Homomorphic Evaluation of Lattice-Based Symmetric Encryption Schemes

Pierre-Alain Fouque[1,3(✉)], Benjamin Hadjibeyli[2], and Paul Kirchner[3]

[1] Institut Universitaire de France, Université de Rennes 1, Rennes, France
[2] École normale supérieure de Lyon, Lyon, France
Benjamin.Hadjibeyli@ens-lyon.fr
[3] École normale supérieure, Paris, France
{Pierre-Alain.Fouque,Paul.Kirchner}@ens.fr

Abstract. Optimizing performance of Fully Homomorphic Encryption (FHE) is nowadays an active trend of research in cryptography. One way of improvement is to use a hybrid construction with a classical symmetric encryption scheme to transfer encrypted data to the Cloud. This allows to reduce the bandwidth since the expansion factor of symmetric schemes (the ratio between the ciphertext and the plaintext length) is close to one, whereas for FHE schemes it is in the order of 1,000 to 1,000,000. However, such a construction requires the decryption circuit of the symmetric scheme to be easy to evaluate homomorphically. Several works have studied the cost of homomorphically evaluating classical block ciphers, and some recent works have suggested new homomorphic oriented constructions of block ciphers or stream ciphers. Since the multiplication gate of FHE schemes usually squares the noise of the ciphertext, we cannot afford too many multiplication stages in the decryption circuit. Consequently, FHE-friendly symmetric encryption schemes have a decryption circuit with small multiplication depth.

We aim at minimizing the cost of the homomorphic evaluation of the decryption of symmetric encryption schemes. To do so, we focus on schemes based on learning problems: Learning With Errors (LWE), Learning Parity with Noise (LPN) and Learning With Rounding (LWR). We show that they have lower multiplicative depth than usual block ciphers, and hence allow more FHE operations before a heavy bootstrapping becomes necessary. Moreover, some of them come with a security proof. Finally, we implement our schemes in HElib. Experimental evidence shows that they achieve lower amortized and total running time than previous performance from the literature: our schemes are from 10 to 10,000 more efficient for the time per bit and the total running time is also reduced by a factor between 20 to 10,000. Of independent interest, the security of our LWR-based scheme is related to LWE and we provide an efficient security proof that allows to take smaller parameters.

1 Introduction

Fully Homomorphic Encryption (FHE) is nowadays one of the most active trend of research in cryptography. In a nutshell, a FHE scheme is an encryption scheme

T.N. Dinh and M.T. Thai (Eds.): COCOON 2016, LNCS 9797, pp. 269–280, 2016.
DOI: 10.1007/978-3-319-42634-1_22

that allows evaluation of arbitrarily complex programs on encrypted data. This idea has been introduced by Rivest et al. [26] in 1978, while the first plausible construction has been given by Gentry [16] in 2009. Since, numerous papers have focused on improving the efficiency of the constructions. Even if there still remains works before FHE becomes practical, it arouses more and more interest and the scope of application goes from genomics to finance [24].

One way of improvement has been introduced in [24]. It focuses on minimizing the communication complexity of the scheme. The idea is to use a "hybrid" encryption scheme: some parts of the scheme are replaced by a symmetric encryption scheme. Instead of encrypting the data under the FHE scheme, the client will only encrypt its symmetric key under the FHE scheme, and encrypt its data under the symmetric scheme. The cloud will then homomorphically evaluate the decryption of the symmetric scheme on the symmetrically encrypted data and the homomorphically encrypted symmetric key, to get a ciphertext corresponding to a homomorphic encryption of the data. Clearly, such a construction has low communication complexity, since the only online data transfer is made under the symmetric scheme. However, the cloud might pay a huge cost at the homomorphic evaluation of the symmetric decryption. Thus, one can look for the most "FHE-friendly" symmetric encryption scheme to use in the hybrid.

Being "FHE-friendly" consists in optimizing several criteria. First, as the application we gave suggests, we want a scheme with a small expansion factor, so that the communication complexity stays low. Then, other criteria depend on the FHE construction we are building upon. All current FHE schemes are based on variants of Gentry's initial idea: ciphertext consists of encryption of data with noise, and homomorphic operations increase this noise. When the upper bound of noise is reached, one has to "bootstrap", to reduce the noise to its initial level. Typically, functions are represented as arithmetic circuits and multiplications have a far higher cost than additions in terms of noise. Thus, we will want to minimize the multiplicative depth of the decryption circuit of our symmetric scheme. In addition, we will also take into account the total running time of our homomorphic evaluation step. This metric highly depends on the chosen FHE scheme, but multiplications often happen to be the main bottleneck again.

