A note on the construction of pairing-friendly elliptic curves for composite order protocols

Sorina Ionica[1] and Malika Izabachène[2]

[1] Laboratoire MIS, Université de Picardie Jules Verne, 33 Rue Saint Leu 80000 Amiens, France
[2] CEA LIST, 91191 Gif-sur-Yvette Cedex, France
sorina.ionica@u-picardie.fr,malika.izabachene@cea.fr

Abstract. In pairing-based cryptography, the security of protocols using composite order groups relies on the difficulty of factoring a composite number N. Boneh *et al* proposed the Cocks-Pinch method to construct ordinary pairing-friendly elliptic curves having a subgroup of composite order N. Displaying such a curve as a public parameter implies revealing a square root s of the complex multiplication discriminant $-D$ modulo N. We exploit this information leak and the structure of the endomorphism ring of the curve to factor the RSA modulus, under certain conditions. Our conclusion is that the values of s modulo each prime in the factorization of N should be chosen as high entropy input parameters when running the Cocks-Pinch algorithm.

Keywords : composite order group, integer factorization, elliptic curve, endomorphism, Coppersmith's algorithm

1 Introduction

Pairings on elliptic curves are widely used in cryptography, due to their application in protocols such as the tripartite Diffie-Hellman [27], identity based encryption [7] and short signatures [17]. Bilinear groups of composite order have been used as a convenient tool to provide plenty of cryptosystems with advanced functionalities. For example, they can be used to build homomorphic encryption [8], to instantiate zero-knowledge proof systems [23], group signatures in the standard model [11], traitor tracing with full traceability [10], functional encryption with full security ([34, 12] for example). From the practical point of view, prime order groups are preferable but the security proofs are in general more complex. Several papers studied general techniques for converting composite-order constructions into prime order groups constructions with the same functionalities [13, 14, 33, 18, 39, 38]. However, conversions do not work mechanically [25, 13, 44, 40, 35]. And although the operations over composite order groups are expensive, they remain a very convenient tool for proving security of cryptographic constructions with advanced functionalities. Indeed, several schemes were firstly conceived using composite order before being transformed in the prime order setting.

While the security of prime order group protocols rely on discrete logarithm-based security assumptions, in the composite order setting, the security is based on the hardness of factoring the modulus N giving the cardinality of the elliptic curve group used in the protocol implementation. In order to construct pairing-based schemes with composite order groups, we can either implement the pairing using supersingular elliptic curves as in the original construction in [8] or use ordinary curves as suggested by Boneh *et al* [9]. Ordinary curves are particularly recommended if the security additionally relies on the Decisional Diffie-Hellman (DDH) assumption. Moreover, a line of papers [2, 21, 24, 28] suggests that the discrete logarithm problem on supersingular curves is weak. In particular, recent announcements (see [22] for instance) show that pairings of Type 1 as defined in [20] are no longer safe.

In this paper, we investigate the security of composite order group protocols that use ordinary elliptic curves. As shown by Boneh *et al.* [9], in order to construct pairing over composite order groups from such curves, the Cocks-Pinch method can then be employed [15]. However, Boneh *et al.* point out several security issues that occur in this setting. In particular, they noticed that given a public ordinary curve, whose number of points is divisible by N and can be counted in polynomial time, one may easily compute a square root s of the CM discriminant $-D \bmod N$.

Our analysis follows the work of [9] and starts from the complex multiplication (CM) construction of pairing friendly elliptic curves whose number of points is divisible by N. First, we show that one may use Coppersmith's techniques for finding small roots of modular polynomial equations to factor N.

Let p_1 be one of the large prime factors of N and denote by λ the integer such that $\lambda \equiv s \pmod{p_1}$ and $0 \leq \lambda < p_1$. We show how to recover λ in the case where $\lambda \approx \lfloor \sqrt{p_1} \rfloor$ and then compute $p_1 = \gcd(N, s - \lambda)$. While à priori the case $\lambda \approx \lfloor \sqrt{p_1} \rfloor$ arises with negligible probability, this may happen when the input parameters of the Cocks-Pinch method are mistakenly chosen to be small.

Secondly, we relate the square root s to an endomorphism on the elliptic curve which corresponds to $\sqrt{-D}$ under the isomorphism $\mathrm{End}(E) \simeq \mathbb{Z}[\frac{D+\sqrt{-D}}{2}]$. Assuming that a generator P_1 of the subgroup G_1 of order p_1 in G is given as a public value and that the discrete logarithm in this group is easy, we show how we can recover λ using an efficiently computable endomorphism ϕ of the elliptic curve.

Organisation. This paper is organised as follows. Section 2 reviews the Cocks-Pinch method and its generalization to the case of a composite modulus. Section 3 briefly presents the Coppersmith technique to find small roots of polynomials. Section 4 identifies weak instances of the Cocks-Pinch construction and shows how to recover part of the factorization of N. Section 5 gives the complexity of our method, and explains how to choose the parameter λ to avoid constructing weak instances of the Cocks-Pinch algorithm.

