Improved Efficiency of a Linearly Homomorphic Cryptosystem

Parthasarathi Das[1(✉)], Michael J. Jacobson Jr.[1], and Renate Scheidler[2]

[1] Department of Computer Science, University of Calgary,
2500 University Drive NW, Calgary, AB T2N 1N4, Canada
{parthasarathi.das,jacobs}@ucalgary.ca
[2] Department of Mathematics and Statistics, University of Calgary,
2500 University Drive NW, Calgary, AB T2N 1N4, Canada
rscheidl@ucalgary.ca

Abstract. We present an extended version of the Castagnos and Laguillaumie linearly homomorphic cryptosystem [5] in which the non-maximal imaginary quadratic order is allowed to have conductor equal to a product of prime powers as opposed to a single prime. Numerical results obtained with an optimized C implementation demonstrate that this variation improves performance when large messages and exponents are used. When compared to the cryptosystems of Paillier [11] and Bresson et al. [3] at the same security levels, the basic version of Castagnos and Laguillaumie is the fastest at high security levels for small messages.

Keywords: Linearly homomorphic encryption · Public key cryptography · Ideal class group · Electronic voting · Encryption switching protocol

1 Introduction

A *linearly homomorphic cryptosystem* is one for which linear combinations of ciphertexts can be computed in such a way that the result is the encryption of the same linear combination of the corresponding plaintexts. Such cryptosystems have a number of applications. For example, when used for electronic voting, encrypted votes (encrypting 1 for "yes" and 0 for "no") can be tallied with a single decryption by homomorphically adding the ciphertexts and decrypting the result. Two well-known examples of linearly homomorphic encryption systems are due to Paillier [11] and Bresson et al. [3]. In both cases, the security relies on the presumed intractability of integer factorization.

In [5], Castagnos and Laguillaumie presented a linearly homomorphic encryption scheme whose security is based on the hardness of the decision Diffie-Hellman (DDH) problem in a group that has a subgroup in which the discrete logarithm (DL) problem can be solved easily; this setting is referred to as a "DDH group with an easy DL subgroup". Assuming the existence of such groups, they

The second and third authors' research is supported by NSERC.

C. Carlet et al. (Eds.): C2SI 2019, LNCS 11445, pp. 349–368, 2019.
https://doi.org/10.1007/978-3-030-16458-4_20

described a linearly homomorphic encryption scheme that is provably one-way and semantically secure subject to relatively standard hardness assumptions. They also gave an instantiation of their cryptosystem using the ideal class group of a non-maximal imaginary quadratic order with prime conductor as the DDH group with easy DL subgroup. Subsequently, this cryptosystem was used in combination with a variant of ElGamal in an encryption switching protocol [4], providing an efficient setting for a secure two-party computation protocol.

The cryptosystem of [5] has two main novel features. Firstly, it is the only purely linearly homomorphic cryptosystem (not counting the fully homomorphic cryptosystems based on the learning with errors problem) whose security does not depend on integer factorization—all hardness assumptions are versions of Diffie-Hellman and discrete logarithm problems. The second feature is that the size of the message space can be chosen independently of the security parameter. This is especially attractive in electronic voting applications, as the message space can be chosen just large enough to handle the required number of votes. In contrast, [11] and [3] are both defined in terms of RSA moduli, and the number of messages that can be encrypted is of the same size as the modulus. When appropriate security levels are used, these allow far more messages than necessary for typical voting scenarios.

Castagnos and Laguillaumie [5] also presented numerical results using an implementation of their cryptosystem, which suggested that it has advantages over Pailler and Bresson et al. at the 112- and 128-bit security levels. However, the implementation was done using a general-purpose computer algebra system as opposed to a more specialized and optimized implementation. In addition, two possible improvements were suggested, designed to allow larger messages without increasing the security level, and to speed up decryption via the Chinese Remainder Theorem. These improvements arise from using conductors that are prime powers and products of distinct primes, respectively, as opposed to primes. Exploring both these ideas was left as future work.

In this paper, we fully explore the efficiency of the cryptosystem of Castagnos and Laguillaumie [5]. Our first contribution is a complete description of the cryptosystem using conductors that are products of prime powers, thereby covering both the suggested improvements in [5]. We present a detailed benchmarking of the cryptosystem at the 128-, 192-, and 256-bit security levels, and compare its performance to both the Pailler [11] and the Bresson et al. [3] cryptosystems. Our implementation makes use of a state-of-the-art C implementation of class group arithmetic in imaginary quadratic orders due to Sayles [12]. We use both the original version of [5] where group elements are sampled from the entire group, as well as standard short exponent versions that also have provable security properties but under variations of the intractability assumptions that are restricted to short exponents, based on the results of Koshiba and Kurosawa [10]. The variations of Castagnos and Laguillaumie considered here offer performance improvements when using large exponents and large messages. When compared to the cryptosystems of Paillier [11] and Bresson et al. [3] at the same security levels, our results show that the basic version of Castagnos and Laguillaumie is the fastest at high security levels for small messages.

2 The Castagnos and Laguillaumie Cryptosystem

2.1 The Basic System

As mentioned in the previous section, Castagnos and Laguillaumie presented a linearly homomorphic encryption scheme based on a DL related problem, effectively solving a thirty-year-old open problem. Their scheme [5] is based on the hardness of the DDH problem in certain groups $\mathcal{G}$ that contain a subgroup $\mathcal{F}$ where solving the DL problem is easy. Castagnos and Laguillaumie call such a setting a DDH group with an easy DL subgroup and instantiate an example of one such group-subgroup pair as the class group of a non-maximal imaginary quadratic order with prime conductor [5]. The following is a simplified version of Definition 1 in [5], with unused parameters omitted.

Definition 1 ([5, Definition 1])**.** *A* DDH *group with an easy* DL *subgroup is a pair of algorithms* Gen *and* Solve*. The* Gen *algorithm takes as input two parameters λ and μ and outputs a tuple $(B, f, \mathfrak{g}, \mathfrak{f}, \mathcal{G}, \mathcal{F})$. Here, $\mathcal{G}$ is a finite cyclic group generated by $\mathfrak{g}$, $\mathcal{F}$ is a subgroup of $\mathcal{G}$ of order f generated by $\mathfrak{f}$, $|\mathcal{G}|/f$ is a λ-bit integer bounded above by B, and f is a μ-bit integer. The* Solve *algorithm is an efficient algorithm for solving the* DL *problem in $\mathcal{F}$ which is assumed to be easy, while the* DDH *problem in $\mathcal{G}$ is assumed to be hard even with access to the* Solve *algorithm.*

In addition, random powers $\mathfrak{g}^r$ with $0 \leq r \leq Bf - 1$ are assumed to be statistically indistinguishable from the uniform distribution on $\mathcal{G}$, and both images and pre-images under the canonical surjection $\mathcal{G} \to \mathcal{G}/\mathcal{F}$ are assumed to be efficiently computable. In slight abuse of terminology, we will refer to $\mathcal{G}$ as a DDH group and $\mathcal{F}$ an easy DL subgroup of $\mathcal{G}$, with an implicit assumption of the associated Gen and Solve algorithms.