Our Contributions. In this paper, we focus on symmetric schemes having shallow decryption circuits. We build secure schemes with constant or small decryption circuit, namely with small multiplication depth. Contrary to the direction followed by many recent work, that tweak block ciphers or stream ciphers [3,9], our approach is related to provable security. Indeed, we notice that one can construct lattice-based schemes with very small decryption circuit and then, we evaluate the performances of our schemes using HElib to compare them with other symmetric ciphers. Finally, we try to use HElib features (full packing and parallelization) in order to achieve better performances. We describe two kinds of ciphers: the first family has its security related to the difficulty of solving the LPN problem in specific instances, while the second family has a *security proof* based on the LWE problem. The first construction is similar to "symmetric cryptography" since we do not have a clean security proof and consequently,

we provide a more thorough security analysis. However, the security seems to be easier to understand than ad-hoc constructions usually used in symmetric cryptography, since the security problem on which the scheme is based can be formally stated. We present a very efficient construction specifically tailored to this problem to secure our construction from Arora-Ge type of attack on LPN. The performance of the schemes from this family can be 10 times more efficient than the most efficient previous cipher. For the second family, we have a rigorous security proof related to LWE, while the scheme is based on LWR. The performance of the second family can be very efficient, about 10,000 times faster, but the caveat is that the decrypted plaintext contains random bits in the least significant bits if we do not compute homomorphically the truncation using the costly ExtractDigits function. Therefore, if we want to remove the erroneous bits, the performances become equivalent to previous ciphers, while being more efficient than AES. In some cases, we can compute with such noise.

We notice that contrary to what is claimed in many works [24], it is not necessary to re-encrypt the symmetrically-encrypted ciphertext using the FHE scheme when the server receives the data. We show that the evaluation of the homomorphic decryption procedure gives ciphertexts encrypted with FHE. This improves the performance of the scheme, since we homomorphically evaluate the function that maps the key K to the $\mathsf{Dec}(K, c)$, given the ciphertext c and some multiplications in Dec will be simplified once c is known.

Then, we describe our efficient FHE-friendly symmetric schemes based on lattices, and more precisely on learning problems. Our results show that we can get circuits with very small multiplication depth for the decryption algorithms of these schemes. In addition, their security relies on hard problems or on hard instances of lattice problems in the worst cases, as opposed to usual block ciphers.

We present a scheme whose security is based on the Learning Parity With Noise problem (LPN) introduced in [18]. We have to specify an error correcting code (ECC) for this scheme so that the decryption circuit is small. We choose to use a repetition code in order to simplify the decoding and reduce its circuit in term of multiplications. More complex ECC exist with constant decoding such as [19] but they are only interesting from an asymptotic point of view. However, prohibiting decryption failures makes the scheme vulnerable to the Arora-Ge [6] attack and to avoid its most efficient variant [2] using Gröbner basis algorithms, we use a very efficient transformation, similar to random locally function [5], which increases the algebraic degree of the polynomials system. We provide a detailed analysis of this attack. The function we propose is also very similar to [1] and we can show that our construction achieves better influence parameters, but it has higher complexity class since we need a logarithmic depth circuit.

Then, we introduce another scheme whose security is based on the Learning With Rounding problem (LWR) and a very similar version whose security relies directly on the Learning With Errors (LWE) problem. In order to encrypt many bits using small parameters, we provide a direct proof from LWE to the security of the scheme. We do not rely on any reduction from LWE to LWR since the first reduction given by Banerjee et al. [7] requires exponential parameters and the one by Alwen et al. [4] requires parameter linear in the number

of samples. Here, our reduction is only logarithmic in the number of samples. Furthermore, we extend both schemes to their ring versions. In this case, we optimized the number of multiplications using a FFT algorithm to compute the polynomial multiplications. Finally, we extend them to their module versions, which generalizes standard and ring versions.

Along with a theoretical analysis, we give a homomorphic evaluation in HElib to make practical comparisons. While the homomorphic evaluation of AES went down to 11 ms per bit [17] and LowMC, a block cipher designed to be FHE-friendly (and whose security has recently been analyzed [11,12]), went down to 3 ms per bit [3], which was the best so far, we go under a millisecond per bit (with the module version of our LPN scheme). In some scenario, our performance for the scheme based on LWR are drastically better if we allow FHE-encrypted plaintexts to contain noise in the least significant bits. Moreover, our schemes are a lot more flexible, in the sense that they need smaller FHE parameters, and while these performance were amortized over a computation taking several minutes, the evaluation of our schemes takes only from a second to a minute.

Related Work. Many papers have presented homomorphic evaluations of block ciphers. It has started in [17], where AES has been chosen as a benchmark for measuring the performance of HElib. Then, performance has been improved in [23]. AES has then been used as benchmark for comparing FHE schemes in [10,13]. Similarly, Simon has been used to compare FHE schemes [21]. Recently, the problem has been taken the other way round, with works trying to find the most FHE-friendly block cipher. First, a lightweight block cipher like Prince has been suggested and evaluated [14]. Then, a new block cipher, LowMC, has been designed specifically for this kind of application [3], as well as for multiparty computations. Finally, using stream ciphers has also been proposed [9].

Organization of the Paper. In Sect. 2 we recall definitions about symmetric encryption and Lattice problems in Cryptography. In Sect. 3, we explain how we use homomorphic operation more efficiently. Then we introduce in Sect. 4 our symmetric schemes based on learning problems: LPN, LWR and LWE. The security and performance analysis of the schemes are proved in the final version.

2 Preliminaries

Symmetric Encryption. We will say that a function of k (from positive integers to positive real numbers) is negligible if it approaches zero faster than any inverse polynomial, and noticeable if it is larger than some inverse polynomial (for infinitely many values of k).

