Software Implementation of 2-Depth Pairing-based Homomorphic Encryption Scheme

Vincent Herbert[1⋆] and Caroline Fontaine[2]

[1] CEA LIST
[2] CNRS & Telecom Bretagne, Lab-STICC
vincent.herbert@cea.fr,caroline.fontaine@imt-atlantique.fr

Abstract. We propose a software implementation of a variant of Boneh-Goh-Nissim scheme [BGN05] with multiplicative depth 2, whereas the original one only tackled multiplicative depth 1. We employ together two improvements of the original scheme, based on [Fre10,CF15]. We give a full description of the resulting scheme, denoted BGN2, where encryption is performed bitwise. In this scheme, the homomorphic multiplication asks to compute pairings. We chose to compute an optimal Ate pairing over an elliptic curve in the Barreto-Naehrig curve family [BN05] using a library called DCLXVI [NNS10]. We provide simulation results, showing the interest of this solution for applications requiring a low multiplicative depth.

1 Introduction

While usual encryption schemes sometimes offer homomorphic properties, for addition [Pai99] or multiplication [RSA78] operations, they do not provide a way to perform both additions and multiplications at the same time. The only scheme based on classical cryptography that enables to handle an arbitrary number of additions and one single multiplication is [BGN05]. An important breakthrough has been made in 2009 according to the work of Gentry [Gen09b,Gen09a] who designed scheme that are able to perform unlimited additions and multiplications over encrypted data. Such schemes are called *Fully Homomorphic Encryption* (FHE) schemes. However, due to the size of operands, more practical ones, namely *Somewhat Homomorphic Encryption* (SHE) schemes, have been proposed in the steps of Aguilar *et al.* [AMGH10], to allow any number of additions but an (upper)-bounded number of multiplications, then drastically reducing the computation complexity. Because they allow arbitrary computations on encrypted data, (S/F)HE schemes suddenly opened the way to exciting new applications.(see *e.g.* [NLV11,GLN12,LLN14,BPB09]).

⋆ This work was done while the author was at CNRS, Lab-STICC.

A lot of schemes have been proposed in the literature, as for example [vDGHV10,SV10,GHS12,GH11,BGV12,CNT12,FV12,GSW13].

Such SHE schemes based on lattices present a large potential as they can handle several multiplications. But, in some use cases where the multiplicative depth of the circuit we want to evaluate over encrypted data is small, it may be of interest to closely look at lighter solutions like the one we present in this paper, which is based on an improvement of [BGN05], that we call BGN2, and which can handle a multiplicative depth of 2. This solution provides smaller ciphertexts than lattice based solutions, and its security is based on a hard problem that has been more deeply investigated. More precisely, it employs together two improvements of the original [BGN05] scheme, based on [Fre10,CF15]. To our knowledge, only Freeman's work has already been coupled with BGN in [Gui13] to greatly improve BGN's speed, and it is the first time that Catalone and Fiore's construction is applied to this particular setting in order to add one more multiplicative depth.

The order of ciphertext expansion in BGN2 is thousands rather than millions for SHE schemes based on lattices. This permits to make the first steps towards practical homomorphic cryptography. We supply running times of each homomorphic operation in our BGN2 implementation using OpenMP and 8 threads.

Organization. This paper is organized as follows. We will first present and describe our scheme in Section 2. Then, we will propose some examples of low depth Boolean circuits for which this scheme is helpful in Section 3. Performances of our scheme and its implementation will then be discussed in Section 4, as well as security issues. In Section 5 we finally draw some conclusions.

2 Cryptosystem description

2.1 Implementation settings

$x_0 = v^3$ and $v = 1868033$

$p(x) = 36x^4 + 36x^3 + 24x^2 + 6x + 1$

$r(x) = 36x^4 + 36x^3 + 18x^2 + 6x + 1$

$t(x) = 6x^2 + 1$

$p = p(x_0), r = r(x_0), t = t(x_0)$

The parameters p and r are 256-bits prime integers , the parameter t is a 128-bits integer. Let E be a curve of equation $y^2 = x^3 + 3$ defined over

$\mathbb{F}_p$. 12 is the embedding degree of r or the one of subgroup $E(\mathbb{F}_p)[r]$ of $E(\mathbb{F}_p)$. The operator $\xleftarrow{\$}$ refers to a random draw according to an uniform distribution.

$i_1, j_1, k_1, l_1, i_2, j_2, k_2, l_2 \xleftarrow{\$} \mathbb{F}_p : i_1 l_1 - j_1 k_1 = i_2 l_2 - j_2 k_2 = 1$

$g \xleftarrow{\$} E(\mathbb{F}_p)[r] : \mathrm{ord}(g) = r$

$$\begin{aligned} \pi \colon E &\to E \\ (x, y) &\mapsto (x^p, y^p) \end{aligned}$$

$h \xleftarrow{\$} E[r] \cap \mathrm{Ker}(\pi - p) : \mathrm{ord}(h) = r$

$u_1 \xleftarrow{\$} <(i_1 g, j_1 g)> \leq E(\mathbb{F}_p)[r]^2$

$v_1 \xleftarrow{\$} <(i_2 h, j_2 h)> \leq (E[r] \cap \mathrm{Ker}(\pi - p))^2$

$u = (u[0], u[1]) \xleftarrow{\$} E(\mathbb{F}_p)[r]^2$, $v = (v[0], v[1]) \xleftarrow{\$} (E[r] \cap \mathrm{Ker}(\pi - p))^2$

2.2 Encryption and decryption of a bit

$m \in \mathbb{F}_2$ $b \xleftarrow{\$} \mathbb{F}_2$

$a = m - b$

$c = \mathrm{Enc}(m) = (a, bu + u_1, bv + v_1) \in \mathbb{F}_2 \times E(\mathbb{F}_p)[r]^2 \times (E[r] \cap \mathrm{Ker}(\pi - p))^2$

To evaluate with this ciphertext, we need the first and at least one of the two last components. According to this, we speak of curve mode, twist mode or mixed mode. We decrypt with the two remaining components.

