Computing products of plaintexts over encrypted data is not as straightforward and requires to use ciphertexts of the multiplication operands under the encryption schemes that form the chain. The following algorithm shows how to proceed.

Product Computation Algorithm: $\gamma = \mathsf{product}(\alpha, \beta)$

Input:
- $\alpha \in \mathsf{Enc}_1(pk_1, a_1)$ for $a_1 \in \{0, 1\}$ and PKC_1 (n_1, t_1)-chainable
- $\beta \in \mathsf{Enc}_2(pk_2, a_2)$ for $a_2 \in \{0, 1\}$ and PKC_2 (n_2, t_2)-chainable

Output:
- $\gamma \in \mathsf{Enc}((pk_1, pk_2), a_1 a_2)$ for $\mathsf{PKC} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}) = \mathsf{chain}(\mathsf{PKC}_1, \mathsf{PKC}_2)$

1 Split α into the bit vector $\alpha' = (\alpha'^{(1)}, \ldots, \alpha'^{(n_1 t_1)}) \in \mathsf{Enc}'_1(pk_1, a_1)$
2 Multiply each one-bit scalar of this vector by β and output the result.

Proposition 3. *The output of the above-described protocol* product *belongs to* $\mathsf{Enc}((pk_1, pk_2), a_1 a_2)$.

Proof. We have $\mathsf{product}(\alpha, \beta) = (\alpha'^{(1)}\beta, \ldots, \alpha'^{(n_1 t_1)}\beta)$. We want to prove that there is $\gamma \in \mathsf{Enc}((pk_1, pk_2), a_1 a_2)$ such that for all $j \in [1, n_1 t_1]$ we have $\alpha'^{(j)}\beta = \gamma^{(j)}$. By the construction of a chained scheme, this is equivalent to: there is $\alpha'_{1,2} \in \mathsf{Enc}'_1(pk_1, a_1 a_2)$ such that $\alpha'^{(j)}\beta \in \mathsf{Enc}_2(pk_2, \alpha'^{(j)}_{1,2})$, for all $j \in [1, n_1 t_1]$.

If $a_2 = 1$ set $\alpha'_{1,2} = \alpha' \in \mathsf{Enc}'_1(pk_1, a_1) = \mathsf{Enc}'_1(pk_1, a_1 a_2)$. For each $j \in [1, n_1 t_1]$

- if $\alpha'^{(j)} = 1$,

$$\alpha'^{(j)}\beta = \beta \in \mathsf{Enc}_2(pk_2, a_2) = \mathsf{Enc}_2(pk_2, \alpha'^{(j)}) \Rightarrow \alpha'^{(j)}\beta \in \mathsf{Enc}_2(pk_2, \alpha'^{(j)}_{1,2})$$

- if $\alpha'^{(j)} = 0$,

$$\alpha'^{(j)}\beta = (0, \ldots, 0) \in \mathsf{Enc}_2(pk_2, 0) = \mathsf{Enc}_2(pk_2, \alpha'^{(j)}) \Rightarrow \alpha'^{(j)}\beta \in \mathsf{Enc}_2(pk_2, \alpha'^{(j)}_{1,2})$$

$\Rightarrow$ if $a_2 = 1$ the output of the algorithm is in $\mathsf{Enc}((pk_1, pk_2), a_1 a_2)$.

If $a_2 = 0$, set $\alpha'_{1,2} = (0, \ldots, 0) \in \mathsf{Enc}'_1(pk_1, a_1 a_2)$. For each $j \in [1, n_1 t_1]$

- if $\alpha'^{(j)} = 1$,

$$\alpha'^{(j)}\beta = \beta \in \mathsf{Enc}_2(pk_2, a_2) = \mathsf{Enc}_2(pk_2, 0) \Rightarrow \alpha'^{(j)}\beta \in \mathsf{Enc}_2(pk_2, \alpha'^{(j)}_{1,2})$$

- if $\alpha'^{(j)} = 0$,

$$\alpha'^{(j)}\beta = (0, \ldots, 0) \in \mathsf{Enc}_2(pk_2, 0) \Rightarrow \alpha'^{(j)}\beta \in \mathsf{Enc}_2(pk_2, \alpha'^{(j)}_{1,2})$$

$\Rightarrow$ if $a_2 = 0$ the output of the algorithm is also in $\mathsf{Enc}((pk_1, pk_2), a_1 a_2)$.