Definition 1. *A symmetric encryption scheme is a tuple* $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ *of Probabilistic Polynomial-time (PPT) algorithms as follows:*

- $\mathsf{Gen}(1^\lambda)$*: given a security parameter* λ*, output a secret key* k*;*
- $\mathsf{Enc}(k, m)$*: given a key* k *and a message* m*, output a ciphertext* c*;*
- $\mathsf{Dec}(k, c)$*: given a key* k *and a ciphertext* c*, output a message* m'*;*

and which satisfies the correctness property: if $k := \mathsf{Gen}(1^\lambda)$*, then for all messages* $m, \Pr[\mathsf{Dec}(k, \mathsf{Enc}(k, m)) \neq m]$ *is negligible (in* λ*).*

For the sake of clarity, we will often write the key as a subscript and the scheme name as a superscript of our algorithms, like in Enc_k^S. Semantic security is implied by the following property, which will be satisfied by our schemes.

Definition 2. *A symmetric encryption scheme* S *has pseudo-random ciphertexts (ciphertexts indistinguishable from random) if no PPT* $\mathcal{A}$ *can distinguish between ciphertexts from the scheme and the uniform distribution, i.e. for all PPT* $\mathcal{A}$ *and for all messages* m*, it holds that* $\left|\Pr[\mathcal{A}(\mathsf{Enc}_k^S(m), 1^\lambda) = 1] - \Pr[\mathcal{A}(r, 1^\lambda) = 1]\right|$ *is negligible (in* λ*), where* r *is drawn randomly over the ciphertext space.*

Learning Problems. Given a finite set S and a probability distribution D on S, $s \leftarrow D$ denotes the drawing of an element of S according to D and $s \leftarrow S$ the random drawing of an element of S endowed with uniform probability.

Learning with Errors. The Gaussian distribution with standard deviation σ is defined on $\mathbb{R}$ by the density function $\frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{1}{2}(\frac{x}{\sigma})^2)$. The Learning With Errors problem (LWE) has been introduced in [25]. For $s \in \mathbb{Z}_q^k$, the LWE distribution $D_{s,\chi}^{LWE}$ is defined over $\mathbb{Z}_q^k \times \mathbb{Z}_q$ and consists in samples $(a, \langle a, s\rangle + e)$ where $a \leftarrow \mathbb{Z}_q^k$ and $e \leftarrow \chi$ for some distribution χ over $\mathbb{Z}_q$. Typically, χ is taken to be some integral Gaussian distribution when assuming that LWE is hard. As in most works [25], we will consider here rounded Gaussian distributions: it basically consists in sampling a Gaussian distribution, reducing the result modulo 1, multiplying it by q and rounding it to the nearest integer. LWE consists, for s chosen according to some distribution over $\mathbb{Z}_q^k$ (typically, the uniform distribution), in distinguishing between any desired number of samples from $D_{s,\chi}^{LWE}$ and the same number of samples drawn from the uniform distribution over $\mathbb{Z}_q^k \times \mathbb{Z}_q$. For rounded Gaussian distributions, LWE is usually considered to be hard when the standard deviation σ verifies $\sigma > \sqrt{k}$ [25].

LWE can be extended into a ring version RLWE [22]. Let $R = \mathbb{Z}[X]/(P(X))$ for a monic irreducible polynomial P of degree k, and let $R_q = R/qR$. Generally, P is chosen to be some power-of-two cyclotomic polynomial, which are of the form $X^{2^z} + 1$. For an element $s \in R_q$, we define the RLWE distribution $D_{s,\chi}^{RLWE}$ over $R_q \times R_q$ by samples $(a, a.s + e)$ where $a \leftarrow R_q$ and $e \leftarrow \chi^k$ where χ^k consists in k independent samples from χ and e is interpreted as an element of R_q. The Ring-LWE problem (RLWE) consists, for s drawn according to some distribution over R_q, in distinguishing $D_{s,\chi}^{RLWE}$ from the uniform distribution over $R_q \times R_q$.

We will also use the Module-LWE problem (MLWE). It has been introduced in [8] under the name of GLWE, for General LWE. However, we will call it MLWE as in [20], because it indeed corresponds to introducing a module structure over LWE. For an element $s \in R_q^k$, where the underlying ring polynomial has degree d, we define the MLWE distribution $D_{s,\chi}^{MLWE}$ over $R_q^k \times R_q$ by samples $(a, \langle a.s\rangle + e)$ where $a \leftarrow R_q^k$ and $e \leftarrow \chi^d$ is interpreted as an element of R_q. MLWE generalizes LWE and RLWE: LWE is when $d = 1$ and RLWE when $k = 1$.

By a standard hybrid argument, LWE can be extended to several secrets. It can be shown that the problem which consists in distinguishing samples $(a, \langle a, s_1\rangle + e_1, \ldots, \langle a, s_n\rangle + e_n) \in \mathbb{Z}_q^k \times \mathbb{Z}_q^n$ from the uniform distribution over $\mathbb{Z}_q^k \times \mathbb{Z}_q^n$, where each $s_j \in \mathbb{Z}_q^k$ is chosen independently for any $n = poly(k)$, is at least as hard as LWE for a single secret s. An analogous statement can be shown for RLWE and MLWE. Finally, the LWE [25], RLWE [22] and MLWE [20] hardness assumptions have been reduced to standard lattice assumptions. The security of MLWE seems to be intermediate between that of LWE based on hardness results in arbitrary lattices and the security of RLWE in ideal lattices.