$\pi_1 \in \mathrm{End}(E(\mathbb{F}_p)[r]^2), \pi_2 \in \mathrm{End}(\mathrm{Ker}(\pi - p)^2)$

$\pi_1(x, y) = (-j_1 k_1 x + i_1 k_1 y, -j_1 l_1 x + i_1 l_1 y)$

$\pi_2(x, y) = (-j_2 k_2 x + i_2 k_2 y, -j_2 l_2 x + i_2 l_2 y)$

$m = \mathrm{Dec}(c) = a + \dfrac{\pi_1(bu + u_1)}{\pi_1(u)}$ [3] if $c \in \mathbb{F}_2 \times E(\mathbb{F}_p)[r]^2$

$m = \mathrm{Dec}(c) = a + \dfrac{\pi_2(bv + v_1)}{\pi_2(v)}$ if $c \in \mathbb{F}_2 \times (E[r] \cap \mathrm{Ker}(\pi - p))^2$

The encryption function takes a bit on input. It outputs :

- an element of $\mathbb{F}_2$,
- two group elements in $E(\mathbb{F}_p)[r]$ of order r, that is four elements of $\mathbb{F}_p$,
- and two group elements in $E[r] \cap \mathrm{Ker}(\pi - p)$ of order r, that is four elements of $\mathbb{F}_{p^{12}} \cong \mathbb{F}_p[X]/(X^{12} + 3)$.

[3] The ratio is well-defined if the numerator is a multiple of the denumerator. Its value is the scalar factor between the two points, modulo 2. This is the case here since u_1 and v_1 belongs respectively to the kernel of π_1 and π_2[Fre10, page 58].

We simplify groups expressions. We have $E(\mathbb{F}_p)[r] = E(\mathbb{F}_p)$ since we have $\#E(\mathbb{F}_p) = r$ with r prime. We also have $E[r] \cap \mathrm{Ker}(\pi - p) = E(\mathbb{F}_{p^{12}})[r]$. $E(\mathbb{F}_p)[r]$ and $E[r] \cap \mathrm{Ker}(\pi - p)$ are commutative groups of order r. They are isomorph to $(\mathbb{Z}/r\mathbb{Z}, +)$. Two elements of $E(\mathbb{F}_p)[r]$ can be represented by two elements of $\mathbb{F}_p$ plus 2 bits (x-coordinates and y-signs), that is $2\log_2(p) + 2 = 514$ bits. The curve E admits a twist of degree 6. With twist, elements of $\mathbb{F}_{p^{12}}$ can be represented using elements of $\mathbb{F}_{p^2}$. Two elements of $E[r] \cap \mathrm{Ker}(\pi - p)$ can be represented by two elements of $\mathbb{F}_p^2$ plus 2 bits, that is $4\log_2(p) + 2 = 1026$ bits. We employ $E' : y^2 = x^3 + 3/\xi$, a sextic twist of E defined over $\mathbb{F}_p^2$, with ξ chosen such that $r \mid \#E'(\mathbb{F}_p^2)$.

We reformulate. The encryption of a bit consists in:

- an element of $\mathbb{F}_2$,
- two group elements in $E(\mathbb{F}_p)$
- two group elements in $E'(\mathbb{F}_p^2)$

2.3 Ciphertext level

Figure 1 indicates which operations are permitted on ciphertexts in BGN2. This correlates with the ciphertext level. A ciphertext, on which no homomorphic operation has been made, has level 1. An homomorphic multiplication produce a ciphertext of level > 1. Table 1 sums up the different ciphertext spaces according to this notion of ciphertext level as well as circuit multiplicative depth.

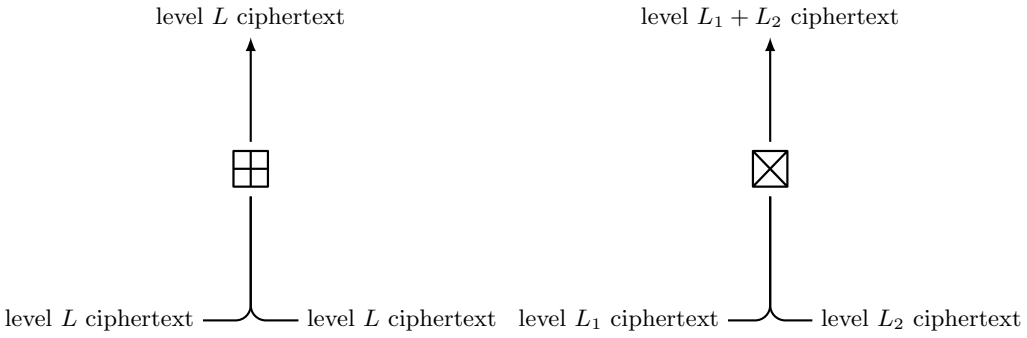

Fig. 1. Operations in BGN2 on ciphertexts in the same space, . $L \leq 4$,$L_1 \leq 2$,$L_2 \leq 2$.