2 Background on pairings and composite order protocols

2.1 Composite order schemes

In this section, we specify the general framework of our analysis and review the schemes which work in this setting. Let G be a group of order a composite modulus $N = p_1 \cdots p_\tau$, where $p_1, \ldots, p_\tau$ are distinct primes. We denote by G_i (resp. H_i) the subgroup of G (resp. H) of order p_i. To simplify, we focus on systems implemented using $\tau = 2$. We further explain in Section 5 how security is impacted in the case where $\tau > 2$.

More precisely, we consider composite order schemes implemented using a group G which is a subgroup of an ordinary elliptic curve. This elliptic curve verifies the following conditions:

1. The number of points on the curve is known and divisible by N.
2. The curve admits an efficient pairing $e : G \times H \mapsto G_T$, with all groups G, H, G_T of order N.

The best known example of a scheme in this setting is the BGN encryption scheme [8]. For completeness, we recall here this scheme in the asymmetric pairing setting.

Let G_1 be the subgroup of G of order p_1 and P, P_1 random generators of G, G_1 respectively. To encrypt a message m with public key (G, N, P, P_1) and secret key p_1, one computes $C = mP + rP_1$, for a random scalar $r \in \mathbb{Z}_N$. To decrypt, one computes $\tilde{P} = p_1 P$ and $\tilde{C} = p_1 C$ and then gets m as the logarithm of $\tilde{C}$ with respect to $\tilde{P}$.

The security of the BGN encryption scheme relies on the Subgroup Decision Problem (SDP) [8], which is a particular instance of the General Subgroup Decision family of assumptions introduced in [3]. The Subgroup Decision Problem states that given $(N, G \cong G_1 \times G_2, G_1, P)$, with P a random element of G, it is hard to decide whether P is in G_1 or not. The hardness of SDP is proven in the generic model assuming the order of the subgroup is unknown [29]. To the best of our knowledge, no result has been proven so far on the reverse statement in a non-generic model.

2.2 The Cocks-Pinch method

Pairings on elliptic curves, i.e. the Weil and the Tate pairing map to the multiplicative group of an extension field of $\mathbb{F}_q$. This extension field needs to contain the N-th roots of unity. We denote by k the embedding degree with respect to N, i.e. the smallest integer such that the extension field $\mathbb{F}_{q^k}$ contains the N-th roots of unity. Pairings are usually computed by using Miller's algorithm, whose efficiency depends on an efficient arithmetic for $\mathbb{F}_{q^k}$. Therefore, we use elliptic curves for which the embedding degree k is relatively small.

Writing the Frobenius endomorphism π as an element of the endomorphism ring leads to the following condition

$$\exists\ y \in \mathbb{Z} \quad \text{s.t.} \quad 4q = t^2 + y^2 D, \tag{2}$$

where D is the complex multiplication discriminant for the elliptic curve and t is the trace of the Frobenius endomorphism. If this condition is satisfied, the CM method outputs a curve whose number of points is $\#E(\mathbb{F}_q) = q + 1 - t$. Cocks and Pinch [6] proposed an algorithm which, given a prime number N and an integer k, constructs a pairing friendly elliptic curve such that $N|\#E(\mathbb{F}_q)$ and whose embedding degree with respect to N is k. This method was extended by Boneh, Rubin and Silverberg [9] to the case where N is a composite number. We briefly recall the method in Algorithm 1.

Algorithm 1 The Cocks-Pinch algorithm

Require: k, r prime numbers p_i, λ_i, D and $N = p_1^{\alpha_1} \dots p_r^{\alpha_r}$ s.t. $\lambda_i^2 = -D \pmod{p_i}$.
Ensure: q, t such that there is a curve with CM by $-D$ over $\mathbb{F}_q$ with $q + 1 - t$ points where $N|(q + 1 - t)$ and $N|(q^k - 1)$, and N does not divide $q^j - 1$ for $j < k$.
1: Choose an integer X which is a primitive k-th root of unity modulo every $p_i^{\alpha_i}$.
2: Compute using the CRT theorem $s \pmod{N}$ such that $s^2 = -D \pmod{N}$.
3: Take an integer Y congruent to $\pm(X-1)s^{-1} \pmod{N}$.
4: **repeat**
5: $\quad q \leftarrow ((X+1)^2 + D(Y + jN)^2)/4$
6: $\quad j \leftarrow j + 1$
7: **until** q is prime
8: **return** q and $t = X + 1$

One may use the Chinese Remainder Theorem to obtain X of order k modulo all the $p_i^{\alpha_i}$'s, $i \in \{1, \dots, r\}$, in Step 1. Similarly, Step 2 computes $s^2 = -D \pmod{N}$ by using the Chinese Remainder Theorem. Moreover, assume that D is a divisor of k such that either k is a multiple of 4 and D divides $k/4$ or k is not a multiple of 4 and $D \equiv 3 \pmod 4$. Boneh et al. [9] note that then one computes a square root of $-D$ by the formula:

$$s = \begin{cases} \sum_{\substack{a=1 \\ (a,2D)=1}}^{2D-1} (\frac{-D}{a}) X^{\frac{ak}{D}} & \text{if } D \equiv 3 \pmod 4, \\ \\ \frac{1}{2} \sum_{\substack{a=1 \\ (a,2D)=1}}^{4D-1} (\frac{-D}{a}) X^{\frac{ak}{4D}} & \text{otherwise.} \end{cases} \tag{3}$$

2.3 Implementation and choice of subgroups on ordinary curves

In this section, we are interested in pairing implementation on ordinary curves.