For the scheme of [5], we let $f, \Delta_K \in \mathbb{Z}$ where $f > 0$, $\Delta_K < -4$, Δ_K is square-free and $\Delta_K \equiv 1 \pmod 4$. Then Δ_K is a fundamental discriminant that defines an imaginary quadratic field K. Let $C(\mathcal{O}_{\Delta_K})$ and $C(\mathcal{O}_{\Delta_f})$ denote the class group of the maximal order $\mathcal{O}_{\Delta_K}$ of K of discriminant Δ_K and the non-maximal suborder $\mathcal{O}_{\Delta_f}$ of $\mathcal{O}_{\Delta_K}$ of discriminant $\Delta_f = f^2\Delta_K$ and conductor f, respectively. Arithmetic in $C(\mathcal{O}_{\Delta_f})$ is conducted on reduced ideals, uniquely represented by a pair (a, b) where a, b are bounded integers and $a > 0$. There is an efficiently computable canonical injection $\psi_f : C(\mathcal{O}_{\Delta_K}) \to C(\mathcal{O}_{\Delta_f})$ and a corresponding canonical surjection $\bar{\varphi}_f : C(\mathcal{O}_{\Delta_f}) \to C(\mathcal{O}_{\Delta_K})$ whose kernel has order

$$|\ker(\bar{\varphi}_f)| = f \prod_{p|f} \left(1 - \left(\frac{\Delta_K}{p}\right)\frac{1}{p}\right) ,$$

where the product runs over the prime factors p of f and (Δ_K/p) is the Kronecker symbol. If every prime factor of f divides Δ_K, then $|\ker(\bar{\varphi}_f)| = f$. If in addition $\ker(\bar{\varphi}_f)$ is cyclic, then one can put $\mathcal{F} = \ker(\bar{\varphi}_f)$ and take $\mathcal{G}$ to be a suitable large cyclic subgroup of $C(\mathcal{O}_{\Delta_f})$.

Castagnos and Laguillaumie specifically chose $\Delta_K = -pq$ and $f = p$, where p is prime and q is a positive integer not divisible by p such that $q > 4p$. Then the ideal class of $\mathfrak{f} = (p^2, p)$ generates $\mathcal{F}$, and the DL in $\mathcal{F}$ is easy since for all $m \in \{1, 2, \ldots, p-1\}$, the ideal class of $\mathfrak{f}^m$ is given by $(p^2, L(m)p)$ where $L(m)$ is the unique odd inverse of $m \pmod p$ in the interval $[-p, p]$; see Proposition 1 of [5]. In Algorithms 1 and 2, we present the `Gen` algorithm that constructs the `DDH` group $\mathcal{G}$ with an easy `DL` subgroup $\mathcal{F}$ and the `Solve` algorithm that solves the `DL` problem in $\mathcal{F}$ in this setting; see [5, Figure 2]. For security reasons, as explained in Subsect. 2.2, we assume that q is also prime and that $(p/q) = (q/p) = -1$. The map $\psi = \psi_f$ in `Gen` is the aforementioned injection from $C(\mathcal{O}_{\Delta_K})$ into $C(\mathcal{O}_{\Delta_f})$; see [5, Lemma 3] and [8, Algorithm 9] for a method to efficiently compute this map. The call to $\texttt{Red}(\cdot)$ in Algorithm 2 outputs the two-integer representation of the unique reduced ideal equivalent to the input.

Algorithm 1. `Gen`

Input: λ, μ with $\lambda \geq \mu + 2$.
Output: $B, f, \mathfrak{g}, \mathfrak{f}, \mathcal{G}, \mathcal{F}$

1. Pick random integers p and q such that p is a μ-bit prime, q is a $(2\lambda - \mu)$-bit prime, $pq \equiv 3 \pmod 4$ and $(p/q) = (q/p) = -1$
2. Set $\Delta_K \leftarrow -pq$
3. Set $f \leftarrow p$
4. Set $\Delta_f \leftarrow f^2 \Delta_K$
5. Set $\mathfrak{f} \leftarrow [(f^2, f)]$ in $C(\mathcal{O}_{\Delta_f})$
6. Choose a small prime r such that $\gcd(r, f) = 1$ and $(\Delta_K / r) = 1$. Set $\mathfrak{r}$ to a prime ideal of $\mathcal{O}_{\Delta_K}$ lying above r
7. Pick $k \xleftarrow{\$} (\mathbb{Z}/f\mathbb{Z})^\star$ and set $\mathfrak{g} \leftarrow [\psi(\mathfrak{r}^2)] \cdot \mathfrak{f}^k$ in $C(\mathcal{O}_{\Delta_f})$
8. Set $B \leftarrow f \cdot \left\lceil \frac{\log(|\Delta_K|) \cdot \sqrt{|\Delta_K|}}{4\pi} \right\rceil$
9. Return $(B, f, \mathfrak{g}, \mathfrak{f}, \mathcal{G}, \mathcal{F})$

Algorithm 2. `Solve`

Input: $f, \mathfrak{f}, \mathfrak{m}$
Output: m such that $\mathfrak{m} = \mathfrak{f}^m$

1. Parse $\texttt{Red}(\mathfrak{m})$ as $(f^2, \tilde{x}\mathfrak{f})$
2. Return $\tilde{x}^{-1} \pmod f$

Algorithm 3. `KeyGen`

Input: λ
Output: Public key pk, secret key sk

1. $(B, f, \mathfrak{g}, \mathfrak{f}) \xleftarrow{\$} \texttt{Gen}(\lambda, \mu)$
2. $x \xleftarrow{\$} (\mathbb{Z}/Bf\mathbb{Z})$ and $\mathfrak{h} \leftarrow \mathfrak{g}^x$
3. $pk \leftarrow (B, f, \mathfrak{g}, \mathfrak{h}, \mathfrak{f})$ and $sk \leftarrow x$
4. Return (pk, sk)

Algorithms 3–7 present the linearly homomorphic encryption system first given in [5]. While we use the notation associated with the specific setting of class groups, this description applies to the generic setting of a DDH group with an easy DL subgroup of Definition 1. Here, plaintexts are integers modulo f, while ciphertexts are pairs of elements in $\mathcal{G}$. Thus, the size of the message space

is completely determined by the size of the easy DL subgroup $\mathcal{F}$; in the class group setting, this is precisely the conductor f of the non-maximal order $\mathcal{O}_{\Delta_f}$.

Algorithm 4. `Encrypt`

Input: λ, pk, message m
Output: Ciphertext (c_1, c_2)

1. Pick $r \xleftarrow{\$} \{0, \cdots, Bf-1\}$
2. Compute $c_1 \leftarrow \mathfrak{g}^r$
3. Compute $c_2 \leftarrow \mathfrak{f}^m \mathfrak{h}^r$
4. Return (c_1, c_2)

Algorithm 5. `Decrypt`

Input: $\lambda, pk, sk, (c_1, c_2)$
Output: Message m

1. Compute $\mathfrak{m} \leftarrow c_2/c_1^x$
2. $m \leftarrow \texttt{Solve}(f, \mathfrak{f}, \mathcal{F}, \mathfrak{m})$
3. Return m

Algorithm 6. `EvalSum`

Input: λ, pk,
$(c_1, c_2) = \texttt{Encrypt}(pk, m)$,
$(c_1', c_2') = \texttt{Encrypt}(pk, m')$
Output: (C_1, C_2) such that
$\texttt{Decrypt}(sk, (C_1, C_2)) = m + m'$

1. Compute $c_1'' \leftarrow c_1 c_1'$, $c_2'' \leftarrow c_2 c_2'$
2. Pick $r \xleftarrow{\$} \{0, \cdots, Bf-1\}$
3. Return $(c_1'' \mathfrak{g}^r, c_2'' \mathfrak{h}^r)$

Algorithm 7. `EvalScal`

Input: λ, pk, α,
$(c_1, c_2) = \texttt{Encrypt}(pk, m)$
Output: (C_1, C_2) such that
$\texttt{Decrypt}(sk, (C_1, C_2)) = \alpha m$

1. Compute $c_1' \leftarrow c_1^\alpha$, $c_2' \leftarrow c_2^\alpha$
2. Pick $r \xleftarrow{\$} \{0, \cdots, Bf-1\}$
3. Return $(c_1' \mathfrak{g}^r, c_2' \mathfrak{h}^r)$

2.2 Security

It is easy to see that if one can solve the discrete logarithm problem in $\mathcal{G}$, then one can recover the secret key sk and totally break the scheme of [5]. Castagnos and Laguillaumie show that the `DL` problem in $\mathcal{G}$ is at least as hard as the `DL` problem in $\mathcal{G}/\mathcal{F}$.