Circuit multiplicative depth	Ciphertext level	Ciphertext space
0^4	1	$\mathbb{F}_2 \times E(\mathbb{F}_p)[r]^2$ $\mathbb{F}_2 \times (E[r] \cap \mathrm{Ker}(\pi - p))^2$ $\mathbb{F}_2 \times E(\mathbb{F}_p)[r]^2 \times (E[r] \cap \mathrm{Ker}(\pi - p))^2$
1	2	$\mathbb{F}_2 \times \mu_r^4$
2^5	3	$\mu_r^4 \times (E(\mathbb{F}_p)[r]^2 \times \mu_r^4)^B$ $\mu_r^4 \times ((E[r] \cap \mathrm{Ker}(\pi - p))^2 \times \mu_r^4)^B$ $\mu_r^4 \times (\mu_r^4 \times E(\mathbb{F}_p)[r]^2)^B$ $\mu_r^4 \times (\mu_r^4 \times (E[r] \cap \mathrm{Ker}(\pi - p))^2)^B$
	4	$\mu_r^4 \times (\mu_r^4 \times \mu_r^4)^B$

Table 1. The different ciphertext spaces according to circuit multiplicative depth.

2.4 Multiply level 1 ciphertexts

A level 1 ciphertext $\in \mathbb{F}_2 \times E(\mathbb{F}_p)[r]^2 \times (E[r] \cap \mathrm{Ker}(\pi - p))^2$. To obtain the product of two level 1 ciphertexts, we remove the second component of first ciphertext and the third component of second ciphertext. We proceed in this way because we use an asymmetric pairing. The result is a level 2 ciphertext.

$c_1 = (a_1, \beta_1) \in \mathbb{F}_2 \times E(\mathbb{F}_p)[r]^2, c_2 = (a_2, \beta_2) \in \mathbb{F}_2 \times (E[r] \cap \mathrm{Ker}(\pi - p))^2$.
$\mathrm{Mult}^{(2)}(c_1, c_2) \coloneqq c = (a, \beta) \in \mathbb{F}_2 \times \mu_r^4$
$b_1, b_2, s \xleftarrow{\$} \mathbb{F}_2$
For $i = 1$ and $i = 2$, c_i is an encryption of $m_i \in \mathbb{F}_2$ and $a_i = m_i - b_i$.
$a = a_1 a_2 - s$
We redo a random uniform draw for u_1 and v_1.

$$u_1 \xleftarrow{\$} <(i_1 g, j_1 g)>,\ v_1 \xleftarrow{\$} <(i_2 h, j_2 h)>$$

We split up the computation of β in order to explain how the formula is obtained. We can jump this paragraph and only retain the final formula for a practical usage. The operator $\oplus$ refers to an homomorphic addition with the scheme BGN-F[6]. Let e_{OA} be the optimal Ate pairing. The notation, $\mathrm{Enc}^{(L)}(s)$ refers to a level L ciphertext of bit s, with the

[4] Depending on the circuit, we can omit, or not, curvepoints or twistpoints computation.
[5] $B = A + 1$. A is the number of additions of level 3 or 4 performed to obtain the ciphertext.
[6] The addition operator, in the scheme BGN2, is denoted $\boxplus$

scheme BGN-F. If no level is indicated, *e.g.* $\mathrm{Enc}(s)$, we consider a level 1 ciphertext.

$$e\colon E(\mathbb{F}_p)[r]^2 \times (E[r] \cap \mathrm{Ker}(\pi - p))^2 \to \mu_r^4$$
$$e((g_1, g_2)(h_1, h_2)) \mapsto (e_{OA}(g_1, h_1), e_{OA}(g_1, h_2), e_{OA}(g_2, h_1), e_{OA}(g_2, h_2))$$

$$\beta = e(\beta_1, \beta_2)e(u, v_1)e(u_1, v) \oplus a_1\beta_2 \oplus a_2\beta_1 \oplus \mathrm{Enc}^{(2)}(s)$$

Let μ_r be the subgroup of r^{th}-roots of unity in $\mathbb{F}_{p^{12}}$. The first term belongs to μ_r^4. Note, the level should be the same for all terms [7].

$u_2, u_3, u_4 \xleftarrow{\$} <(i_1 g, j_1 g)>$

$v_2, v_3, v_4 \xleftarrow{\$} <(i_2 h, j_2 h)>$

$\beta = e(\beta_1, \beta_2)e(u, v_1)e(u_1, v) \oplus e(\mathrm{Enc}(1), a_1\beta_2)e(u, v_2)e(u_2, v) \oplus e(a_2\beta_1, \mathrm{Enc}(1))e(u, v_3)e(u_3, v) \oplus e(\mathrm{Enc}(1), \mathrm{Enc}(s))e(u, v_4)e(u_4, v)$

Using bilinearity, we can simplify this expression. In practice, it is not useful to define $u_2, u_3, u_4, v_2, v_3, v_4$. On the other hand, it is useful to understand how we obtain the following formula.

$$\beta = e(\beta_1, \beta_2)e(\mathrm{Enc}(1), a_1\beta_2 + \mathrm{Enc}(s))e(a_2\beta_1, \mathrm{Enc}(1))e(u, v_1)e(u_1, v)$$

We compute 5×4 pairings to get a level 2 ciphertext.