On an ordinary curve, if the embedding degree k is greater than 1, it is most efficient to choose to implement the pairing $e : G \times H \to G_T$ by choosing

$$G = E[N] \cap \mathrm{Ker}(\pi - [1]) \text{ and } H = E[N] \cap \mathrm{Ker}(\pi - [q]). \tag{4}$$

Indeed, when we compute a pairing of elements in these subgroups, many operations in Miller's algorithm can be done in subfields of $\mathbb{F}_{q^k}$, which critically

reduces the cost of the pairing computation (see [31]). When implementing composite order protocols, if we need to ensure that the DDH assumption holds in both G and H, then we need to choose these subgroups such that there are no *distortion maps* for points in these subgroups. Given a point $P \in E[N]$, a distortion map for P is a map ϕ such that for a point P, $\phi(P) \notin \langle P \rangle$. The following result was given by Verheul [43], whose purpose was to investigate the existence of distortion maps for points of order N.

Theorem 1. *Let E be an ordinary curve defined over $\mathbb{F}_q$ and let P be a point over E of prime order $N \neq q$. Suppose the embedding degree k is greater than 1 and denote by Q a point defined over $\mathbb{F}_{q^k}$, such that $\pi(Q) = qQ$. Then P and Q are eigenvectors of any other endomorphism of E.*

We will show later in Section 4 that assuming we are given an efficiently computable endomorphism of the curve and that G and H are chosen as in Equation (4), then if one can solve the SDP problem, one can reduce the hardness of factoring the modulus N to that of solving a discrete logarithm problem in a subgroup of order p_1.

3 Coppersmith's method

In 1996, Coppersmith [16] introduced lattice-based reduction techniques to find small roots of polynomials. These techniques have been reformulated in a simplified manner [26] and also generalized to more variables and have become a powerful cryptanalytic toolbox. In this paper, we will use Coppersmith's technique to find small roots of a polynomial equation $F(t) = 0 \mod p_1$ where F is a monic polynomial of degree 1 and p_1 is unknown and a factor of a RSA modulus N.

For a vector $x \in \mathbb{R}^n$, with $n > 0$, we denote by $\|x\|$ the Euclidian norm. We rely on the following results.

Lemma 1. *[26] Let $F \in \mathbb{Z}[t]$ be a polynomial of degree d. Let $p_1, h, X \in \mathbb{N}$ and let b_F be the vector given by the coefficients of the real polynomial $F(tX) \in \mathbb{R}[t]$. Assume that $F(t_0) = 0 \pmod{p_1^h}$ where $|t_0| \leq X$ and that $\|b_F\| < \frac{p_1^h}{\sqrt{d+1}}$. Then $F(t_0) = 0$ holds over the integers.*

Theorem 2. *[19, Theorem 19.4.2] Let $N = p_1 p_2$ with $p_1 < p_2 < 2p_1$. Let $0 < \epsilon < 1/4$ and let $F(t)$ be a degree 1 polynomial in $\mathbb{Z}[t]$. If t_0 is a small root of F satisfying $|t_0| \leq \frac{1}{2\sqrt{2}} N^{1/4-\epsilon}$, then one can recover p_1 in polynomial time in $\log N$ and $1/\epsilon$.*

In the following, we recall how the algorithm announced in Theorem 2 works for $F(t) = t + a$, when we want to solve $F(t) = 0 \pmod{p_1}$. First, we collect a set of small polynomials defined by $g_{u,v}(t) = N^{h-u} F(t)^u t^v$ where $0 \leq u \leq h-1$ and $v \geq 0$. The parameter h will be determined later.

We represent polynomials as row vectors and use a lattice reduction algorithm to find small vectors in a lattice. Let $\kappa = 2h$ and $X = \lfloor N^{1/4-\epsilon} \rfloor$. We construct

the $\kappa + 1$-dimensional lattice L spanned by the vectors of coefficients of the following polynomials:

- $g_{u,0}(tX)$ for $u = 0, \ldots, h-1$
- $g_{h,v}(tX)$ for $v = 0, \ldots, \kappa - h$.

The determinant of the lattice is then given by $\det(L) = N^{h(h+1)/2} X^{h(2h+1)}$. The condition of Lemma 1 ensures that if the first vector b_1 of the LLL reduced basis satisfies $\|b_1\| \leq p_1^h/\sqrt{\kappa+1}$, the corresponding polynomial will have a root over $\mathbb{Z}$.