Learning Parity with Noise (LPN). We denote by $\mathcal{B}_\eta$ the Bernoulli distribution of parameter $\eta \in [0, 1]$, i.e. a bit $b \leftarrow \mathcal{B}_\eta$ is chosen such that $\Pr[b = 1] = \eta$ and $\Pr[b = 0] = 1 - \eta$. The LPN problem consists in LWE for $q = 2$. The distribution χ chosen over $\mathbb{Z}_2$ corresponds to a Bernoulli distribution. We extend LPN to RLPN and MPLN. The only difference is that the underlying polynomial will not be cyclotomic anymore, but some irreducible polynomial modulo 2. Similarly, these problems are also extended to a polynomial number of secrets. The main difference between LWE and LPN is that the security of LPN remains heuristic because no reduction has been made so far to lattice problems.

Learning with Rounding (LWR). The LWR problem has been introduced in [7] as a derandomization of LWE. The idea is to replace the addition of a random noise by a rounding function. Let k be the security parameter and moduli $q \geq p \geq 2$ be integers. We define the function $\lfloor . \rceil_p : \mathbb{Z}_q \to \mathbb{Z}_p$ by $\lfloor x \rceil_p = \lfloor (p/q).\bar{x} \rceil$, where $\bar{x}$ is an integer congruent to x mod q. We extend $\lfloor . \rceil_p$ component-wise to vectors and matrices over $\mathbb{Z}_q$. Let R denote the cyclotomic polynomial ring $R = \mathbb{Z}[z]/(z^k+1)$ for k a power of two. For any modulus q, we define the quotient ring $R_q = R/qR$ and extend $\lfloor . \rceil_p$ coefficient-wise to it. Note that we can use any common rounding method, like the floor or ceiling functions. In our implementations, we use the floor, because it is equivalent to dropping the least-significant digits in base 2 when q and p are both powers of 2.

For a vector $s \in \mathbb{Z}_q^k$, the LWR distribution D_s^{LWR} is defined over $\mathbb{Z}_q^k \times \mathbb{Z}_p$ by elements $(a, \lfloor \langle a, s\rangle \rceil_p)$ with $a \leftarrow \mathbb{Z}_q^k$. For a vector $s \in R_q$, the ring-LWR (RLWR) distribution D_s^{RLWR} is defined over $R_q \times R_p$ by elements $(a, \lfloor a.s \rceil_p)$ with $a \leftarrow R_q$. And for a vector $s \in R_q^k$, the module-LWR (MLWR) distribution D_s^{MLWR} is defined over $R_q^k \times R_p$ by elements $(a, \lfloor \langle a.s\rangle \rceil_p)$ with $a \leftarrow R_q^k$. For a given distribution D over $s \in \mathbb{Z}_q^k$, LWR consists in distinguishing between any desired number of independent samples from D_s^{LWR} and the same number of samples drawn uniformly and independently from $\mathbb{Z}_q^k \times \mathbb{Z}_p$. RLWR and MLWR are defined analogously. All these problems can be extended to several secrets, as stated for LWE. The LWR has been reduced to LWE when q/p is exponential in k [7], and when q/p is poly(k) and linear in the number of samples by [4].

3 Fully-Homomorphic Encryption (FHE)

While classical encryption preserves the privacy of information, homomorphic encryption aims also at making some computation on the encrypted data.

FHE Definitions. Formally, we have a message space M with a set of functions f we would like to compute on messages, and we want an algorithm which efficiently computes functions f' on the ciphertext space C such that $\mathsf{Dec}(f'(\{c_i\}_i)) = f(\{\mathsf{Dec}(c_i)\}_i)$. Thus, we want the decryption function to be a homomorphism from C to M for these functions f. This notion, originally called a privacy homomorphism, was introduced in [26]. Here is a formal definition of a homomorphic scheme, sometimes referred to as "somewhat homomorphism".

Definition 3. *Let $\mathcal{F}$ be a set of functions. A $\mathcal{F}$-homomorphic encryption (HE) scheme is a tuple of PPT algorithms* $(\mathsf{Gen}, \mathsf{Encrypt}, \mathsf{Decrypt}, \mathsf{Eval})$ *as follows:*

- $\mathsf{Gen}(1^\lambda)$*: given a security parameter λ, output a public key pk, a secret key sk and an evaluation key ek;*
- $\mathsf{Enc}(pk, m)$*: given the public key pk and a message m, output a ciphertext c;*
- $\mathsf{Dec}(sk, c)$*: given the secret key sk and a ciphertext c, output a message m';*
- $\mathsf{Eval}(ek, f, \Psi = (c_1, \ldots, c_l))$*: given the evaluation key, a function f and a tuple Ψ of l ciphertexts, where l is the arity of f, output a ciphertext c';*

satisfying the correctness property: for all functions $f \in \mathcal{F}$ and messages $\{m_i\}_{i \leq l}$, where l is the arity of f, if $(pk, sk, ek) := \mathsf{Gen}(1^\lambda)$ and $c_i := \mathsf{Enc}(pk, m_i)$ for all i, then $\Pr[\mathsf{Dec}(sk, \mathsf{Eval}(ek, f, (c_1, \ldots, c_l))) \neq f(m_1, \ldots, m_l)]$ is negligible (in λ).