2.5 Add level L ciphertexts with $1 \leq L \leq 2$

On input, there are two ciphertexts (a_1, β_1) and (a_2, β_2), with the same level $1 \leq L \leq 2$ and $a_1, a_2 \in \mathbb{F}_2$. On output, there is one level L ciphertext (a, β) The three ciphertexts are in the same space.

Three configurations are possible.

- $\beta_1, \beta_2 \in E(\mathbb{F}_p)[r]^2$
- $\beta_1, \beta_2 \in (E[r] \cap \mathrm{Ker}(\pi - p))^2$
- $\beta_1, \beta_2 \in \mu_r^4$

$$a = a_1 + a_2$$

[7] If it is not the case, we multiply homomorphically the other terms by $\mathrm{Enc}(1)$, an encryption of bit 1. More generally, this is applied several times when we compute the sum of ciphertexts with several levels of difference.

We redo a random uniform draw for u_1 and v_1.

$$u_1 \xleftarrow{\$} <(i_1 g, j_1 g)>,\ v_1 \xleftarrow{\$} <(i_2 h, j_2 h)>$$

$$\beta = \beta_1 + \beta_2 + u_1 \text{ if } \beta_1, \beta_2 \in E(\mathbb{F}_p)[r]^2$$

$$\beta = \beta_1 + \beta_2 + v_1 \text{ if } \beta_1, \beta_2 \in (E[r] \cap \mathrm{Ker}(\pi - p))^2$$

$$\beta = \beta_1 \beta_2 e(u, v_1) e(u_1, v) \text{ if } \beta_1, \beta_2 \in \mu_r^4$$

2.6 Decrypt level 2 ciphertext

We define a notation used by Freeman, more compact than the usual one.

Let $\mathcal{M} = (m_{i,j})$ be an n-order matrix over $\mathbb{F}_p$

$\gamma^{\mathcal{M}} \coloneqq (\prod_{i=1}^n \gamma_i^{m_{i1}}, \ldots, \prod_{i=1}^n \gamma_i^{m_{in}})$ with γ in a product group.

$\mathcal{A} = \begin{pmatrix} -j_1 k_1 & -j_1 l_1 \\ i_1 k_1 & i_1 l_1 \end{pmatrix},\ \mathcal{B} = \begin{pmatrix} -j_2 k_2 & -j_2 l_2 \\ i_2 k_2 & i_2 l_2 \end{pmatrix}$

$\mathcal{A} \otimes \mathcal{B}$ is a matrix of order 4. We can divide it into 4 matrices of order 2. The $(i,j)^{\text{th}}$ block is equal to $a_{i,j}\mathcal{B}$ with $\mathcal{A} = (a_{i,j})_{i,j \in \{1,2\}}$.

$\pi_T \in \mathrm{End}(\mu_r^4)$

$\pi_T(\beta) = (\beta_1, \beta_2, \beta_3, \beta_4)^{\mathcal{A} \otimes \mathcal{B}}$

$m = \mathrm{Dec}(c) = a + \log_{\pi_T(e(u,v))}(\pi_T(\beta))$ if $c \in \mathbb{F}_2 \times \mu_r^4$

Public-key and private-key At this stage, we have used all the material needed to encrypt and decrypt. We can explicit the keys in BGN2 scheme.

- Public-key is $((E(\mathbb{F}_p)[r])^2, (i_1 g, j_1 g), (E[r] \cap \mathrm{Ker}(\pi - p))^2, (i_2 h, j_2 h), \mu_r, e, u, v)$.
- Private-key is (π_1, π_2, π_T).

2.7 Multiply ciphertexts to obtain a level L ciphertext with $3 \leq L \leq 4$

On input, there are two ciphertexts (a_1, β_1) and (a_2, β_2), with levels $L_1, L_2 \in [\![1, 2]\!]$, $3 \leq L_1 + L_2 \leq 4$ and $a_1, a_2 \in \mathbb{F}_2$. On output, there is one level L ciphertext (α, β) with $L = L_1 + L_2$[8]. As previously said, $\oplus$ refers to an homomorphic addition with the scheme BGN-F.

[8] We can obtain level 4 ciphertexts but no product between a level 1 ciphertext and a level 3 ciphertext is defined.

$$\alpha = \mathrm{Enc}(a_1 a_2) \oplus \beta_2^{a_1} \oplus \beta_1^{a_2} \quad [9]$$

$$\beta = (\beta_1, \beta_2)$$

We can only add ciphertexts of same level, see Section 2.5. To compute α, we should get level 2 terms.

Three configurations are possible.