□

This algorithm can be used iteratively to obtain encrypted products of d plaintexts. As these products are ciphertexts of a chained (and thus chainable) encryption scheme, we can add them and the result will decrypt to the evaluation of a degree d binary polynomial (if the homomorphic parameter m of the scheme is larger than the number of monomials M). The following algorithms provide a complete protocol for degree d polynomial evaluation over encrypted data. We want to stress that the algorithms can be easily modified (to get more efficient and simple), in case the input polynomial P has a more compact representation, e.g. $P = (X_1 + 1)^d$.

Polynomial Evaluation Algorithms: $P = \sum_{\ell=1}^{M} X_{\ell_1} \dots X_{\ell_d} \in \mathbb{Z}_2[X_1, \dots, X_v]$[3]

Setup $\mathsf{KeyGen}_P(1^\kappa, M)$:
Input:
- A security parameter 1^κ
- A maximum number of monomials M
Output: A keypair $(pk, sk) \in \mathsf{KeyGen}(1^\kappa, M)$ for $\mathsf{PKC} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ an (n, t)-chainable scheme PKC

Encryption $\mathsf{Enc}_P(pk, (a_1, \dots, a_v))$:
Input:
- A public key pk of the afore-mentioned (n, t)-chainable scheme PKC
- A point $(a_1, \dots, a_v)$ in $\{0,1\}^v$ in which the polynomial should be evaluated
Output: A set of ciphertexts $\alpha_i \in \mathsf{Enc}(pk, a_i)$ for $i \in [1, v]$

1 Set $\alpha_i \leftarrow \mathsf{Enc}(pk, a_i)$ for $i \in [1, v]$
2 Return $\alpha_1, \dots, \alpha_v$

Evaluation $\mathsf{Eval}_P((\alpha_1, \dots, \alpha_v), P)$:
Input:
- An encryption $(\alpha_1, \dots, \alpha_v)$ of a point in $\{0,1\}^v$, through PKC
- The description of a polynomial $P = \sum_{\ell=1}^{M} X_{\ell_1} \dots X_{\ell_d} \in \mathbb{Z}_2[X_1, \dots, X_v]$
Output: A sum of ciphertexts, α, that decrypts to $P(a_1, \dots, a_v)$

1 For $\ell = 1, \dots, M$
2 $\quad \alpha_{\ell,1} \stackrel{\mathsf{def}}{=} \alpha_{\ell_1}$
3 $\quad$ For $j = 2, \dots, d$
4 $\quad\quad \alpha_{\ell,j} = \mathsf{product}(\alpha_{\ell,j-1}, \alpha_{\ell_j})$
5 Return $\alpha = \sum_{\ell=1}^{M} \alpha_{\ell,d}$

Decryption $\mathsf{Dec}_P(sk, \alpha)$
Input:
- A secret key sk of the afore-mentioned (n, t)-chainable scheme PKC
- The output, α, of the evaluation algorithm
Output: $P(a_1, \dots, a_v)$

1 $\mathsf{PKC}_{1,2} = \mathsf{chain}(\mathsf{PKC}, \mathsf{PKC})$
2 For $j = 3, \dots, d$: $\mathsf{PKC}_{1,j} = \mathsf{chain}(\mathsf{PKC}_{1,j-1}, \mathsf{PKC})$
4 Return $(a \bmod 2)$ for $a \leftarrow \mathsf{Dec}_{1,d}(sk, \alpha)$

Note that if the polynomial has a monomial of degree $d' < d$ it is enough to add the following computation: For $j \in [d', d-1]$: $\alpha_{\ell,j+1} = \mathsf{product}(\alpha_{\ell,j}, \alpha_0)$, where $\alpha_0 \in \mathsf{Enc}(pk, 1)$. This step ensures that the protocol processes the polynomial correctly.