Homomorphic Evaluation of Symmetric Encryption Schemes. We now give a more precise description of the scenario where a symmetric encryption scheme is used to improve FHE performance, as described in [24], and on which we will to rely to analyse the performance of our schemes.

Optimizing Communication with the Cloud. Consider the setting where a client uploads its data encrypted under a FHE scheme on a cloud service and wants the cloud to compute on this data and return encrypted outputs. Typically, FHE schemes come with an expansion factor of the order of 1,000 to 1,000,000. To mitigate this problem, the client will send its data encrypted under some semantically secure symmetric encryption scheme (which, by itself, is not homomorphic at all) along with the homomorphic encryption of its symmetric key. Then, the steps of symmetric decryption can all be carried out on homomorphically encrypted entries. Thus, the cloud can obtain the data encrypted under the FHE scheme by homomorphically evaluating the decryption circuit.

Here is a formal description of the protocol. Let $H = (\mathsf{Gen}^H, \mathsf{Enc}^H, \mathsf{Dec}^H, \mathsf{Eval}^H)$ be a FHE scheme and let $S = (\mathsf{Gen}^S, \mathsf{Enc}^S, \mathsf{Dec}^S)$ be a symmetric encryption scheme. Let λ be the security parameter and m be the data the client wants to send to the cloud. Let $(pk, sk, ek) := \mathsf{Gen}^H(1^\lambda)$ and $k := \mathsf{Gen}^S(1^\lambda)$.

- The client sends messages $c_1 := \mathsf{Enc}^H_{pk}(k)$ and $c_2 := \mathsf{Enc}^S_k(m)$ to the cloud.
- Given a couple of ciphertexts (c_1, c_2) received from the client, the cloud computes (either at the reception or just before further computing) $x = \mathsf{Enc}^H_{pk}(c_2)$ and then $c = \mathsf{Eval}_{ek}(\mathsf{Dec}^S, c_1, x)$: this is why we need an efficient homomorphic evaluation of the decryption circuit of our symmetric scheme.

– Now, the cloud possesses a FHE-encrypted ciphertext c, which means that $\mathsf{Dec}^H_{sk}(c) = m$. Furthermore, it can now homomorphically evaluate any function f: for all f, $\mathsf{Dec}^H_{sk}(\mathsf{Eval}^H_{ek}(f, c)) = f(c)$.

Indeed, if the evaluation algorithm allows constant arguments, i.e. arguments which are not homomorphically encrypted, this scenario can be optimized further, simply by noticing that c_2 does not have to be homomorphically encrypted. Thus, when receiving c_1 and c_2, the cloud will just directly compute $c = \mathsf{Eval}_{ek}(\mathsf{Dec}^S, c_2, c_1)$. It still has to homomorphically evaluate the decryption circuit of the symmetric scheme, but it saves a homomorphic encryption, and operations with constants might be faster. This can also be seen as evaluating the function $K \mapsto Dec^S(K, m)$, which depends on m.

All the symmetric encryption schemes we will use are based on lattices, and, more precisely, on learning problems. Some of them will rely on the LPN problem, while the others will rely on the LWR or on the LWE problem. Our initial goal is to construct efficient FHE-friendly encryption schemes. Symmetric encryptions are used in FHE scenario in order to transfer the cloud. Here, we first describe a much more efficient scenario for symmetric and homomorphic encryption than the classical scenario described in [24].

4 FHE-Friendly Symmetric Encryption Based on Learning

An Encryption Scheme Based on MLPN. Our first encryption scheme is a generalization of the scheme introduced in [18], under the name of $\mathtt{LPN-C}$ by Gilbert *et al.* A $[n, m, d]$ linear binary (error-correcting) code C is a linear subspace of $\mathbb{F}_2^n$ with dimension m such that d is the minimum ℓ^1 distance between two elements of the code. We associate it with an encoding function $E : \mathbb{F}_2^m \to C$ and a decoding function $D : C \to \mathbb{F}_2^m$. $\mathtt{LPN-C}$ is a symmetric encryption scheme whose security can be reduced to the hardness of LPN. Let E and D be respectively the encoding and decoding functions of a $[n, m, d]$ linear binary code.

Here, we describe the more general version of our scheme. Similarly, we can define $\mathtt{LPN-C}$ and $\mathtt{RLPN-C}$ based on LPN and RLPN problems.

Definition 4 ($\mathtt{MLPN-C}$). *Let d, k and n be polynomials in λ. Let consider $\mathbb{F}_{2^d}$ a finite field defined by an irreducible polynomial P of degree d. The symmetric encryption scheme* $\mathtt{MLPN-C}$ *is defined as follows:* $\mathsf{Gen}(1^\lambda)$*: output* $S \leftarrow \mathbb{F}_{2^d}^{k \times n}$*;* $\mathsf{Enc}_S(x)$*: output* $(a, E(x) \oplus a.S \oplus e)$*, where* $a \leftarrow \mathbb{F}_{2^d}^k$ *and* $e \leftarrow \mathcal{B}_\eta^{d \times n}$ *is interpreted as an element of* $\mathbb{F}_{2^d}^n$*; and* $\mathsf{Dec}_S(a, y)$*: output* $D(y \oplus a.S)$*.*

For a message of m bits, this scheme produces a ciphertext of $n + k$ bits. Indeed, the expansion factor can tend to the one of the linear code we are using, which is n/m, since n is any polynomial in k. Furthermore, one can consider that a does not have to be sent, and can be replaced, for example, by the seed used in order to generate it.