Since $\|b_1\| \leq 2^{\kappa/4} \det(L)^{1/(\kappa+1)}$, it is sufficient to have the following condition

$$2^{\kappa/4} \det(L)^{1/\kappa+1} < p_1^h/\sqrt{\kappa+1}. \tag{5}$$

By using the LLL algorithm and root finding algorithms for polynomials over the integers, one can find $|t_0| \leq X$ in polynomial time in $(\log\ N, \frac{1}{\epsilon})$.

Now, by substituting the bound for X in Equation (5) with $\kappa = 2h$, we obtain that if h satisfies $\frac{1}{4(2h+1)} < \epsilon$, then we are guaranteed to find t_0.

This result extends also to the case where N has more than two factors. In this case, the success condition could be expressed in a similar way as follows (see [19, Proof of Th.19.4.2]): if $p_1 = N^\alpha$ and $|t_0| < N^\beta$, then the size of h giving the dimension of the lattice used by Coppersmith's algorithm is given by:

$$\frac{h(h+1)}{2} + \frac{\beta\kappa(\kappa+1)}{2} < \alpha h(\kappa+1),$$

where $h = \sqrt{\beta}\kappa$.

4 Finding weak instances

Boneh *et al* [9] note that in the case of a RSA modulus N, a square root $s \pmod N$ is leaked during the Cocks-Pinch construction. This is a potential security concern, since computing all square roots modulo a composite is as hard as factoring. In this section, we show that, under certain conditions, when a square root of $s \pmod N$ is revealed from q, N and E, one may use this information to factor N. As explained before, we present our method in the case where $N = p_1 p_2$.

Let us begin by explaining how to recover s. An elliptic curve over $\mathbb{F}_q$ will have $\#E(\mathbb{F}_q) = q+1-t$. Since the curve is public, an attacker may use Schoof's point counting algorithm to get t. Thus he may compute Y in Step 3 of the Cocks-Pinch algorithm by getting the square root of $\frac{t^2-4q}{D}$ (assuming D is known, since the curve is public). If $(N, Y) = 1$, he will then get $s = (X-1)Y^{-1} \pmod N$, where $X = t - 1$. Note that if $k = 1$, we have $X = 1 \pmod N$ and $Y = 0 \pmod N$. Hence s may not be recovered from E.

In the remainder of this paper, the embedding degree k is greater than 1. We denote by λ the integer such that

$$\lambda = \pm s \pmod{p_1}.$$

and $\lambda < p_1$. Hence if one recovers $\lambda \pmod{p_1}$, one may compute p_1 as $p_1 = \gcd(N, \lambda \mp s)$.
We give two toy examples computed with MAGMA.

Example 1. Let $p_1 = 1073741827$ and $p_2 = 1074790447$. We take $N = p_1 p_2$ and $D = 3$. With $\lambda \equiv 32768 \pmod{p_1}$ and $\lambda' \equiv 547381745 \pmod{p_2}$, the CRT theorem gives

$$s = 211557198247737806 \pmod{N}.$$

Using the Cocks-Pinch method, we obtain the following curve with embedding degree 2 with respect to N:

$$y^2 = x^3 + 13 \text{ defined over } \mathbb{F}_q, \tag{6}$$

with $q = 1140730183325927132841992508979589859787$. Since $\lambda < N^{1/4}$, one may apply the technique presented in Section 3 to find λ as a root of the polynomial $F(t) = t \pm s \pmod{p_1}$. Our implementation of Coppersmith's method using the fplll library [41] for lattice reduction recovers λ as a square root of the polynomial $F(t) = t - s \pmod{p_1}$ in approximatively 2 seconds, on a Intel Core i3-3227U at 1.90 GHz.

Remark 1. As explained in Section 2.2, in some cases a square root s of $-D \pmod{N}$ is given by equation (3). This value should obviously be used when running the Cocks-Pinch algorithm. Note that s is a sum of a small number of powers of X. If $X \pmod{p_1}$ is small, then $s \pmod{p_1}$ may also be small. Hence the output of the Cocks-Pinch algorithm will be a curve providing weak instances.

Example 2. [9, Example 6.2] Assume $k = D = 3$ and $p_1 = p_2 = 1 \pmod 3$. Then one must choose X such that $X^2 + X + 1 \equiv 0 \pmod{N}$. Assume $X \bmod p_1$ is small. Using equation (3), one may compute $s \equiv 2X + 1 \pmod{N}$, which verifies $s^2 \equiv -3 \pmod{N}$. Hence $\lambda \equiv s \pmod{p_1}$ is small and the Cocks-Pinch construction using these parameters will provide weak instances for composite order protocols.

These examples illustrate that if the size of λ is twice smaller than the size of p_1, then implementing composite order group schemes on curves constructed via the Cocks Pinch method is insecure. Normally, the fact that λ is small arises with negligible probability when setting input parameters for the Cocks-Pinch construction, unless it is chosen on purpose to be so. In the remainder of this section, we show how we handle the case of large λ.