- $\beta \in (E(\mathbb{F}_p)[r]^2 \times \mu_r^4) \cup ((E[r] \cap \mathrm{Ker}(\pi - p))^2 \times \mu_r^4)$
- $\beta \in (\mu_r^4 \times E(\mathbb{F}_p)[r]^2) \cup (\mu_r^4 \times (E[r] \cap \mathrm{Ker}(\pi - p))^2)$
- $\beta \in \mu_r^4 \times \mu_r^4$

The three corresponding values of α are:

- $e(\mathrm{Enc}(a_1 a_2), \mathrm{Enc}(1))\beta_2^{a_1} e(a_2\beta_1, \mathrm{Enc}(1))e(u, v_1)e(u_1, v)$
- $e(\mathrm{Enc}(a_1 a_2), \mathrm{Enc}(1))\beta_1^{a_2} e(a_1\beta_2, \mathrm{Enc}(1))e(u, v_1)e(u_1, v)$
- $e(\mathrm{Enc}(a_1 a_2), \mathrm{Enc}(1))\beta_2^{a_1}\beta_1^{a_2} e(u, v_1)e(u_1, v)$

The first two cases permit to evaluate the same products since the multiplication is commutative over $\mathbb{F}_2$. We choose to limit outself to the first case where the first ciphertext has level 1, and the second ciphertext has level 2. Once again, in the first case, we restrict, for convenience, β in the product group $(E(\mathbb{F}_p)[r]^2 \times \mu_r^4)$. In every instance, $\alpha \in \mu_r^4$. The number of successive multiplications is limited to one because the scheme BGN2 evaluates ciphertexts up to level 4. Notice, the computation of a level 3 ciphertext needs the computation of 4×4 pairings instead of 3×4 pairings for the computation of a level 4 ciphertext. The additional pairings are an extra part of the computation cost when we multiply two ciphertexts of different levels.

2.8 Add level L ciphertexts with $3 \leq L \leq 4$

On input, there are two level L ciphertexts (α_1, β_1) and (α_2, β_2), with $L \in [\![3, 4]\!]$. The two ciphertexts are in the same ambient space. On output, there is one level L ciphertext (α, β).

$$\alpha = \alpha_1 \oplus \alpha_2 = \alpha_1\alpha_2 e(u, v_1)e(u_1, v)$$

For every instance, $\alpha, \alpha_1, \alpha_2 \in \mu_r^4$.

[9] Abuse of notation in this subsection. Exponentations should be replaced by multiplications, for level 1 ciphertexts, where it operates on additive groups.

$$\beta = (\beta_1, \beta_2)$$

Each addition and multiplication (see Section 2.7) to obtain a level L ciphertext, extend the ciphertext size. For this reason, we operate a limited number of such additions in practice.

For a ciphertext of level $3 \leq L \leq 4$, obtained after A additions of level $3 \leq L \leq 4$, there are three cases:

- $\beta \in (E(\mathbb{F}_p)[r]^2 \times \mu_r^4)^B \cup ((E[r] \cap \mathrm{Ker}(\pi - p))^2 \times \mu_r^4)^B$
- $\beta \in (\mu_r^4 \times E(\mathbb{F}_p)[r]^2)^B \cup (\mu_r^4 \times (E[r] \cap \mathrm{Ker}(\pi - p))^2)^B$
- $\beta \in (\mu_r^4 \times \mu_r^4)^B$

where $B = A + 1$.

2.9 Decrypt level L ciphertext with $3 \leq L \leq 4$

On input a ciphertext (α, β) obtained with A additions of different level L ciphertexts with $3 \leq L \leq 4$. α is a level 2 ciphertext.

$$\beta \coloneqq (\beta_{1,1}, \beta_{2,1}, \beta_{1,2}, \beta_{2,2}, \ldots, \beta_{1,B}, \beta_{2,B})$$

where $\forall\ i, j \in [\![1, B]\!]$, $\beta_{i,j}$ is either a level 1 ciphertext or a level 2 ciphertext.

On output a plaintext $m \in \mathbb{F}_2$.

$$m = \mathrm{Dec}(\alpha) + \sum_{i=1}^{B} \mathrm{Dec}(\beta_{1,i})\,\mathrm{Dec}(\beta_{2,i})$$

3 Low multiplicative depth Boolean circuits

We treat bit per bit encryption. Circuits are rewritten with two operators: $\veebar$ (exclusive disjunction) and $\wedge$. They can be written under different forms depending on operation order.

3.1 Binary data

Table 2 supplies constrainsts on ciphertext levels with typical examples of low-depth multiplicative circuits.

Notation: ciphertexts a (resp. b, c, d) with level L_1 (resp. level L_2, L_3, L_4), ciphertexts x (resp. y) with level M_1 (resp. level M_2).

Functions	Boolean function[10]	Multiplicative Depth	Inputs levels	Output level
Test $a == 1$	a	0	$\in [\![1, 4]\!]$	L_1
REFRESH a	$a \veebar 0$			
Test $a == 0$, NOT a	$a \veebar 1$			
Test $a \neq b$, a XOR b, sum of bits	$(a \veebar b)$			$\max(L_1, L_2)$
Test $a == b$, a XNOR b	$(a \veebar b) \veebar 1$			
a AND b, product of bits	$a \wedge b$	1	$\in [\![1, 2]\!]$	$L_1 + L_2$
a OR b	$((a \wedge b) \veebar a) \veebar b$			
2-to-1 MUX, if x then a else b	$((a \veebar b) \wedge x) \veebar b$			$\max(L_1, L_2) + M_1$
4-to-1 MUX selector inputs: x and y output $\in \{a, b, c, d\}$	$(a \wedge x \wedge y)$ $\veebar(b \wedge x \wedge (y \veebar 1))$ $\wedge(x \veebar 1) \wedge y)$ $\veebar(d \wedge (x \veebar 1) \wedge (y \veebar 1)$	2	1	$\max(L_1, L_2, L_3, L_4) + M_1 + M_2$

Table 2. Constrainsts on ciphertext levels with some low-depth multiplicative circuits

In BGN2, no multiplication is defined with a factor having a level ≥ 3. Therefore, operands level is limited according to multiplicative depth.