Theorem 1 ([5, Theorem 2])**.** *Let $\mathcal{G}$ be a* `DDH` *group with an easy* `DL` *subgroup. Then the* `DL` *problem in $\mathcal{G}/\mathcal{F}$ reduces to the* `DL` *problem in $\mathcal{G}$.*

The `DDH` problem in our context reads as follows.

Definition 2 (Decisional Diffie Hellman Problem). *Let $\mathcal{G}$ be a* `DDH` *group of order n with an easy* `DL` *subgroup $\mathcal{F}$ and $\mathfrak{g}$ a generator of $\mathcal{G}$. Let x, y, z be integers such that $x, y, z \xleftarrow{\$} \mathbb{Z}/n\mathbb{Z}$. The Decisional Diffie Hellman Problem consists of deciding whether $\mathfrak{g}^{xy} = \mathfrak{g}^z$, given $(\mathfrak{g}, \mathfrak{g}^x, \mathfrak{g}^y, \mathfrak{g}^z)$ and access to the* `Solve` *algorithm.*

Theorem 2 ([5, Theorem 4])**.** *The scheme described in Algorithms 1–7 is semantically secure under chosen plaintext attacks (*`ind-cpa`*) if and only if the* `DDH` *problem is hard in $\mathcal{G}$.*

The following problems were introduced by Bresson et al. in [3] and Paillier in [11] respectively, and were then adapted by Castagnos and Laguillaumie in [5].

Definition 3 (Lift Diffie-Hellman Problem). *Let $\mathcal{G}$ be a* DDH *group of order n with an easy* DL *subgroup $\mathcal{F}$ and $\mathfrak{g}$ a generator of $\mathcal{G}$. Let $x, y \xleftarrow{\$} \mathbb{Z}/n\mathbb{Z}$ and let $\pi\colon \mathcal{G} \to \mathcal{G}/\mathcal{F}$ be the canonical surjection. The Lift Diffie-Hellman (*LDH*) problem consists of computing g^{xy}, given $(\mathfrak{g}, g^x, g^y, \pi(g^{xy}))$ and access to the* Solve *algorithm.*

Definition 4 (Partial Discrete Logarithm Problem). *Let $\mathcal{G}$ be a* DDH *group of order n with an easy* DL *subgroup $\mathcal{F}$ and $\mathfrak{g}$ a generator of $\mathcal{G}$. Let $x \xleftarrow{\$} \mathbb{Z}/n\mathbb{Z}$. The Partial Discrete Logarithm (*PDL*) problem consists of computing $x \pmod{|\mathcal{F}|}$, given $\mathfrak{g}$ and $\mathfrak{g}^x$ and access to the* Solve *algorithm.*

Theorem 3 ([5, Theorem 3]). *The scheme described in Algorithms 1–7 is one-way under chosen plaintext attacks (*ow-cpa*) if and only if the* LDH *problem (equivalently, the* PDL *problem) is hard.*

Castagnos and Laguillaumie show that the LDH and PDL problems are equivalent [5, Theorem 1]. They also show that knowledge of the order n of $\mathcal{G}$ makes it possible to solve the PDL problem efficiently [5, Lemma 1].

For security reasons, it is desirable to work in a cyclic subgroup of $C(\mathcal{O}_{\Delta_f})$ that is as large as possible. To that end, Hamdy and Möller [7] recommend to choose a fundamental Δ_K for which the 2-Sylow subgroup of the class group $C(\mathcal{O}_{\Delta_K})$, and hence the even part of the class number $h(\Delta_K) = |C(\mathcal{O}_{\Delta_K})|$, is minimal. The construction in [5] achieves this, since for $\Delta_K = -pq$, the even part of the class number is exactly 2 if p, q are primes with $(p/q) = (q/p) = -1$, and that value is as small as possible for non-prime discriminants. In addition, Castagnos and Laguillaumie also require $\mu > 80$ in order to ensure that the probability of the conductor p dividing the odd part of $h(\Delta_K)$ is extremely low according to the Cohen-Lenstra heuristics. A large cyclic subgroup of $C(\mathcal{O}_{\Delta_K})$ of order s, where s is a large factor of $h(\Delta_K)$, thus produces a large cyclic subgroup $\mathcal{G}$ of $C(\mathcal{O}_{\Delta_f})$ of order ps, and s is the security parameter for the scheme. The Cohen-Lenstra heuristics in fact predict that the odd part of $C(\mathcal{O}_{\Delta_K})$ is itself cyclic with very high probability. Under these assumptions, finding the order $|\mathcal{G}|$ is believed to be intractable.

2.3 A Variant of the Basic System

Castagnos and Laguillaumie proposed a variant that aims to reduce the size of the first component c_1 of a ciphertext (c_1, c_2) [5, Section 4.2]. They suggested constructing the generator $\mathfrak{g}$ of $\mathcal{G}$ in $C(\mathcal{O}_{\Delta_K})$ so that $\mathfrak{h} \in C(\mathcal{O}_{\Delta_K})$ and hence $c_1 \in C(\mathcal{O}_{\Delta_K})$. The ciphertext c_2 can then be generated by lifting $\mathfrak{h}$ to $C(\mathcal{O}_{\Delta_p})$ using the ψ map. Thus, we have the following changes for this variant. Note that the semantic security of this variant now relies on the intractability of a different, less standard modification of the DDH problem.

Modification to Algorithm 1

7. Set $\mathfrak{g} \leftarrow [\mathfrak{r}^2]$ in $C(\mathcal{O}_{\Delta_K})$

Algorithm 8. `Encrypt`

Input: λ, pk, message m
Output: Ciphertext (c_1, c_2)

1. Pick $r \xleftarrow{\$} \{0, \cdots, Bf - 1\}$
2. Compute $c_1 \leftarrow \mathfrak{g}^r$
3. Compute $c_2 \leftarrow \mathfrak{f}^m \psi(\mathfrak{h}^r)$
4. Return (c_1, c_2)

Algorithm 9. `Decrypt`

Input: $\lambda, pk, sk, c_1, c_2$
Output: Message m

1. Compute $\mathfrak{m} \leftarrow c_2/\psi(c_1^x)$
2. $m \leftarrow \texttt{Solve}(B, f, \mathfrak{g}, \mathfrak{f}, \mathfrak{m})$
3. Return m

2.4 Expanding the Message Space

The condition $q > 4p$ implies $|\Delta_K| > 4p^2$ and hence $p^2 < \sqrt{\Delta_p}/2$, which ensures that $\texttt{Red}(\mathfrak{f}) = (p^2, p)$ is a reduced ideal in $\mathcal{O}_{\Delta_f}$ [9, Theorem 5.6]. This restriction allows for a polynomial time `Solve` algorithm, but it also introduces a fixed upper bound on the size of the message space for a given security level. To see this more clearly, consider a factorization based linearly homomorphic scheme such as the Paillier cryptosystem. Its hardness is based on the factorization of the RSA modulus and thus, the size of the message space is the size of the modulus which is the security parameter. In the CL schemes described above, the message space has size p, so the bound $q > 4p$ forces $\Delta_K > 4p^2$. For example, based on [1, Table 4], a security level of 128 bits corresponds to factoring a modulus of bit size 3072 and computing discrete logarithms in a class group corresponding to a 1828-bit discriminant Δ_K. In the Paillier scheme, a security level of 3072 bits thus corresponds to messages of bit length of 3072. Yet, in the Castanos-Laguilliomie scheme, messages whose length is equal to the corresponding security level of 1828 bits necessitate using a discriminant of size at least $2 \cdot 1828 + 2 = 3658$ bits, far larger than what is required at the same security level. Thus, the CL variants discussed so far lose their advantage over factoring based schemes.