[3] Note that we do not use a standard indexing such as $\sum_{\ell=1}^{M} X_{i_{1,\ell}} \dots X_{i_{d,\ell}}$ and rather implicitly associate to each $\ell \in [1, M]$ a tuple $(\ell_1, \dots, \ell_d) \in [1, v]^d$ to reduce index notations.

Proposition 4. *The Polynomial Evaluation Algorithm is correct, produces an output of* $(nt)^d \log m$ *bits and if* PKC *is IND-CPA, the choice of the evaluation point is private.*

Proof (Sketch.). The correctness of the Product Algorithm guarantees that $\alpha_{\ell,j} \in \mathsf{Enc}_{1,j}(pk, a_{\ell_1} \cdots a_{\ell_j})$ for any $j \in [1, d]$ and any $\ell \in [1, M]$ (denoting $\mathsf{PKC}_{1,1} = \mathsf{PKC}$). Indeed, by induction, for $j = 1$ we have $\alpha_{\ell,1} \in \mathsf{Enc}_{1,1}(pk, a_{\ell_1})$. Suppose that we have $\alpha_{\ell,j} \in \mathsf{Enc}_{1,j}(pk, a_{\ell_1} \cdots a_{\ell_j})$. By the product algorithm correctness we know that $\alpha_{\ell,j+1} = \mathsf{product}(\alpha_{\ell,j}, \alpha_{\ell_{j+1}})$ is an encryption of $a_{\ell_1} \cdots a_{\ell_{j+1}}$ using the encryption scheme $\mathsf{chain}(\mathsf{PKC}_{1,j}, \mathsf{PKC}) \stackrel{\mathsf{def}}{=} \mathsf{PKC}_{1,j+1}$. In other words, $\alpha_{\ell,j+1} \in \mathsf{Enc}_{1,j+1}(pk, a_{\ell_1} \cdots a_{\ell_{j+1}})$, which completes the induction proof.

As each monomial computed in the main loop of the evaluation algorithm is a ciphertext of $\mathsf{PKC}_{1,d}$, and $\mathsf{PKC}_{1,d}$ is $(n(nt)^{d-1}, t)$-chainable, the result of the final step has $(nt)^d \log m$ bits. Moreover, as the instance of $\mathsf{PKC}_{1,d}$ associated to (pk, sk) is an M-limited homomorphism the result decrypts $(\bmod 2)$ to $P(a_1, \ldots, a_v)$.

If PKC is IND-CPA, the indistinguishability of the evaluation points is straightforward using a standard hybrid argument. □

In order to have an efficient evaluation of a polynomial through chained schemes, $(nt)^d \log m$ must be sub-linear in m. As nt is the ciphertext size of PKC, we must use an encryption scheme such that ciphertext size grows as $o((m/\log m)^{1/d})$. Such instantiations are presented in the next section.

4.2 Higher Moduli

The definitions, algorithms, and propositions, in this and the previous section need only to be slightly changed in order to produce chained schemes that allow to compute sums, products, and more generally evaluate polynomials, over $\mathbb{Z}_r$ for $r > 2$. Namely,

- In Definition 1, the homomorphic property must hold $\forall a_1, \ldots, a_m \in [0, r-1]$
- In the $\mathsf{product}$ algorithm the output is a sum of up to r ciphertexts;
- In the algorithm Eval_P for the evaluation of a polynomial, we need an extra final step in each monomial computation in which the associated ciphertext is added to itself a given number of times (the coefficient in front of the monomial).

In order to remain correct, the $\mathsf{product}$ algorithm and the polynomial Eval_P algorithm, require respectively $m > r$ and $m > Mr^d$ (M being the number of monomials of the polynomial). The rest of the definitions, algorithms and propositions remain unchanged, but the proofs get harder to read, and we have thus preferred to provide them only in the long version of this paper.