3.2 Integer data

Data are n-bits integers. Input ciphertexts have level $L = 1$.

- Adder $a_{n-1} \dots a_0 + b_{n-1} \dots b_0 \mod 2^n$
  In terms of multiplicative depth, the hardest part is the evaluation of the carry which enables to compute the most significant bit (MSB) of the sum modulo 2^n. Let us compute the MSB with $n = 3$, which is the maximal value for BGN2. Then, it can be written :

$$a_2 \veebar b_2 \veebar ((a_1 \wedge b_1) \wedge (a_0 \wedge b_0))$$

  The Boolean circuit has multiplicative depth $\lceil \log_2 2(n-1) \rceil = 2$. The corresponding ciphertext is the sum of three ciphertexts of level

[10] A Boolean function is not associated to a unique Boolean circuit. We restrict to circuits with $\veebar$ and $\wedge$ gates because these gates correspond to elementary homomorphic operations in BGN2. The parentheses indicates the order of operations. It permits to define a unique Boolean circuit and then to indicate its multiplicative depth.

$2(n-1)L = 4$. Indeed, to add ciphertexts, we need to have ciphertexts of the same level. In this case a_2 and b_2 have level L but $((a_1 \wedge b_1) \wedge (a_0 \wedge b_0))$ has level 4. To increment the level of a ciphertext, we multiply it homomorphically by an encryption of 1.

– Test $a_{n-1} \dots a_0 == b_{n-1} \dots b_0$
With BGN2, we can manage up to $n = 4$ bits with the circuit of depth $\lceil log_2(n) \rceil = 2$:

$$((a_3 \veebar b_3) \veebar 1) \wedge ((a_2 \veebar b_2) \veebar 1) \wedge ((a_1 \veebar b_1) \veebar 1) \wedge ((a_0 \veebar b_0) \veebar 1)$$

The output is a ciphertext of level $nL = 4$.

4 Implementation performances and security

Our implementation uses the DCLXVI library [NNS10] to manage pairing in an efficient way. We used version 20130329 of the library, which has been designed to target a 128-bits security level, with software speed-up. The rest of the implementation is of our own.

4.1 Memory usage and running time

Figure 2 show how data size in BGN2 is dependent on two kind of parameters. Those related with functionality (treatments operated on ciphertexts) and those linked with security (implementation choices on cryptosystem settings). Time/space/communication cost depend on:

– operations (encryption scheme, level of ciphertexts)
– operand size (security assumption, security level)
– operation number (Boolean circuit)

Homomorphic operations can differ according to the ciphertext level. A fresh ciphertext is level 1. Multiply level L_1 and level L_2 ciphertexts give a level $L_1 + L_2$ ciphertext.

Ciphertext size and key size are indicated in Table 3 and in Table 4. Size depend on group order and data representation. In our implementation, we employ 256-bits prime p and r (see Subsection 2.1) to define groups.

In some configurations, few errors (such as false positives in a test) can be acceptable. In this case, we can restrain constraints and design an

[11] Depending on the circuit, we encrypt plaintexts in curve mode, twist mode, or both.
[12] The symbol k (resp. l) is the number of level 3 (resp. 4) additions performed to obtain the ciphertext.