To solve this problem, Castagnos and Laguillaumie proposed a variant of their scheme that drops the requirement $q > 4p$ and has no restriction on the size of q in `Gen` (Algorithm 1). In this case, however, the ideal (p^2, p) of $\mathcal{O}_{\Delta_p}$ and its powers may no longer be reduced. In order to still guarantee a polynomial time `Solve` algorithm, one solution is to lift the ideal (p^2, p) to the order $\mathcal{O}_{\Delta_{p^2}}$ of discriminant $\Delta_{p^2} = p^4 \Delta_K$ where the lifted ideal is reduced since $p^2 < \sqrt{|\Delta_{p^2}|}/2$ if $|\Delta_K| > 4$. The class $\mathfrak{f} = [(p^2, p)] \in C(\mathcal{O}_{\Delta_p})$ lifts to the ideal class $\mathfrak{f}_l \in c(\mathcal{O}_{\Delta_{p^2}})$ whose unique reduced representative is again $[(p^2, p)]$, where the lift is now effected by the map $\psi : C(\mathcal{O}_{\Delta_p}) \to C(\mathcal{O}_{\Delta_{p^2}})$.

Castagnos and Laguillaumie show that $\mathfrak{f}_l$ belongs to the cyclic subgroup of $C(\Delta_{p^2})$ generated by $[(p^2, p)]$ where $\mathfrak{f}_l = \psi(\mathfrak{f})$ is the lift of $\mathfrak{f}$ under the lifting map ψ that maps elements in $\mathcal{O}_{\Delta_p}$ to elements in $\mathcal{O}_{\Delta_{p^2}}$ [5, Section 4.1]. So we precompute the discrete logarithm z of $\mathfrak{f}_l$ with respect to $[(p^2, p)]$ in `Gen` using a

technique analogous to that used in `Solve`, but computing inside $C(\Delta_{p^2})$. Our computations show that $z = 1$ almost always. We have the following `Gen` and `Solve` algorithms for this variant.

Algorithm 10. `Gen`

Input: $\lambda,\ \mu$ with $\lambda \geq \mu + 2$.
Output: $B, f, z, \mathfrak{g}, \mathfrak{f}, \mathcal{G}, \mathcal{F}$

1. Pick random integers p and q such that p is a μ-bit prime, q is a $(2\lambda-\mu)$-bit prime, $pq \equiv 3 \pmod 4$ and $(p/q) = (q/p) = -1$ if $q \neq 1$
2. Set $\Delta_K \leftarrow -pq$
3. Set $f \leftarrow p$
4. Set $\Delta_f \leftarrow f^2 \Delta_K$
5. Set $\mathfrak{f} \leftarrow [(f^2, f)]$ in $C(\mathcal{O}_{\Delta_f})$
6. Parse $\texttt{Red}(\psi(\mathfrak{f}))$ as $(f^2, \tilde{z}f)$
7. $z \leftarrow \tilde{z}^{-1} \pmod f$
8. Choose a small prime r such that $\gcd(r, f) = 1$ and $(\Delta_K/r) = 1$. Set $\mathfrak{r}$ to be a prime ideal of $\mathcal{O}_{\Delta_K}$ lying above r
9. Pick $k \xleftarrow{\$} (\mathbb{Z}/f\mathbb{Z})^\star$ and set $\mathfrak{g} \leftarrow [\psi(\mathfrak{r}^2)] \cdot \mathfrak{f}^k$ in $C(\mathcal{O}_{\Delta_f})$
10. Set $B \leftarrow f \cdot \left\lceil \frac{\log(|\Delta_K|) \cdot \sqrt{|\Delta_K|}}{4\pi} \right\rceil$
11. Return $(B, f, z, \mathfrak{g}, \mathfrak{f}, \mathcal{G}, \mathcal{F})$

Algorithm 11. `Solve`

Input: $z, f, \mathfrak{f}, \mathfrak{m}$
Output: m such that $\mathfrak{m} = \mathfrak{f}^m$

1. Compute $\mathfrak{m}' \leftarrow \psi(\mathfrak{m})$
2. Parse $\texttt{Red}(\mathfrak{m}')$ as $(f^2, \tilde{y}f)$
3. Return $z\tilde{y}^{-1} \pmod f$

In this version, p can be chosen independently of the security level, subject to the restriction that it is large enough so p does not divide q with very high probability (e.g. at least 80 bits). We note that this idea can also be applied to the variant presented in Subsect. 2.3.

3 Extensions

The original probabilistic encryption scheme in [5] and its modifications presented in Sect. 2 all use a prime conductor p. Castagnos and Laguillaumie also suggested the use of a composite conductor f, which could potentially improve the efficiency of their schemes, and to allow the message space to be increased arbitrarily without increasing the security level (governed by the size of the fundamental discriminant Δ_K). Specifically, they proposed $f = \prod_{i=1}^{N} p_i$ or

$f = p^t$ where $N, t \in \mathbb{Z}_{\geq 1}$ and p_i, p are primes. In this section, we describe modified versions of the algorithms presented Sect. 2 for the more general conductor $f = \prod_{i=1}^{N} p_i^t$ that includes the two proposed forms as the special cases $t = 1, N > 1$ and $t > 1, N = 1$ and the original scheme as the case $N = t = 1$. To ensure that the kernel of the surjection $\bar{\varphi}_f : C(\mathcal{O}_{\Delta_f}) \to C(\mathcal{O}_{\Delta_K})$ is f, we put $\Delta_K = -p_1 p_2 \cdots p_N q$. It is easy to deduce that the ideal (f^2, f) is reduced in $\mathcal{O}_{\Delta_f}$ when $q > 4(p_1 p_2 \cdots p_N)^{2t-1}$. If this is not the case, we need to proceed as in Subsect. 2.4 and lift the class $\mathfrak{f}$ of (f^2, f) to $C(\mathcal{O}_{\Delta_{f^2}})$ via the map $\psi : C(\mathcal{O}_{\Delta_f}) \to C(\mathcal{O}_{\Delta_{f^2}})$. In order to focus entirely on the differences arising in all our algorithms when replacing a prime conductor $f = p$ by a composite conductor $f = \prod_{i=1}^{N} p_i^t$, we assume that no such lifting is necessary. The `Gen` algorithm for this extension is as follows. The `KeyGen` algorithm remains unchanged. We present modified versions of `Encrypt`, `Decrypt` and `Solve` separately for the cases $t = 1$ and $t > 1$.