Extending the algorithm to larger values of λ. We show that by using the technique presented in Section 4 combined with an exhaustive search, one may recover λ, when this is relatively large.

The setting is the following: we are given an asymmetric pairing group structure (G, H, G_T, P, Q) where P and Q have order N such that $N = p_1 p_2$ is an

RSA modulus and divides $\#E(\mathbb{F}_q)$. We assume that $E(\mathbb{F}_q)$ is constructed using Algorithm 1 and that its embedding degree with respect to N is k. We further assume that s can be recovered as explained in Section 4. Note that although the construction of $E(\mathbb{F}_q)$ uses the fact that the factorization of N is known, we assume that, when encrypting, the sender does not know the factorization of N. Now, assume that $p_1 < p_2$ and hence that $p_1 < N^{\frac{1}{2}}$. Let $w = \lfloor N^{\frac{1}{4}} \rfloor$ and write $\lambda = x + \lfloor N^{\frac{1}{4}} \rfloor y$, with $0 \leq x, y < w$. Note that x and y are unique with this property. We will search for x and y verifying $\lambda = x + \lfloor N^{\frac{1}{4}} \rfloor y$, $0 \leq x, y < w$. Note that the polynomial $F_y(X) = X + y\lfloor N^{\frac{1}{4}} \rfloor \pm s$ has a small root. Thus, if we know y, we can find this root using Coppersmith's method for finding small roots of polynomials over $\mathbb{Z}$. To find y, we perform exhaustive search in an interval $[0, w']$, with $w' < w$. Indeed, let y' be any integer in the interval $[0, w']$. Assuming that Coppersmith's algorithm will return a root x' for a given y', we verify that $\lambda = x' + y'[N^{1/4}]$ is a valid solution by checking whether $\gcd(N, (x' + y'\lfloor N^{1/4} \rfloor \pm s))$ is non-trivial. Heuristically, this almost never happens, unless $y = y'$ (the probability for a random polynomial with coefficients of size N to have a root of size bounded by $\lfloor N^{1/4} \rfloor$ is negligible). This procedure is summarized in Algorithm 2, where $\text{Coppersmith}(y, s, N)$ denotes Coppersmith's method for finding a root for the polynomials F_y.

Algorithm 2 Recovering the value of λ

Require: An elliptic curve E defined over $\mathbb{F}_q$, $N|\#E(\mathbb{F}_q)$, (G, G_1, P, P_1), a parameter s, an endomorphism ϕ, integers $w' \leq w$.
Ensure: λ such that $\lambda \equiv s \pmod{p_1}$.
1: $y \leftarrow 0$, $x \leftarrow 1$
2: **while** $\gcd(N, x + y\lfloor N^{1/4} \rfloor \pm s) \neq 1$ **do**
3: $\quad y \leftarrow y + 1$
4: $\quad x \leftarrow \text{Coppersmith}(y, s, N)$
5: **end while**
6: **return** $\lambda = x + y\lfloor N^{1/4} \rfloor$

Note that in practice this method is interesting only if the search space $[0, w']$ is small. Indeed, if $w' = w$, then this algorithm is exponential in p_1 and thus slower than known subexponential algorithms for factoring. In Section 5 we give an asymptotic complexity analysis and compare our approach with classical factoring methods.

Example 3. We consider $p_1 = 2171163061$, $p_2 = 8684652439$, $N = p_1 p_2$ and $D = 3$. Then $s = 58007414937220612199598481252$ is a root of $-D$ modulo N. We have $\lfloor N^{\frac{1}{4}} \rfloor = 65896$ and we search for $x, y \in [0, 65896[$. For $y = 26812$, Coppersmith's method finds $x = 13851$ a small root of the polynomial $F(t) = t + 65896y - s \pmod{p_1}$.

Using endomorphisms to recover λ. Here we provide further evidence on the relationship between discrete-log type based assumptions and factorization in the composite order setting, in the case where the group G is a group coming from an elliptic curve.

Let P_1 be an element of order p_1 which is part of the public key. It is obvious that if one can solve the discrete logarithm problem in G, then he can recover N/p_1 since $P_1 = (N/p_1)P$. We denote by ϕ the endomorphism whose characteristic equation is $\phi^2 + D = 0$. For $P_1 \in G_1$, we have $\phi(P_1) = \lambda P_1$, with $\lambda < p_1$. Note that if D is small, the endomorphism ϕ can be computed very efficiently, by using Vélu's formulae [42]. Details on this computation and its complexity are given in Section 5. For example, the endomorphism ϕ corresponding to $\sqrt{-3}$ on the curve given by Equation (6) is given by the equation

$$\phi(x,y) = \left(\frac{x^3 + 52}{x^2}, \frac{x^3 y + 1140730183325927132841992508979589859683 y}{x^3} \right).$$

Given a point P_1 in a subgroup of order p_1, one may then compute $\phi(P_1)$ with a few operation in the finite field $\mathbb{F}_q$. At this point, since P_1 is public, the attacker may easily compute $\phi(P_1)$. Hence if the attacker can solve the discrete logarithm problem in the group G_1, then he recovers $\lambda \pmod{p_1}$ and computes p_1 as $p_1 = \gcd(N, \lambda \pm s)$.