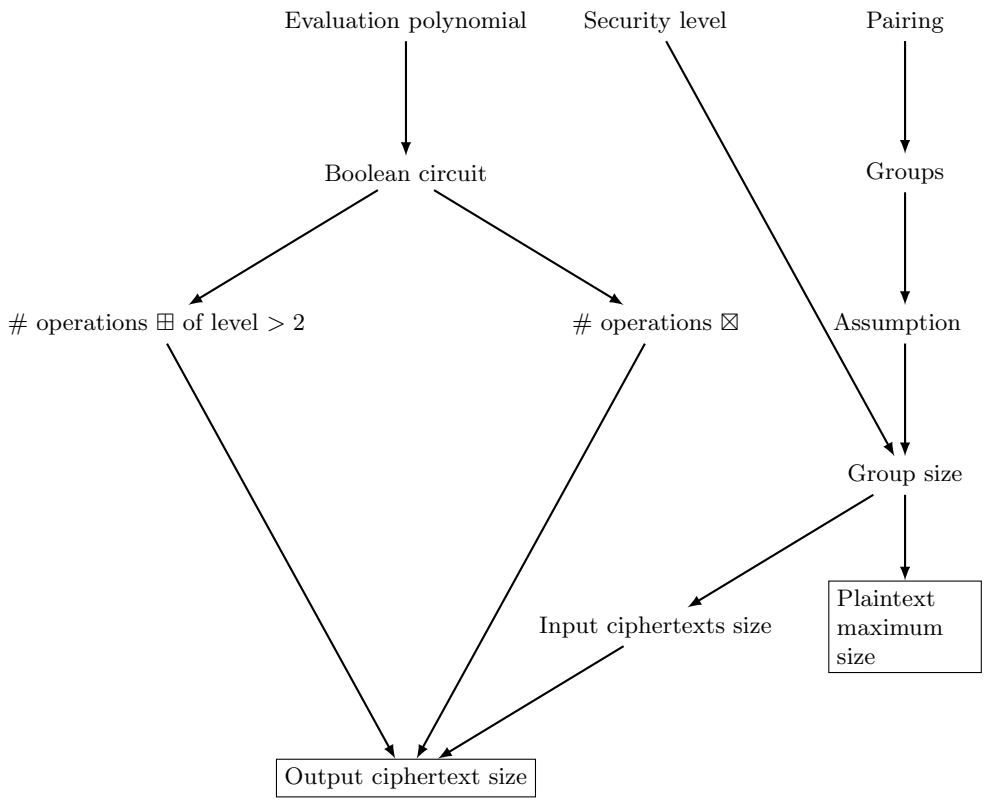

Fig. 2. Parameters influencing memory usage in BGN2 scheme.

Ciphertext level	Size in bytes
1 (curve mode[11])	784
1 (twist mode)	1552
2	4624
3	$10032 + k^{12} \times 5376$
4	$13872 + l \times 9216$

Table 3. Ciphertext size in our BGN2 implementation (256-bits primes p and r)

ad-hoc circuit with a lower depth than an error-free circuit. Cheap tasks could be done with precomputation and postcomputation. It can permit to decrease both the multiplicative depth and the number of additions of level > 2.

	Size in bytes	Short description
Public key	4608	four curve points, four twist points
Private key	1024	eight prime field elements

Table 4. Key size in our BGN2 implementation (256-bits primes p and r)

We provide in Table 5 the running time of different homomorphic cryptographic operations. In BGN2, it does not only depend on the operation (encryption decryption, addition, multiplication). Addition of ciphertexts of level >2 and any multiplication of ciphertexts modify ciphertext spaces (see Table 1). It is thus necessary to specify time for each ciphertext level.

Operation	Time
Encryption (curve mode)	1.37 ms
Encryption (twist mode)	0.878 ms
Multiplication $L1^{13}$	5.27 ms
Multiplication $L1L2$	4.3 ms
Multiplication $L2$	3.49 ms
Addition $L1$ (curve mode)	0.643 ms
Addition $L1$ (twist mode)	0.606 ms
Addition $L2$	2.37 ms
Addition of two $L3$	2.25 ms
Addition of two $L4$	2.18 ms
Decryption (curve mode)	1.09 ms
Decryption (twist mode)	0.851 ms
Decryption $L2$	18.9 ms
Decryption $L3$	39.7 ms
Decryption $L4$	57.9 ms
Decryption sum of two $L3$	60.9 ms
Decryption sum of two $L4$	98.6 ms

Table 5. Running time of operations in BGN2 scheme on a Dell Precision T7810, using two E5-2623v3 chips (each of them has 4 cores and can manage 8 threads, at a frequency of 3GHz). In our experiments we used 16 threads at a time.

4.2 Security

The security of BGN2 is based on the generalized subgroup decision assumption. We chose to employ asymetric pairings to compute homomomorphic product of fresh ciphertexts. The use of symmetric pairings would change the computational hardness assumption [Fre10]. This problem is derived from the decision Diffie-Hellman assumption [Bon98]. Two possible choices to instantiate groups are to select either an elliptic curve or an hyperelliptic curve. In the first case, the security assumption reduces to the elliptic curve discrete logarithm problem and the recommended group size is given by different academic and private organizations at www.keylength.com according to standard security levels.

We would like to point out that our implementation with 256-bits primes p and r ensures today around 110-bits security and not 128-bits security, as it was targetted a few months ago. Indeed, pairing-based cryptography using target field $\mathbb{F}_{p^6}$ and $\mathbb{F}_{p^{12}}$ is affected by Kim-Barbulescu variant of the Number Field Sieve[KB15]. This forces to reevaluate parameters size if we want to maintain a 128-bits security level. According to estimates, last month, in [MSS16], 383-bits primes are now required for a 128-bits security level. As our implementation relies on DCLXVI library, we need to modify it to update parameters size. But this step is not trivial, as DCLXVI has been conceived to optimize software speed rather than scalability. Hence, we will address this issue in a future publication.

5 Conclusion

In this paper, we proposed a variant of BGN homomorphic encryption scheme that is called BGN2 and can address one more multiplicative depth. This scheme may help to address practical situations where the multiplicative depth is of 2, with smaller keys and ciphertext expansion than homomorphic encryption schemes based on lattices. Moreover, its security is better understood than for lattices based schemes, as it relies on Discrete Logarithm computation, which has been more deeply studied than, for example, LWE or RLWE. It is also less complex, as no bootstrapping nor relinearization is needed here.

References

AMGH10. Carlos Aguilar-Melchor, Philippe Gaborit, and Javier Herranz. Additively homomorphic encryption with d-operand multiplications. In *Advances in Cryptology–CRYPTO 2010*, pages 138–154. Springer, 2010.

[13] Multiplication L1 means multiplication of two level one ciphertexts.

BGN05. Dan Boneh, Eu-Jin Goh, and Kobbi Nissim. Evaluating 2-DNF Formulas on Ciphertexts. In Joe Kilian, editor, *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*, pages 325–341. Springer, 2005.