Algorithm 12. `Gen`

Input: $\lambda,\ \mu$

Output: $B, f, \mathfrak{g}, \mathfrak{f}, \mathcal{G}, \mathcal{F}$

1. Pick random primes $p_1, p_2, \cdots, p_N, q$ such that $p_1 p_2 \cdots p_N$ is a μ-bit integer, q is a $(2\lambda - \mu)$-bit prime, $p_1 p_2 \cdots p_N q \equiv 3 \pmod 4$ and $(p_i/p_j) = 1$ and $(p_i/q) = (q/p_i) = -1$ for $1 \leq i, j \leq N$
2. Set $\Delta_K \leftarrow -p_1 p_2 \cdots p_N q$
3. Pick $t \xleftarrow{\$} \mathbb{Z}_{>0}$ and set $f \leftarrow (p_1 p_2 \cdots p_N)^t$
4. Set $\Delta_f \leftarrow f^2 \Delta_K$
5. Set $\mathfrak{f} \leftarrow [(f^2, f)]$ in $C(\mathcal{O}_{\Delta_f})$
6. Choose a small prime r such that $\gcd(r, f) = 1$ and $(\Delta_K/r) = 1$. Set $\mathfrak{r}$ a prime ideal lying above r
7. Pick $k \xleftarrow{\$} (\mathbb{Z}/f\mathbb{Z})^\star$ and set $\mathfrak{g} \leftarrow [\psi(\mathfrak{r}^2)] \cdot \mathfrak{f}^k$ in $C(\mathcal{O}_{\Delta_f})$
8. Set $B \leftarrow |\mathcal{M}| \cdot \left\lceil \frac{\log(|\Delta_K|) \cdot \sqrt{|\Delta_K|}}{4\pi} \right\rceil$
9. Return $(B, f, \mathfrak{g}, \mathfrak{f}, \mathcal{G}, \mathcal{F})$

3.1 Case $t > 1$

Note that since $\ker(\bar{\varphi}_f)$ contains subgroups of order p_i^t for all i, one could also encrypt $m_i \pmod{p_i^t}$. The resulting decryption simply needs to solve the simultaneous congruences $m \equiv m_i \pmod{p_i^t}$ via Chinese remaindering. This yields the following modifications.

Modification to Algorithm 12

5. Set $\mathfrak{f}_i \leftarrow [(p_i^{2t}, p_i^t)]$ in $C(\mathcal{O}_{\Delta_f})$ $\quad \forall i \in \{1, \cdots, N\}$

Algorithm 13. `Encrypt`

Input: λ, pk, m
Output: Ciphertext (c_1, c_2)

1. Pick $r \xleftarrow{\$} \{0, \cdots, Bf - 1\}$
2. Compute $m_i \leftarrow m \pmod{p_i}$
3. Compute $c_1 \leftarrow \mathfrak{g}^r$
4. Compute $\hat{c}_i \leftarrow \mathfrak{f}_i^{m_i} h^r$
5. Return $c_1, \hat{c}_1, \cdots, \hat{c}_n$

Algorithm 14. `Decrypt`

Input: $\lambda, pk, sk, c_1, \hat{c}_1, \cdots, \hat{c}_n$
Output: Message m

1. Compute $\mathfrak{m}_i \leftarrow \hat{c}_i / c_1^x$
2. Compute $m_i \leftarrow \texttt{Solve}(p_i^t, \mathfrak{f}_i, \mathfrak{m}_i)$
3. Solve $m \equiv m_i \pmod{p_i}$
4. Return m

Algorithm 15. `Solve`

Input: $f, \mathfrak{f}, \mathfrak{m}$
Output: m such that $\mathfrak{f}^m = \mathfrak{m}$

1. **for** $i = 1$ to N **do**
2. Compute $\mathfrak{f}_i \leftarrow \mathfrak{f}^{f/p_i^t}$
3. Compute $\mathfrak{m}_i \leftarrow \mathfrak{m}^{f/p_i^t}$
4. Set $x_0 \leftarrow 0$
5. Compute $\gamma \leftarrow \mathfrak{f}_i^{p_i^{t-1}}$
6. **for** $k \leftarrow 0$ to $t - 1$ **do**
7. Compute $\mathfrak{m}'_k \leftarrow (\mathfrak{f}_i^{-x_k} \mathfrak{m}_i)^{{p_i}^{t-1-k}}$
8. Compute $d_k \leftarrow \texttt{Solve}(p_i, \gamma, \mathfrak{m}'_k)$
9. Set $x_{k+1} \leftarrow x_k + p^k d_k$
10. **end for**
11. Set $m_i \leftarrow x_t$
12. **end for**
13. Solve $m \equiv m_i \pmod{p_i^t} \quad \forall i \in \{1, \cdots, N\}$ using CRT
14. Return m

3.2 $t = 1$

If $t = 1$, we have $f = p_1 p_2 \cdots p_N$. If we assume $q > 4f$ as before, *i.e.*, $\lambda \geq \mu + 2$ in `Gen`, then the reduced representative of the ideal class $\mathfrak{f} \in C(\mathcal{O}_{\Delta_f})$ is (f^2, f) and $\mathfrak{f}$ generates a cyclic group of order f in $C(\mathcal{O}_{\Delta_f})$. Thus, our `Encrypt`, `Decrypt` and `Solve` algorithms remain unchanged from their original versions. However, since the `Solve` algorithm is essentially an inversion modulo f (a prime in the `CL` schemes) and f is now composite, we can perform computations modulo the individual prime factors of f and retrieve the message modulo f using the Chinese Remainder Theorem (CRT). This can be done in three ways:

1. The first CRT modification is straightforward: we simply compute inversions modulo each prime divisor of f and use CRT to retrieve the message modulo f. The modified `Solve` algorithm is as follows:

Algorithm 16. `Solve`

Input: $f, \mathfrak{f}, \mathfrak{m}$
Output: m such that $\mathfrak{m} = \mathfrak{f}^m$
1. Parse $\texttt{Red}(\mathfrak{m})$ as $(f^2, \tilde{x}f)$
2. Compute $m_i \leftarrow \tilde{x}^{-1} \pmod{p_i}$
3. Solve $m \equiv m_i \pmod{p_i}$
4. Return m

2. The second CRT modification utilizes the idea that $\mathcal{F}$ contains order p_i subgroups for each i that are generated by the elements $\mathfrak{f}_i = \mathfrak{f}^{(f/p_i)}$ represented by the ideals (p_i^2, p_i). Thus, one can compute $m_i \equiv m \pmod{p_i}$ and encrypt $\mathfrak{m} = \prod_{i=1}^{n} \mathfrak{f}_i^{m_i}$. Clearly, $\mathfrak{m}$ is of form $(f^2, \tilde{x}f)$ *i.e.*, $\mathfrak{m} \in \langle \mathfrak{f} \rangle$ and we have the following modifications.