We choose the 3-repetition code to have a small multiplication depth circuit of degree 2. We define the encoding scheme for $a \in \mathbb{F}_{2^d}$ as $(a^{2^{d-1}}, a^{2^{d-1}}, a^{2^{d-1}})$. In order to decode a code word $(a, b, c) \in (\mathbb{F}_{2^d})^3$, we compute $ab + bc + ac$. (The normal encoding with would be (a, a, a) and the decoding $(ab + ac + bc)^{2^{d-1}}$, but we prefer to incorporate the power 2^{d-1} in the encoding in order to make the homomorphic part more efficient.)

Proposition 1 [18]**.** $\mathtt{LPN} - \mathtt{C}$ *(resp.* $\mathtt{RLPN} - \mathtt{C}$, $\mathtt{MLPN} - \mathtt{C}$*) is semantically secure as soon as the corresponding* LPN *(resp.* RLPN, MLPN*) problem is hard.*

As it stands, if we do not bound the number of errors sent along with a message, this scheme will produce decryption failures. They will happen when the Hamming weight of the noise vector e is greater than the correction capacity of the error-correcting code. We study the probability of decryption failures and we can choose the noise parameter η so that this probability is very low. However, in this case, more efficient attacks than BKW algorithm $O(2^{k/\log(k/-\log(1-2\eta))})$ can be used to recover the secret in time $O(k^3/(1-\eta)^k)$. To thwart attacks, we will increase their complexity using *delinearization steps* described later.

An important point is the choice of the error-correcting code used in the scheme. In our context, we would like a code with shallow decoding circuit, and indeed, codes with shallow decoding circuits are quite rare. For example, linear codes, which have really simple encoding circuits, have complicated decoding circuits. We would like to use a 3-repetition code, which has decoding depth 1. We will keep using such a code in practice as it leads to very efficient homomorphic performance, but the particular structure given to the noise requires a careful analysis of the security, that we will do in the following section. Consequently, in order to also thwart this attack, the delinearization steps can be useful.

Delinearization Steps. In order to counter the Arora-Ge attack, we choose to add some noise on our values after computing the scalar product. In practice, the ciphertext we send consists of $(a, E(x) \oplus F(a.S) \oplus e)$ where F is some function involving enough layers of multiplication so that the Arora-Ge attack does not work. Of course, this step increases the parameters we have to choose for homomorphically evaluating our scheme, however, a few steps (3 in order to have a sufficient security parameter) are needed in order to prevent the attacks. We admit that such techniques are far away from provable security and come from symmetric cryptography since F is a kind of cheap non-linear operation. However, contrary to the symmetric setting, here the adversary cannot control the inputs to this function and many well-known chosen plaintext attacks are then prohibited and only known plaintext attacks need to be studied. It is easy to see how to adapt the decryption process. The function F we choose works as follows: on a vector V, it consecutively applies several transformations T_i, for $i \leq d$, such that $[T_i(V)]_j = V_j + V_{x_{ij}} * V_{y_{ij}}$, where the set of indices x_{ij} and y_{ij} is chosen so that monomials do not cancel. The degree of F in the inputs is 2^d. We estimated the number of applications of such transformations needed in order to counteract the most efficient variant of the Arora-Ge attack and for $n = 512$, three steps seem reasonable.

Even though, our scheme has a security proof, the parameters we choose do not allow us to use the reduction. Indeed, we pick either a structured noise (and Arora-Ge algorithms must be taken into account) or a very small noise to reduce the decryption failure. Therefore the delinearization steps increase the complexity of these attacks and a thorough security analysis is needed. These steps are similar to local random functions [1,5] and we can show similar security.

An Encryption Scheme Based on LWR. We present a symmetric encryption scheme whose security can be reduced to the LWE problem. We describe the more general $\mathtt{MLWR-SYM}$ and we can similarly define $\mathtt{LWR-SYM}$ and $\mathtt{RLWR-SYM}$.

Definition 5 ($\mathtt{MLWR-SYM}$). *Let d, k and n be polynomials in λ. Let consider R, with underlying polynomial P of degree d. The symmetric encryption scheme* $\mathtt{MLWR-SYM}$ *is defined as follows:* $\mathsf{Gen}(1^\lambda)$*: output* $S \leftarrow R_q^k$*;* $\mathsf{Enc}_S(x)$*: output* $(a, x + \lfloor a.S \rceil_p)$*, where* $a \leftarrow R_q^k$ *and* $\mathsf{Dec}_S(a, y)$*: output* $(y - \lfloor a.S \rceil_p)$.