5 Specific Realizations

In this section we describe some encryption schemes that satisfy the conditions given in Definition 1. These schemes are all based on lattices, and can be used at any point of a chain.[4] On the other hand, it is obvious that the last encryption scheme in a chain does not need to have a ciphertext space which is an additive group. Therefore, we can use for the last scheme PKC_d other homomorphic encryption schemes, not necessarily based on lattices, as long as their plaintext spaces are additive groups. This includes schemes like Paillier's [26] or Boneh-Goh-Nissim's [5] (BGN for short). The advantage of using the BGN scheme is that it provides an additional level of multiplications *for free*. That is, if we have a d-chained encryption scheme where PKC_d is the BGN cryptosystem, then we could use the global scheme to evaluate multivariate polynomials of degree up to $d+1$ (as long as the result of the evaluation is relatively small, which is the drawback of BGN). Such a hybrid lattice-based and number-theory encryption scheme allowing an efficient evaluation of degree $d > 2$ polynomials over encrypted data is a surprising consequence of our approach.

5.1 A Scheme Based on uSVP

In [20], Kawachi, Tanaka and Xagawa propose a set of lattice-based encryption schemes, derived from [16, 28, 29, 3], that present some homomorphic properties. In particular, we are interested in the variant of [28], whose IND-CPA security is based on a worst-case/average-case reduction from $\tilde{O}(\kappa^{1.5+r}) - uSVP$ for given security parameters κ, r (related to the underlying lattice). In this scheme, the plaintext space is $(\mathbb{Z}_p, +)$, for an arbitrary parameter p, and the ciphertext space is $(\mathbb{Z}_N, +)$, with $N = 2^{8\kappa^2}$. As it is proved in [20], the scheme is an m-limited additive homomorphism via addition modulo N, when $m \cdot p < \kappa^r$. As $\mathbb{Z}_N$ is a $\mathbb{Z}$-module, adding up the ciphertext as integers, and applying the mod N operation just before the decryption gives the same result, and thus we have an m-limited additive homomorphism via integer addition. Moreover, for any keypair 0 is an encryption of 0 and thus, as long as $m < p$, the scheme is $(1, t)$-chainable, with $t = \log N = 8\kappa^2$.

The output of the secure evaluation of a degree d polynomial with this scheme has a size $t^d \log m = 8^d \kappa^{2d} \log m$. Using $m < p$ and $m \cdot p < \kappa^r$ we get that the output of the evaluation has roughly a size of $8^d m^{4d/r} \log m$ bits, and therefore we must have $r > 4d$ in order to have an efficient evaluation. In terms of security, this implies that this instantiation relies on the worst case hardness of $\tilde{O}(\kappa^{1.5+4d}) - uSVP$.

5.2 Other schemes

As noted in the related work section, GHV [15] is another lattice-based encryption scheme which can be used with our construction. The security of their

[4] Code-based schemes seem also an interesting alternative to be explored.

scheme is based on the worst-case hardness of LWE (for a given approximation factor), which is equivalent to the worst-case hardness of several standard lattice problems (see [27]).

Gentry et al. note that their scheme has the same "multiplication-for-free" property (described at the beginning of this section) as the cryptosystem of Boneh et al. [5], allowing thus the evaluation of degree $d+1$ polynomials with a chain of just d schemes. In fact, it is possible to do much better, as even after the multiplication for free GHV's ciphertexts can undergo m additive operations, and thus it is possible to alternate. First, we do a multiplication for free with their scheme and then a multiplication with our construction. As the result of our multiplication is a set of GHV's ciphertexts, they can again undergo a multiplication for free, and so on. With this improvement it is possible to evaluate polynomials of degree $2d$ with just a chain of d schemes.

A second advantage of GHV is that ciphertext size grows only logarithmically in m (whereas with the uSVP instantiation we present, it grows polynomially). In order to use the full potential of this fact we must change our construction and split the ciphertexts in groups of bits, instead of bits, just as it is presented in [2] for the instantiation of our construction with the lattice-based scheme of [1]. With such a construction it is possible to get a very small expansion factor at each iteration of the chain, asymptotically close to 1, tweaking slightly GHV. This is a major step forward, as it allows us to obtain an expansion factor linear, instead of exponential, in d. Indeed, by chaining instances with shrinking expansion factors, $(2, 3/2, ..., (d-1)/(d-2), d/(d-1))$, the product of the expansion factors of d chained schemes with the alternative construction is d. Moreover, using the scheme's blinding properties the instantiation also ensures formula privacy. The full details of this alternative construction and its instantiation with GHV are left to the long version of this paper.