Modification to Algorithm 12

5. Set $\mathfrak{f}_i \leftarrow [(p_i^2, p_i)]$ in $C(\mathcal{O}_{\Delta_f})$

Algorithm 17. `Encrypt`

Input: λ, pk, m
Output: Ciphertext (c_1, c_2)
1. Pick $r \xleftarrow{\$} \{0, \cdots, Bf - 1\}$
2. Compute $m_i \leftarrow m \pmod{p_i}$
3. Compute $c_1 \leftarrow \mathfrak{g}^r$
4. Compute $c_2 \leftarrow \mathfrak{f}_1^{m_1} \mathfrak{f}_2^{m_2} \cdots \mathfrak{f}_N^{m_N} h^r$
5. Return c_1, c_2

Algorithm 18. `Solve`

Input: $f, \mathfrak{f}, \mathfrak{m}$
Output: m such that $\mathfrak{m} = \mathfrak{f}^m$
1. Parse $\texttt{Red}(\mathfrak{m})$ as $(f^2, \tilde{x}f)$
2. $m_i \leftarrow (\tilde{x}f/p_i)^{-1} \pmod{p_i}$
3. Solve $m \equiv m_i \pmod{p_i}$
4. Return m

3. The third CRT variant also uses the fact that $\mathfrak{f}_i = [(p_i^2, p_i)]$ is cyclic of order p_i and generates N ciphertexts $\mathfrak{m}_i = \mathfrak{f}_i^{m_i}$ where $m_i \equiv m \pmod{p_i}$. The modification to the `Gen` algorithm is identical to that of the previous variant, and the modified `Encrypt` and `Decrypt` algorithms take the following form:

Algorithm 19. `Encrypt`

Input: λ, pk, m
Output: Ciphertext $(c_1, \tilde{c}_1, \ldots, \tilde{c}_n)$
1. Pick $r \xleftarrow{\$} \{0, \cdots, Bf - 1\}$
2. Compute $m_i \leftarrow m \pmod{p_i}$
3. Compute $c_1 \leftarrow \mathfrak{g}^r$
4. Compute $\hat{c}_i \leftarrow \mathfrak{f}_i^{m_i} h^r$
5. Return $c_1, \hat{c}_1, \cdots, \hat{c}_n$

Algorithm 20. `Decrypt`

Input: $\lambda, pk, sk, c_1, \hat{c}_1, \cdots, \hat{c}_n$
Output: Message m
1. Compute $\mathfrak{m}_i \leftarrow \hat{c}_i / c_1^x$
2. Compute $m_i \leftarrow \texttt{Solve}(p_i, \mathfrak{f}_i, \mathfrak{m}_i)$
3. Solve $m \equiv m_i \pmod{p_i}$
4. Return m

3.3 Security Considerations for the Extensions

It is easy to verify that the extensions presented in Sect. 3 preserve the linearly homomorphic properties. Moreover, the security considerations for the original CL scheme remain unchanged throughout these extensions, with the appropriate conditions on the Legendre symbols (p_i/p_j), (p_i/q) and (q/p_i) as stated in Algorithm 12. As described in Subsect. 2.2, the fundamental discriminants Δ_K should be chosen such that 2-Sylow subgroup of the class group $C(\mathcal{O}_{\Delta_K})$ is as small as possible.

If N is the number of prime factors of Δ_K, then the 2-rank of $C(\mathcal{O}_{\Delta_K})$ is $N-1$, and 2^{N-1} divides $h(\Delta_K)$. We wish to ensure that this is in fact the highest power of 2 dividing $h(\Delta_K)$. For discriminants of the form $\Delta_K = -pq$ as in Algorithm 1, we have $N = 2$, so $h(\Delta_K)$ is even. The conditions $(p/q) = (q/p) = -1$ guarantee that $h(\Delta_K)/2$ is odd. Similarly, when $\Delta_K = -p_1 p_2 \cdots p_N q$, we see that 2^{N-1} divides $h(\Delta_K)$. If $(p_i/p_j) = 1$ and $(p_i/p_N) = (p_N/p_i) = -1$ for $1 \leq i, j < N$, then no higher power of 2 divides $h(\Delta_K)$ (see, for example, [2]).

4 Parameter Choices

As described in [5], the main concern with selecting parameters is that it should be computationally infeasible to compute $h(\Delta_K)$, the class number of the maximal order $\mathcal{O}_{\Delta_K}$, as knowledge of the class number in this setting enables the computation of discrete logarithms in $C(\mathcal{O}_{\Delta_f})$. Biasse et al. in [1] gave estimates of discriminant sizes to provide various levels of security using the best-known index calculus algorithms of subexponential complexity. In Table 1, we give these sizes for the 128-, 192-, and 256-bit security levels, along with the corresponding RSA modulus sizes required for Paillier [11] and Bresson et al. [3]. Note that generic group algorithms do not play a role here, as their complexity is worse than the index calculus algorithms.

Table 1. Parameter sizes (in bits)

Security level	RSA modulus	Δ_K	Δ_f (for n-bit messages)			
			16	80	256	32768
128	3072	1828	1860	1988	2340	67364
192	7680	3598	3630	3758	4110	69134
256	15360	5972	6004	6132	6484	71508

We also list, in Table 1, the sizes of the non-fundamental discriminants Δ_f required to provide the given security level for various message sizes. As mentioned earlier, Paillier and Bresson et al. can encrypt messages of up to the same size as the RSA modulus used, which is determined by the desired security level. The variants of the Castagnos and Laguillaumie cryptosystem have their security

level fixed primarily by the size of the fundamental discriminant Δ_K, and can work with different message sizes by varying the conductor. We see that smaller message spaces should be quite favorable for the Castagnos and Laguillaumie system and its variations, as even the non-maximal discriminans are quite small compared to the RSA moduli required Paillier and Bresson et al. at the same security levels. We also see that larger message spaces can be used at a fixed security level, but note that the extensions involving prime power conductors are necessary, since all primes dividing the conductor must also divide Δ_K. Even with this extension, the CL cryptosystems would not be very efficient on such large messages, as the discriminants required are very large.

The two other considerations for security parameter choices are the sizes of primes dividing the conductor f and the upper bound for selecting random exponents in the protocol. We now discuss these two considerations.

4.1 Restrictions on Prime Factors of f

Castagnos and Laguillaumie insisted on using a conductor (a prime p) of size at least 80 bits to ensure $\gcd(p, h(\Delta_K)) = 1$ with high probability implying that the odd part of the class number is completely unknown. This is important as the odd part of $h(\Delta_K)$ is the security parameter and knowing the size of the odd part, s, leads to a total break of the scheme as shown in Subsect. 2.2. If a divisor of s is known, then computing s itself may be easier. Extrapolating this idea to our extension in which the conductor is a product of prime powers would then imply that the prime divisors of the conductor be at least 80 bits. This restriction is detrimental to the performance of both the original and extended versions of the cryptosystem.

However, we believe that this is an unnecessary restriction when one considers how a known factor of $h(\Delta_K)$ could be exploited in practice. The best known algorithms to compute the class number are subexponential index-calculus algorithms and generic group algorithms. There is no known way to speed up the index calculus algorithms given a divisor of the class number, as the complexity depends on the discriminant as opposed to the class number. Thus, the discriminant sizes recommended by Biasse et al. in [1] offer enough protection against index calculus algorithms even if a divisor of the class number is known.