For a message of size n over $\mathbb{Z}_p$, this scheme produces a ciphertext consisting of a random vector of length k over $\mathbb{Z}_q$ and a vector of length n of $\mathbb{Z}_p$. Thus, the expansion factor is $1 + \frac{\log q}{\log p}\frac{k}{n}$. Now, for the same reasons as for $\mathtt{LPN-C}$, this expansion factor can basically be considered as 1. The decryption circuit has depth one plus the depth of the rounding function. When using the floor function and if q and p are power of two, then the rounding consists in dropping the least significant bits of the result. The most efficient FHE-friendly encryption scheme works in only adding the plaintext on the $\log p$ most significant bits of $\lfloor a.S \rceil_p$ to avoid the costly ExtractDigits homomorphic function and the returned plaintext contains noise.

Proposition 2. $\mathtt{LWR-SYM}$ *(resp.* $\mathtt{RLWR-SYM}$, $\mathtt{MLWR-SYM}$*) is semantically secure as soon as the corresponding LWR (resp. RLWR, MLWR) problem is hard.*

An Encryption Scheme Based on LWE. We can adapt $\mathtt{LWR-SYM}$ so that its security proof relies directly on the LWE assumption. This new scheme will basically be the same as the previous one, except that the vector a will be chosen according to some biased distribution D_S. The distribution D_S we will use is defined on $\mathbb{Z}_q^n$ and depends on some matrix $S \in \mathbb{Z}_q^{k \times n}$ and a distribution χ.

We will quickly present it in the case where $k = 1$. It verifies the property that $\Pr[D_s = a]$ is proportional to $\Pr[\lfloor \frac{p}{q}.(\overline{\langle a, s\rangle + e})\rceil - \frac{p}{q}.(\overline{\langle a, s\rangle + e})| < \frac{1}{4}]$, where $e \leftarrow \chi$ and $\overline{\langle a, s\rangle + e}$ means that $\langle a, s\rangle + e$ is interpreted as an element of $\mathbb{Z}$. This basically means that we want the value $(p/q).(\overline{\langle a, s\rangle + e})$ to be close to its rounding for our samples a. One can efficiently sample according to this distribution D_s: sample a uniformly, and output it if and only if, when sampling e according to χ, the value $(p/q).(\overline{\langle a, s\rangle + e})$ is at distance less than $1/4$ from its rounding. Since the distribution of $\langle a, s\rangle + e$ is indistinguishable from uniform, the probability that a vector a gets rejected is (around) $1/2$. To extend this distribution to a matrix S, we will take a distance of $1/2 - 1/4n$ instead of $1/4$.

Definition 6 ($\mathtt{LWE-SYM}$). *Let k and n be polynomials in λ. The symmetric encryption scheme* $\mathtt{LWE-SYM}$ *is defined as follows:* $\mathsf{Gen}(1^\lambda)$*: output* $S \leftarrow \mathbb{Z}_q^{k \times n}$*;* $\mathsf{Enc}_S(x)$*: output* $(a, x + \lfloor a.S \rceil_p)$*, where* $a \leftarrow D_S$ *and* $\mathsf{Dec}_S(a, y)$*: output* $y - \lfloor a.S \rceil_p$.

Our scheme relying on RLWR and MLWR can also be adapted to schemes called $\texttt{RLWE}-\texttt{SYM}$ and $\texttt{MLWE}-\texttt{SYM}$ in a similar way, that we do not explicit here. We now show that the security of $\texttt{LWE}-\texttt{SYM}$ (resp. $\texttt{RLWE}-\texttt{SYM}$, $\texttt{MLWE}-\texttt{SYM}$) directly reduces to the LWE (resp. RLWE, MLWE) hardness assumption. Our reduction is better than previous ones in the case of one secret. We introduce the problem LWR_D (resp. RLWR_D, MLWR_D) as the same problem as LWR (resp. RLWR, MLWR) except that a is drawn according to the distribution D. To choose secure parameters for LWR, we picked $k = 128$ and $p < \sqrt{q}$ according to [15].

Proposition 3. $\texttt{LWE}-\texttt{SYM}$ *(resp.* $\texttt{RLWE}-\texttt{SYM}$*,* $\texttt{MLWE}-\texttt{SYM}$*) is semantically secure as soon as the corresponding LWE (resp. RLWE, MLWE) problem is hard.*

Since σ is usually chosen to be at least $\sqrt{k}$ in LWE, our modulus-to-error ratio q/p verifies $q/p > O(n\sqrt{k \log m})$, which is an improvement compared to previous reductions which depend on m rather than $\log m$.

The schemes $\texttt{LWR}-\texttt{SYM}$ and $\texttt{LWE}-\texttt{SYM}$ are similar, except that $\texttt{LWE}-\texttt{SYM}$ involves some checking when generating vectors a. Thus, $\texttt{LWE}-\texttt{SYM}$ has exactly the same efficiency as $\texttt{LWR}-\texttt{SYM}$ for the homomorphic part. The only difference of performance lies in the symmetric encryption, because the generation of the vectors a is a constant factor longer. Thus, we only present the implementation of $\texttt{LWR}-\texttt{SYM}$, because we are only interested in the homomorphic part.