5 Complexity analysis

The fastest algorithm for factoring a composite modulus N is the number field sieve algorithm (NFS), whose complexity is given by

$$L[N] = e^{1.923(\log N)^{1/3}(\log\log N)^{2/3}}.$$

However, in order to find small factors of a composite number, it is more efficient to use the ECM method. The expected time required in order to find a factor p of a composite number N is then

$$E[N,p] = (\log N)^2 e^{\sqrt{2\log p \log\log p}}.$$

We use formulas in Lenstra's paper [32] to compute the size of N and of each of its factors, for 80 bit and 128 bit security levels. We also took into account that the current record for factoring with ECM is finding a factor of 79 digits (see [45]). As a consequence, we take all factors of N of size at least 300 bits, in order to prevent attacks with ECM. Our computations, for composite numbers with 2, 3 and 4 factors, are summarized in Table 1. The column EX in this table gives the size of y such that $\lambda = x + y\lfloor N^{\frac{1}{4}} \rfloor$. For completeness, we also add corresponding sizes for discrete logarithms in G_T, computed using the NSF algorithm (see for instance [1])[1].

[1] Note that more recent attacks in composite degree extension fields [30] do not apply here, since for our purposes it suffices to consider prime degree extension fields.

Table 1. Composite number size (in bits) for a fixed security level

	80 bit security level				128 bit security level			
Number of prime factors	RSA		EX	DLP	RSA		EX	DLP
	$\log p_i$	$\log N$	$\log y$	DLP	$\log p_i$	$\log N$	$\log y$	DLP
2	512	1024	256	1024	1612	3224	806	3072
3	341	1024	170	1024	1045	3155	522	3072
4	300	1200	150	1024	811	3244	405	3072

For cryptographic sizes, computing the number of points on the elliptic curve (i.e. $\#E(\mathbb{F}_q) = q+1-t$) by using the SEA algorithm has logarithmic complexity in q and is performed with MAGMA 2.15-15 within seconds on a Intel Core i3-3227U Processor (3M Cache, 1.90 GHz).

Finding small roots via Coppersmith's method. The complexity of Coppersmith's algorithm depends mainly on the running time of the LLL algorithm. We briefly recall that the complexity for Coppersmith's lattice-based algorithm is upper bounded by $O\left(d^5(d+\beta)\beta\right)$ if one uses the Nguyen-Stehlé L^2 algorithm [36] and by $O(d^{5+\varepsilon}\beta + d^{\omega+1+\varepsilon}\beta^{1+\varepsilon})$ (for any $\varepsilon > 0$) if one uses Novocin *et al*'s L^1 algorithm [37], where d is the dimension of the lattice, β is the maximal bit-size of an entry in the input basis, and ω is the matrix multiplication exponent.

Altogether, this gives an asymptotical running time of $O(h^7(\log N)^2)$ using L^2 and $O(h^{5+\epsilon}\log N)$ using $\tilde{L}^1$ for our $(2h+1)$-dimensional lattice with coefficient size bounded by $\log N^{\frac{5h}{4}}$.

Table 2 gives the maximal size of the small root we can compute, depending on the choice of the size of h the number of factors of N for $\epsilon = 1/8h$. The formulas for ϵ are taken from [19] as explained in Section 3.

Table 2. Choice of parameters

Number of prime factors	h	size of p_1	size of root of degree 1 pol.
2	15	2^{512}	2^{247}
2	25	2^{512}	2^{250}
3	15	2^{342}	2^{102}
3	25	2^{342}	2^{114}
4	25	2^{300}	2^{75}

Implementations in the literature have shown that at the 80 bits security level, i.e. a modulus N of 1024 bits and h chosen as in Table 2, Coppersmith's

method finds a small modular root within seconds. The reader is referred for instance to [5, 4].

If the RSA modulus has three factors or more, we compare the complexity of our algorithm to that of the ECM method. If $m > 2$, Coppersmith's method for finding roots of polynomials $\pmod{p_1}$ only allows to recover x in an interval of length smaller than $p^{1/2}$. For example, for a composite number with 3 factors at the 80 bit security level, Coppersmith's algorithm recovers values of λ with approximatively 114 bits. On the other hand, note that for a fixed security level, the higher the number m of factors of N is, the smaller the search space for the exhaustive search part of our algorithm is.

Our analysis here relies on asymptotic timing of the LLL algorithm. Further implementation work would allow us to determine the constant hidden in the LLL algorithm complexity for target instances and estimate the size of w' for which Algorithm 2 yields an attack faster than NFS at a given security level. Note also that this exhaustive search algorithm could easily be parallelized, hence such an investigation should also take into account the amount of hardware available. We leave this as topic for future work. From our study, we can say that when running the Cocks-Pinch algorithm for constructing composite order elliptic curves, one should take λ as an input parameter with maximal entropy, i.e. 512 bits when working with a two factors RSA modulus at 80 bits security level. To the best of our knowledge, this choice has no impact on the efficiency of the protocol implementation and is by far the safest solution.