Acknowledgments

We want to warmly thank Shai Halevi for his support and very valuable comments on different versions of this work. We also want to thank Daniele Micciancio for his encouraging and useful recommendations, as well as the reviewers of Crypto'10 for their detailed comments.
The work of Javier Herranz is supported by Spanish MICINN Ministry, under project MTM2009-07694; and also by a *Ramón y Cajal* grant, partially funded by the European Social Fund (ESF) of the Spanish MICINN Ministry.

References

1. 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

2. Aguilar Melchor, C., Gaborit, P., Herranz, J.: Additively homomorphic encryption with d-operand multiplications. Cryptology ePrint Archive, Report 2008/378 (2008) http://eprint.iacr.org/.
3. 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
4. Barrington, D.A.M.: Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. J. Comput. Syst. Sci. **38**(1) (1989) 150–164
5. 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
6. 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
7. 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
8. van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully homomorphic encryption over the integers. In: 29th Annual Eurocrypt Conference (EUROCRYPT'10), French Riviera. Lecture Notes in Computer Science, Springer (2010) –
9. Domingo-Ferrer, J.: A new privacy homomorphism and applications. Information Processing Letters **60**(5) (1996) 277–282
10. 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
11. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory **31**(4) (1985) 469–472
12. 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
13. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of STOC'09, ACM Press (2009) 169–178
14. Gentry, C.: A fully homomorphic encryption scheme. PhD thesis, Stanford University (2009) crypto.stanford.edu/craig.
15. Gentry, C., Halevi, S., Vaikuntanathan, V.: A simple BGN-type cryptosystem from LWE. In: 29th Annual Eurocrypt Conference (EUROCRYPT'10), French Riviera. Lecture Notes in Computer Science, Springer (2010) –
16. 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
17. S. Goldwasser and S. Micali: Probabilistic encryption. Journal of Computer and System Sciences **28**(2) (1984) 270–299
18. 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

19. Ishai, Y., Paskin, A.: Evaluating branching programs on encrypted data. In: Second Theory of Cryptography Conference, TCC 2007, Amsterdam, The Netherlands, February 21-24, 2007, Proceedings. Volume 4392 of Lecture Notes in Computer Science., Springer (2007) 575–594
20. 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
21. Kushilevitz, E., Ostrovsky, R.: Replication is not needed: Single database, computationally-private information retrieval (extended abstract). In: FOCS: IEEE Symposium on Foundations of Computer Science (FOCS). (1997) 364–373
22. Lyubashevsky, V., Micciancio, D.: On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In: Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings. Volume 5677 of Lecture Notes in Computer Science., Springer (2009) 577–594
23. Mahajan, M.: Polynomial size log depth circuits: between NC^1 and AC^1. BEATCS: Bulletin of the European Association for Theoretical Computer Science **91** (2007)
24. Micciancio, D., Regev, O.: Lattice-Based Cryptography. In: Post Quantum Cryptography. Springer (2009) 147–191
25. Ostrovsky, R., E. Skeith III, W.: Private searching on streaming data. J. Cryptology **20**(4) (2007) 397–430
26. 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
27. Peikert, C.: Public-key cryptosystems from the worst-case shortest vector problem. In: Proceedings of STOC'09, ACM Press (2009) 333–342
28. Regev, O.: New lattice based cryptographic constructions. Journal of the ACM **51**(6) (2004) 899–942
29. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. Journal of the ACM **56**(6) (2009) 34
30. Rivest, R.L., Adleman, L., Dertouzos, M.L.: On data banks and privacy homomorphisms. In: Foundations of Secure Computation. Academic Press (1978) 169–180
31. 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
32. 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
33. Smart, N., Vercauteren, F.: Fully homomorphic encryption with relatively small key and ciphertext sizes. In: Public Key Cryptography - PKC 2010, 13th International Conference on Practice and Theory in Public-Key Cryptography, Paris, France, May 26-28, 2010, Proceedings. Volume 6056 of Lecture Notes in Computer Science., Springer (2010) 420–443
34. Steinwandt, R., Geiselmann, W.: Cryptanalysis of Polly Cracker. IEEE Transactions on Information Theory **48**(11) (2002) 2990–2991
35. 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
36. 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