When considering generic algorithms, on the other hand, a known divisor of the class number does improve the running time, as one can target the unknown part directly. We consider the worst case that the entire conductor f divides the odd part of the class number (note that f itself is odd). Let 2^{k_1} be the even and $s = f \cdot s'$ be the odd factors of $h(\Delta_K)$ where $k_1 \in \mathbb{Z}_{\geq 0}$ and s' is the unknown part of the odd part s. Since $h(\Delta_K) < \frac{1}{\pi} \log(|\Delta_K|)\sqrt{\Delta_K}$ (see, for example, [6, §5.10]), we have,

$$s' < \frac{1}{2^{k_1} \cdot f \cdot \pi} \log(|\Delta_K|)\sqrt{|\Delta_K|}. \tag{1}$$

Generic group algorithms can be used to compute s' in time $O(\sqrt{s'})$. Ignoring constants and lower-order terms, in order to provide b bits of security, we require

that $\sqrt{s'} > 2^b$. Combining this with (1) yields the following upper bound on $\log_2 f$:

$$\log_2 f \leq \log_2(\log(|\Delta_K|)) + \frac{1}{2}\log_2(|\Delta_K|) - k_1 - 2 - 2b. \tag{2}$$

For example, following the recommendations in [1], Castagnos and Laguillaumie chose a discriminant of size 1828 bits at the 128-bit security level to prevent index calculus attacks in $C(\mathcal{O}_{\Delta_K})$. Substituting these values in Eq. 2 results in

$$\log_2 f \leq 667 - k_1,$$

meaning that we can tolerate known divisors of the conductor of size over 600 bits before the generic attacks would work in fewer than 2^b operations. Note that using conductors with prime divisors larger than this bound is also highly unlikely to be an issue, because, as discussed in [5], the probability that primes of this size divide the class number is negligible—indeed, in [5], using primes larger than 2^{80} was deemed to be sufficient. Thus, we conclude that based on the current state of knowledge of possible attacks on the cryptosystem, prime divisors of any size in the conductor are unlikely to result in any loss of security. This is because the discriminant sizes required to avoid index calculus attacks result in class numbers that are sufficiently large to prevent the generic algorithms, which could exploit a known factor of the class number, from working in fewer than 2^b operations.

4.2 Selection of Random Exponents

The bound B on exponents in Algorithm 1, taken directly from [5], is designed to ensure that the resulting group elements are selected from the entire class group uniformly at random, a necessary condition for the security proofs to hold. In [5], the formula for B has a factor of 2^{80} in order to ensure statistical distance of 2^{-80} from the uniform distribution; in our exposition above, we instead use the size of the message space f, thus allowing the resulting statistical distance $1/f$ to vary with the size of the message space. In the following section, in which we benchmark the practical performance of their version as well as our extensions, we will consider this version of the cryptosystem.

However, it is also known that one can obtain similar security proofs using much shorter exponents if one is willing to use slightly non-standard versions of the intractability assumptions, a critical performance optimization. There is no known way to take advantage of knowledge that discrete logarithms are small in the index calculus algorithms, so these have no bearing on the exponent bounds. The only concern is with generic algorithms of square root complexity, which imply that all exponents should be chosen with at least $2b$ bits for a b-bit security level. Koshiba and Kurosawa [10] proved that security proofs relying on Diffie-Hellman problems also hold assuming that such short exponent versions of the discrete logarithm problem are intractable. Thus, it is at least plausible that the security proofs of [5] also hold under similarly modified intractability assumptions. We will also consider short exponent versions of our cryptosystems,

as is typically done in practice, in the benchmarks presented in the next section, using exponents of $2b$ bits.

5 Numerical Results

In this section we present numerical results from benchmarking our extended version of the cryptosystem of Castagnos and Laguillaumie [5]. Our first set of experiments were designed to determine which variation and parameter selection yields the fastest encryption and decryption times for different combinations of security level and message size. The second set of experiments compares the best versions against the Paillier [11] and Bresson et al. [3] cryptosystems.

Our experiments were carried out on a standard desktop with 4 Intel Core i5-2400 CPUs, each CPU with 4 cores, running at 3.10 GHz, and 8 GB RAM, running Fedora 28. Our programs are written in C/C++ (using gcc version 8.1.1) with GMP (version 6.1.2) and NTL (version 11.3.2) support for arbitrary precision arithmetic. We used Maxwell Sayles's optimized binary quadratic forms library [12] for ideal arithmetic. Generic single ideal exponentiations were computed using the 8-NAF method while double exponentiations were computed using the interleaving method with window size 8. However, the exponentiations of $\mathfrak{f} = [(f^2, f)]$ were performed by simply computing the inverse of the exponent modulo the order of the ideal class of $\mathfrak{f}$ and setting $\mathfrak{f}^x$ as $[(f^2, x^{-1}f)]$.

5.1 Comparison of Variations of the Castagnos and Laguillaumie Cryptosystem

The objective of these experiments was to find the fastest CL variation among different choices of the conductor at each security level. Since the message space can be chosen independently of the security parameter, we considered message space sizes of 16, 80 and 256 bits, as well as the same message size of the Pailler [11] cryptosystem at the same security level. We considered 80 bits as Castagnos and Laguillaumie promote 80 bits of message space for practical applications in their paper, and the remaining message sizes were selected to illustrate performance with smaller and larger message sizes at fixed security levels. As one can select short exponents in the CL and BCP schemes, we performed each experiment twice for these cryptosystems, once with full domain exponents and next with shorter exponents.

We used conductors of the general form $f = (p_1 \cdots p_N)^t$, and varied N and t to find the optimal (N, t) pair for each security level, message space size, and variant. We performed some preliminary experiments to find the maximum values of N and t that one should consider during these experiments. Our observations showed that the bounds $N = 9$ and $t' \leq t < t' + 5$ were sufficient to find the optimal (N, t) pairs for a message space at a given security level. Here, $t' = \lceil |\mathcal{M}|/\Delta_K \rceil$ is the minimum t value required to achieve the message space size at a security level. The choice of $N = 9$ and five more values of t were merely to see the effect of increasing N and t values on the performance. We generated

10 different parameters (f, Δ_K) for each of the 45 combinations of N and t. As N and t increase, it is difficult to maintain exact conductor sizes as desired and so we made sure that the conductor has at least the required minimum number of bits, but at most 3 bits more.

We considered all four variations of the cryptosystem of Castagnos and Laguillaumie described in Sect. 2 as well as the modifications to the original conductor choice in Sect. 3. We denote the four schemes as Basic, Variant, BasicPlus and VariantPlus, where "Variant" denotes the version described in Subsect. 2.3 with smaller ciphertexts, and the latter two are the Basic and Variant with the expansion technique from Subsect. 2.4 applied. We also used all the CRT-based encryption and decryption variations described in Sect. 3.

In summary, for every combination of security level, message size, conductor decomposition (N and t), and specific cryptosystem variant, we computed the average encryption and decryption times in milliseconds taken over the same set of 1000 messages. Table 2 contains a summary of these experiments. For each security level and message size pair, we list the average encryption and decryption times for the fastest variant along with the corresponding N and t values. We record this for both full domain exponents and short exponents. The variants are specified using the short-hand notation B, V, BP, and VP, for Basic, Variant, BasicPlus, and VariantPlus, respectively. ED denotes encryption and decryption, 2ED denotes decryption with CRT2 and its corresponding encryption and 3ED denotes encryption with CRT3 and its corresponding encryption.