6 Conclusion

Computing pairing friendly ordinary curves for implementing composite order group protocols can be done by using the Cocks-Pinch method. When running a protocol using a curve constructed with this method, one reveals a square root of the CM discriminant $-D$ modulo the composite modulus $N = p_1 p_2$. We show that if the square root λ of $-D \pmod{p_1}$ does not have maximal entropy, one may use Coppersmith's algorithm combined with exhaustive search to find this value and thus to factorize N. Our conclusion is that extra precautions should be taken when choosing the RSA modulus N. More precisely, λ should be given as an input parameter of the Cocks-Pinch algorithm.

References

1. Key length recommendations. `http://www.keylength.com`.
2. G. Adj, A. Menezes, T. Oliviera, and F. Rodríguez-Henríquez. Weakness of $\mathbb{F}_{3^{6\cdot 509}}$ for Discrete Logarithm Cryptography. *Finite Fields and Their Applications*, 32:148–170, 2015.
3. M. Bellare, B. Waters, and S. Yilek. Identity-based encryption secure against selective opening attack. In *Proc. of* TCC'2011, volume 6597 of *LNCS*, pages 235–252. IACR, Springer-Verlag, 2011.

4. J. Bi, J.-S. Coron, J.-C. Faugère, G. Renault, and R. Zeitoun. Rounding and Chaining LLL:Finding faster small roots of univariate polynomial congruences. In *Proc. of PKC*, volume 8383 of *LNCS*, pages 185–202. Springer, 2014.
5. J. Bi and P. Nguyen. Rounding LLL:Finding faster small roots of univariate polynomial congruences, 2013.
6. I.F. Blake, G. Seroussi, and N.P. Smart. *Advances in Elliptic Curve Cryptography*, volume 317. Cambridge University Press, London Mathematical Society Lecture Note Series, 2005.
7. D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In *Proc. of Crypto'01*, volume 2139 of *LNCS*, pages 213–229. IACR, Springer-Verlag, 2001.
8. D. Boneh, E-J. Goh, and K. Nissim. Evaluating 2-DNF formulas on ciphertexts. In *Proc. of TCC 2005*, volume 3378 of *LNCS*, pages 325–341. Springer, 2005.
9. D. Boneh, K. Rubin, and A. Silverberg. Finding composite order ordinary elliptic curves using the Cocks-Pinch method. *In* Journal of Number Theory, 131(5):832–841, 2011. `http://http://eprint.iacr.org/2009/533`.
10. D. Boneh, A. Sahai, and B. Waters. Fully Collusion Resistant Traitor Tracing with Short Ciphertexts and Private Keys. In *Proc. of Eurocrypt 2006*, volume 4004 of *LNCS*, pages 573–592. IACR, Springer-Verlag, 2006.
11. X. Boyen and B. Waters. Full-domain subgroup hiding and constant-size group signatures. In *Proc. of* PKC'07, volume 4450 of *LNCS*, pages 1–15, 2007.
12. A. Caro, V. Iovino, and G. Persiano. Fully secure anonymous HIBE and secret-key anonymous ibe with short ciphertexts. In *Proc. of* Pairing'10, volume 6487 of *LNCS*, pages 347–366. IACR, Springer-Verlag, 2012.
13. J. Chen, R. Gay, and H. Wee. Improved dual system ABE in prime-order groups via predicate encodings. In *Proc. of Eurocrypt*, LNCS, pages 595–624. Springer, 2015.
14. J. Chen and H. Wee. Dual system groups and its applications - compact HIBE and more. Cryptology ePrint Archive, Report 2014/265, 2014. `http://eprint.iacr.org/2014/265`.
15. C. Cocks and R.G.E. Pinch. Id-based cryptosystems based on the weil pairing. Unpublished manuscript, 2001.
16. D. Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *Journal of Cryptology*, 10(4):233–260, 1997. Revised version of two articles from Eurocrypt '96.
17. B. Lynn D. Boneh and H. Shacham. Short signatures from the Weil pairing. In *Proc. of* Asiacrypt 01, volume 2248.
18. D. M. Freeman. Converting pairing-based cryptosystems from composite-order. In *Proc. of* Eurocrypt'10, volume 6110 of *LNCS*, pages 44–61. IACR, Springer-Verlag, 2010.
19. S. Galbraith. *Mathematics of Public Key Cryptography, Chapter 20*. Cambridge University Press, April 2012.
20. S.D. Galbraith, K.G. Paterson, and N.P. Smart. Pairings for cryptographers. Journal of Applied Mathematics, 156, numb.3113-3121, 2008.
21. F. Göloglu, R. Granger, G. McGuire, and J. Zumbrägel. On the function field sieve and the impact of higher splitting probabilities. In *Proc. of* Crypto'13, volume 8043 of *LNCS*, pages 109–128. IACR, Springer-Verlag, 2013.
22. R. Granger, T. Kleinjung, and J. Zumbrägel. Breaking 128-bit secure supersingular binary curves (or how to solve discrete logarithms in $\mathbb{F}_{2^{4\cdot 1223}}$ and $\mathbb{F}_{2^{12\cdot 367}}$). `http://eprint.iacr.org/2014/119`.