BGV12. Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 309–325. ACM, 2012.

BN05. Paulo S. L. M. Barreto and Michael Naehrig. Pairing-Friendly Elliptic Curves of Prime Order. In Bart Preneel and Stafford E. Tavares, editors, *Selected Areas in Cryptography, 12th International Workshop, SAC 2005, Kingston, ON, Canada, August 11-12, 2005, Revised Selected Papers*, volume 3897 of *Lecture Notes in Computer Science*, pages 319–331. Springer, 2005.

Bon98. Dan Boneh. The Decision Diffie-Hellman Problem. In *Proceedings of the Third International Symposium on Algorithmic Number Theory*, ANTS-III, pages 48–63, London, UK, UK, 1998. Springer-Verlag.

BPB09. Tiziano Bianchi, Alessandro Piva, and Mauro Barni. On the implementation of the discrete Fourier transform in the encrypted domain. *IEEE Transactions on Information Forensics and Security*, 4(1):86–97, 2009.

CF15. Dario Catalano and Dario Fiore. Using Linearly-Homomorphic Encryption to Evaluate Degree-2 Functions on Encrypted Data. In Indrajit Ray, Ninghui Li, and Christopher Kruegel, editors, *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-6, 2015*, pages 1518–1529. ACM, 2015.

CNT12. Jean-Sébastien Coron, David Naccache, and Mehdi Tibouchi. Public key compression and modulus switching for fully homomorphic encryption over the integers. In *Advances in Cryptology–EUROCRYPT 2012*, pages 446–464. Springer, 2012.

Fre10. David Mandell Freeman. Converting Pairing-Based Cryptosystems from Composite-Order Groups to Prime-Order Groups. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 44–61. Springer, 2010.

FV12. Junfeng Fan and Frederik Vercauteren. Somewhat Practical Fully Homomorphic Encryption. *IACR Cryptology ePrint Archive*, 2012:144, 2012.

Gen09a. Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009.

Gen09b. Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, volume 9, pages 169–178, 2009.

GH11. Craig Gentry and Shai Halevi. Fully homomorphic encryption without squashing using depth-3 arithmetic circuits. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 107–109. IEEE, 2011.

GHS12. Craig Gentry, Shai Halevi, and Nigel P Smart. Fully homomorphic encryption with polylog overhead. In *Advances in Cryptology–EUROCRYPT 2012*, pages 465–482. Springer, 2012.

GLN12. Thore Graepel, Kristin E. Lauter, and Michael Naehrig. ML Confidential: Machine Learning on Encrypted Data. In *ICISC*, volume 7839 of *LNCS*, pages 1–21. Springer, 2012.

GSW13. Craig Gentry, Amit Sahai, and Brent Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In *Advances in Cryptology–CRYPTO 2013*, pages 75–92. Springer, 2013.

Gui13. Aurore Guillevic. Comparing the pairing efficiency over composite-order and prime-order elliptic curves. In *Applied Cryptography and Network Security - 11th International Conference, ACNS 2013, Banff, AB, Canada, June 25-28, 2013. Proceedings*, pages 357–372, 2013.

KB15. Taechan Kim and Razvan Barbulescu. Extended Tower Number Field Sieve: A New Complexity for Medium Prime Case. Cryptology ePrint Archive, Report 2015/1027, 2015.

LLN14. Kristin Lauter, Adriana López-Alt, and Michael Naehrig. Private Computation on Encrypted Genomic Data. In *LATINCRYPT*, LNCS, 2014.

MSS16. Alfred Menezes, Palash Sarkar, and Shashank Singh. Challenges with assessing the impact of nfs advances on the security of pairing-based cryptography. Cryptology ePrint Archive, Report 2016/1102, 2016. `http://eprint.iacr.org/2016/1102`.

NLV11. Michael Naehrig, Kristin E. Lauter, and Vinod Vaikuntanathan. Can homomorphic encryption be practical? In *ACM CCSW*, pages 113–124. ACM, 2011.

NNS10. Michael Naehrig, Ruben Niederhagen, and Peter Schwabe. New Software Speed Records for Cryptographic Pairings. In Michel Abdalla and Paulo S. L. M. Barreto, editors, *Progress in Cryptology - LATINCRYPT 2010, First International Conference on Cryptology and Information Security in Latin America, Puebla, Mexico, August 8-11, 2010, Proceedings*, volume 6212 of *Lecture Notes in Computer Science*, pages 109–123. Springer, 2010.

Pai99. Pascal Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *Proc. of Advances in Cryptology — EUROCRYPT 1999*, number 1592 in LNCS, pages 223–238, 1999.

RSA78. Ronald Linn Rivest, Adi Shamir, and Leonard Max Adleman. A Method for Obtaining Digital Signatures and Public-key Cryptosystems. *Commun. ACM*, 21(2):120–126, February 1978.

SV10. Nigel P Smart and Frederik Vercauteren. Fully homomorphic encryption with relatively small key and ciphertext sizes. In *International Workshop on Public Key Cryptography*, pages 420–443. Springer, 2010.

vDGHV10. Marten van Dijk, Craig Gentry, Shai Halevi, and Vinod Vaikuntanathan. Fully homomorphic encryption over the integers. In *Advances in cryptology–EUROCRYPT 2010*, pages 24–43. Springer, 2010.