Table 2. Summary of best performances by CL schemes (in ms)

Security	Message	Short exponents					Full exponents				
		N	t	Scheme	Enc.	Dec.	N	t	Scheme	Enc.	Dec.
128	16	1	1	B-ED	14	9	1	1	B-ED	58	28
	80	1	1	B-ED	15	9	1	1	B-ED	63	33
	256	1	1	B-ED	18	11	1	1	V-ED	96	50
	3072	1	1	B-ED	156	98	6	2	VP-ED	964	794
192	16	1	1	B-ED	47	27	1	1	B-ED	223	115
	80	1	1	B-ED	47	27	1	1	B-ED	249	128
	256	1	1	B-ED	54	31	1	1	B-ED	**320**	166
							2	1	V-ED	328	**160**
	7680	1	1	B-ED	871	508	6	3	VP-ED	7276	6702
256	16	1	1	B-ED	116	65	1	1	B-ED	672	342
	80	1	1	B-ED	118	66	1	1	B-ED	728	370
	256	1	1	B-ED	126	70	2	1	B-2ED	**865**	440
							2	1	V-2ED	955	**432**
	15360	5	1	B-ED	4449	2436	6	3	VP-ED	35790	34551

We see that conductors with multiple prime divisors ($N > 1$) only improve performance for sufficiently large messages and large exponents. Using prime powers ($t > 1$) does not generally improve performance, but is necessary to handle messages that are larger than the fundamental discriminant. In that case, the smallest required value of t was optimal. Among the cryptosystem variations, the basic version was optimal when using short exponents and/or small messages, while the small ciphertext variation and some of the CRT modifications came out on top when using full exponents and larger security levels and messages.

5.2 Comparison to Paillier and Bresson et al.

We next compare the best versions of the Castagnos and Laguillaumie cryptosystem to the Paillier [11] and Bresson et al. [3] schemes. Paillier mentioned two encryption schemes in his paper [11] and presented CRT improvements for both decryption routines. We implemented the schemes along with their CRT improvements and observed that *Scheme 1* with CRT gives the best encryption and decryption results with small message sizes and the best decryption result with large message sizes. *Scheme 3* with CRT gives the best encryption results with large message sizes. Since the BCP scheme is also based on the DDH problem, we have two versions of the BCP scheme as well, one with full domain exponents and the other with short exponents. Note that contrary to BCP, Paillier encryption performs operations with the message as an exponent. Thus, for a fixed security level, we expect that Pailler encryption times should vary slightly with different message sizes, whereas the other operations should remain relatively constant.

We compare below these results with those of the best results from the Castagnos and Laguillaumie variants in Tables 3, 4 and 5. For the Castagnos and Laguillaumie timings, we list the best observed encryption and decryption times amongst all the variants we implemented. Note that for the largest message spaces, no single variant results in both optimal encryption and decryption; in practice, we recommend the version with faster decryption, as the difference in encryption times relative to the optimal version is much smaller than the corresponding difference between decryption times.

At the 128-bit security level, Paillier was the fastest when using full exponents and BCP was the fastest for short exponents, for all message sizes considered. BCP was fastest for short exponents at the 192-bit level, while the Castagnos and Laguillaumie variants were superior when using full exponents for 16- and 80-bit messages; the results were mixed for larger messages. At the 256-bit security level, Castagnos and Laguillaumie variants are fastest for the three smallest message sizes when using full and short exponents. Among the Castagnos and Laguillaumie cryptosystem variations under consideration here, the basic version from [5] proved to be the best for small messages and exponents, but other variations and conductor decompositions were advantageous once the messages and exponents were sufficiently large.

Table 3. Summary of best performance (in ms)—128-bit security

Message	System	Short exponents		Full exponents	
		Encryption	Decryption	Encryption	Decryption
16	Pai			37	12
	BCP	7	3	147	73
	CL	14	9	58	28
80	Pai			38	12
	BCP	7	3	147	73
	CL	14	9	63	33
256	Pai			40	12
	BCP	7	3	147	73
	CL	18	11	96	50
3072	Pai			74	12
	BCP	7	3	145	72
	CL	156	98	964	794

Table 4. Summary of best performance (in ms)—192-bit security

Message	System	Short exponents		Full exponents	
		Encryption	Decryption	Encryption	Decryption
16	Pai			376	128
	BCP	38	18	1508	754
	CL	47	27	223	115
80	Pai			381	129
	BCP	38	18	1508	754
	CL	47	27	249	128
256	Pai			393	129
	BCP	38	18	1508	754
	CL	54	31	**320**	166
				328	**160**
7680	Pai			**745**	254
				755	**129**
	BCP	38	18	1487	743
	CL	871	508	7276	6702

Table 5. Summary of best performance (in ms)—256-bit security

Message	System	Short exponents		Full exponents	
		Encryption	Decryption	Encryption	Decryption
16	Pai			2069	753
	BCP	146	77	8306	4154
	CL	116	65	672	342
80	Pai			2079	752
	BCP	146	77	8298	4152
	CL	116	65	728	370
256	Pai			2104	751
	BCP	146	77	8295	4151
	CL	126	70	**865**	440
				955	**432**
15360	Pai			**4072**	1475
				4125	**751**
	BCP	141	74	8170	4087
	CL	4449	2436	35790	34551

6 Further Work

Our results show that, as expected, the Castagnos and Laguillaumie cryptosystem has some performance advantages as compared to Paillier and BCP for small messages and at high security levels. The variations described in this paper provide improvements when large exponents and message sizes are used.

One further optimization that could be considered to improve the extended versions is to take advantage of the fact that sufficiently small prime divisors of the conductor can be handled without multiprecision. This was not done in our experiments and could potentially make these versions more competitive.

We remark that the short exponent versions of the cryptosystems, as expected, are quite efficient. It would be of interest to revise and complete the security proofs in this context, where the intractability assumptions are all replaced by their short exponent analogues.

References

1. Biasse, J.-F., Jacobson Jr., M.J., Silvester, A.K.: Security estimates for quadratic field based cryptosystems. In: Steinfeld, R., Hawkes, P. (eds.) ACISP 2010. LNCS, vol. 6168, pp. 233–247. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14081-5_15
2. Bosma, W., Stevenhagen, P.: On the computation of quadratic 2-class groups. J. Théor. Nombres Bordeaux **8**(2), 283–313 (1996). http://jtnb.cedram.org/item?id=JTNB_1996__8_2_283_0

3. Bresson, E., Catalano, D., Pointcheval, D.: A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 37–54. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-40061-5_3
4. Castagnos, G., Imbert, L., Laguillaumie, F.: Encryption switching protocols revisited: switching modulo p. In: Katz, J., Shacham, H. (eds.) CRYPTO 2017. LNCS, vol. 10401, pp. 255–287. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63688-7_9
5. Castagnos, G., Laguillaumie, F.: Linearly homomorphic encryption from $\mathsf{DDH} - \mathsf{DL}$. In: Topics in Cryptology - CT-RSA 2015, The Cryptographer's Track at the RSA Conference 2015, San Francisco, CA, USA, 20–24 April 2015. Proceedings, pp. 487–505 (2015). https://doi.org/10.1007/978-3-319-16715-2_26
6. Cohen, H.: A Course in Computational Algebraic Number Theory. Graduate Texts in Mathematics, vol. 138. Springer, Berlin (1993). https://doi.org/10.1007/978-3-662-02945-9
7. Hamdy, S., Möller, B.: Security of cryptosystems based on class groups of imaginary quadratic orders. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 234–247. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44448-3_18
8. Hühnlein, D., Jacobson Jr., M.J., Paulus, S., Takagi, T.: A cryptosystem based on non-maximal imaginary quadratic orders with fast decryption. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 294–307. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0054134
9. Jacobson Jr., M.J., Williams, H.C.: Solving the Pell Equation. CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC. Springer, New York (2009). https://doi.org/10.1007/978-0-387-84923-2
10. Koshiba, T., Kurosawa, K.: Short exponent Diffie-Hellman Problems. In: Bao, F., Deng, R., Zhou, J. (eds.) PKC 2004. LNCS, vol. 2947, pp. 173–186. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24632-9_13
11. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_16
12. Sayles, M.: Optarith and qform libraries for fast binary quadratic forms arithmetic (2013). http://github.com/maxwellsayles