References

1. Akavia, A., Bogdanov, A., Guo, S., Kamath, A., Rosen, A.: Candidate weak pseudorandom functions in $AC0 \circ MOD_2$. In: Innovations in Theoretical Computer Science, ITCS 2014, Princeton, NJ, USA, 12–14 January 2014, pp. 251–260 (2014)
2. Albrecht, M.R., Cid, C., Faugère, J., Fitzpatrick, R., Perret, L.: Algebraic algorithms for LWE problems. In: IACR Cryptology ePrint Archive 2014, p. 1018 (2014)
3. Albrecht, M.R., Rechberger, C., Schneider, T., Tiessen, T., Zohner, M.: Ciphers for MPC and FHE. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 430–454. Springer, Heidelberg (2015)
4. Alwen, J., Krenn, S., Pietrzak, K., Wichs, D.: Learning with rounding, revisited - new reduction, properties and applications. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013, Part I. LNCS, vol. 8042, pp. 57–74. Springer, Heidelberg (2013)
5. Applebaum, B.: Cryptographic hardness of random local functions - survey. In: Electronic Colloquium on Computational Complexity (ECCC), vol. 22, p. 27 (2015)
6. Arora, S., Ge, R.: New algorithms for learning in presence of errors. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) ICALP 2011, Part I. LNCS, vol. 6755, pp. 403–415. Springer, Heidelberg (2011)
7. Banerjee, A., Peikert, C., Rosen, A.: Pseudorandom functions and lattices. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 719–737. Springer, Heidelberg (2012)
8. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In: Ostrovsky, R. (ed.) 52nd FOCS, pp. 97–106. IEEE Computer Society Press, October 2011
9. Canteaut, A., Carpov, S., Fontaine, C., Lepoint, T., Naya-Plasencia, M., Paillier, P., Sirdey, R.: How to compress homomorphic ciphertexts. In: IACR Cryptology ePrint Archive 2015, p. 113 (2015)

10. Cheon, J.H., Coron, J.-S., Kim, J., Lee, M.S., Lepoint, T., Tibouchi, M., Yun, A.: Batch fully homomorphic encryption over the integers. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 315–335. Springer, Heidelberg (2013)
11. Dinur, I., Liu, Y., Meier, W., Wang, Q.: Optimized interpolation attacks on LowMC. Cryptology ePrint Archive, Report 2015/418 (2015). http://eprint.iacr.org/
12. Dobraunig, C., Eichlseder, M., Mendel, F.: Higher-order cryptanalysis of LowMC. Cryptology ePrint Archive, Report 2015/407 (2015). http://eprint.iacr.org/
13. Doroz, Y., Hu, Y., Sunar, B.: Homomorphic AES evaluation using NTRU. Cryptology ePrint Archive, Report 2014/039 (2014). http://eprint.iacr.org/2014/039
14. Doröz, Y., Shahverdi, A., Eisenbarth, T., Sunar, B.: Toward practical homomorphic evaluation of block ciphers using prince. In: Böhme, R., Brenner, M., Moore, T., Smith, M. (eds.) FC 2014 Workshops. LNCS, vol. 8438, pp. 208–220. Springer, Heidelberg (2014)
15. Duc, A., Tramèr, F., Vaudenay, S.: Better algorithms for LWE and LWR. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 173–202. Springer, Heidelberg (2015)
16. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Mitzenmacher, M. (ed.) 41st ACM STOC, pp. 169–178. ACM Press, May/June 2009
17. Gentry, C., Halevi, S., Smart, N.P.: Homomorphic evaluationof the AES circuit. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 850–867. Springer, Heidelberg (2012)
18. Gilbert, H., Robshaw, M., Seurin, Y.: How to encrypt with the LPN problem. In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 679–690. Springer, Heidelberg (2008)
19. Goldwasser, S., Gutfreund, D., Healy, A., Kaufman, T., Rothblum, G.N.: Verifying and decoding in constant depth. In: Johnson, D.S., Feige, U. (eds.) 39th ACM STOC, pp. 440–449. ACM Press, June 2007
20. Langlois, A., Stehlé, D.: Worst-case to average-case reductions for module lattices. Des. Codes Cryptogr. **75**(3), 565–599 (2015)
21. Lepoint, T., Naehrig, M.: A comparison of the homomorphic encryption schemes FV and YASHE. In: Pointcheval, D., Vergnaud, D. (eds.) AFRICACRYPT 2014. LNCS, vol. 8469, pp. 318–335. Springer, Heidelberg (2014)
22. Lyubashevsky, V., Peikert, C., Regev, O.: A toolkit for ring-LWE cryptography. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 35–54. Springer, Heidelberg (2013)
23. Mella, S., Susella, R.: On the homomorphic computation of symmetric cryptographic primitives. In: Stam, M. (ed.) IMACC 2013. LNCS, vol. 8308, pp. 28–44. Springer, Heidelberg (2013)
24. Naehrig, M., Lauter, K., Vaikuntanathan, V.: Can homomorphic encryption be practical? In: Proceedings of the 3rd ACM Workshop on Cloud Computing Security Workshop, CCSW 2011, pp. 113–124. ACM, New York (2011)
25. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Gabow, H.N., Fagin, R. (eds.) 37th ACM STOC, pp. 84–93. ACM Press (2005)
26. Rivest, R.L., Adleman, L., Dertouzos, M.L.: On data banks and privacy homomorphisms. Found. Secure Comput. **4**, 169–179 (1978). Academia Press