23. J. Groth and A. Sahai. Efficient non-interactive proof systems for bilinear groups. In *Proc. of* Eurocrypt'08, volume 4965 of *LNCS*, pages 415–432. IACR, Springer-Verlag, 2008.
24. T. Hayashi, T. Shimoyama, N. Shinohara, and T. Takagi. Breaking pairing-based cryptosystems using η_t pairing over $GF(3^{97})$, booktitle=Proc. of Asiacrypt'12, volume=7658, pages=46-60, year=2012, series = lncs, organization = iacr, publisher = springer.
25. D. Hofheinz, J. Koch, and C. Striecks. In *In Proc. Of PKC, pages= 799–822, title = Identity-Based Encryption with (Almost) Tight Security in the Multi-instance, Multi-ciphertext Setting., year = 2015,.*
26. N. Howgrave-Graham. Finding small roots of univariate modular equations revisited. In *Proceedings of Cryptography and Coding*, volume 1355 of *LNCS*, page 131–142. IACR, Springer-Verlag, 1997.
27. A. Joux. A one round protocol for Tripartite Diffie-Hellman. In *Proc. of ANTS'00*, volume 1838 of *LNCS*, pages 385–394. IACR, Springer-Verlag, 2000.
28. A. Joux. Faster index calculus for the medium prime case: application to 1175-bit and 1425-bit finite fields. In *Proc. of* Eurocrypt'13, volume 7881 of *LNCS*, pages 177–193. IACR, Springer-Verlag, 2013.
29. J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *Proc. of* Eurocrypt'08, volume 4965 of *LNCS*, pages 146–162. IACR, Springer-Verlag, 2008.
30. T. Kim and R. Barbulescu. Extended Tower Number Field Sieve: A new complexity for the medium prime case. 9814:543–571, 2016.
31. Neal Koblitz and Alfred Menezes. Pairing-based cryptography at high security levels. In *IMA Int. Conf.*, LNCS, pages 13–36. Springer, 2005.
32. A. K. Lenstra. Unbelievable security, matching AES security using public key systems. In *Proc. of* Asiacrypt 2001, volume 3796 of *LNCS*, pages 13–36. IACR, Springer-Verlag, 2005.
33. A.B. Lewko. Tools for simulating features of composite order bilinear groups in the prime order setting. In *Proc. of* Eurocrypt'12, volume 7237 of *LNCS*, pages 318–335. IACR, Springer-Verlag, 2012.
34. A.B. Lewko and B. Waters. New techniques for dual system encryption and fully secure HIBE with short ciphertexts. In *Proc. of* TCC'10, volume 5978 of *LNCS*, pages 455–479. IACR, Springer-Verlag, 2010.
35. S. Meiklejohn, H. Shacham, and D. Mandell Freeman. Limitations on transformations from composite-order to prime-order groups: The case of round-optimal blind signatures. In *Proc. of* Asiacrypt'10, volume 6477 of *LNCS*, pages 519–538. IACR, Springer-Verlag, 2010.
36. P.Q. Nguyen and D. Stehlé. An LLL algorithm with quadratic complexity. *SIAM J. Comput.*, 39(3):874–903, 2009.
37. A. Novocin, D. Stehlé, and G. Villard. An LLL-reduction algorithm with quasi-linear time complexity. In *Proc. STOC '11*. ACM, 2011.
38. T. Okamoto and K. Takashima. Homomorphic encryption and signatures from vector decomposition. In *In Proc. Of Pairing*, LNCS, pages 57–74. Springer, 2008.
39. T. Okamoto and K. Takashima. Hierarchical predicate encryption for inner-products. In *In Proc. Of Asiacrypt*, LNCS, pages 214–231. Springer, 2009.
40. J.H. Seo. On the (im)possibility of projecting property in prime-order setting. In *Proc. of* Asiacrypt'12, volume 7658 of *LNCS*, pages 61–79. IACR, Springer-Verlag, 2012.
41. D. Stehlé. The fplll library. `http://perso.ens-lyon.fr/damien.stehle/fplll/index.html`.

42. J. Vélu. In *Comptes Rendus De Academie Des Sciences Paris*, Serie I-Mathematique, Serie A., 1971.
43. E. R. Verheul. Evidence that xtr is more secure that supersingular elliptic curve cryptosystems. In *Proc. of* Eurocrypt'01, volume 2045 of *LNCS*, pages 195–201. IACR, Springer-Verlag, 2001.
44. H. Wee. Dual system encryption via predicate encodings. In *Proc. of TCC*, LNCS, pages 616–637. Springer, 2014.
45. P. Zimmermann. Top 50 ECM records. `http://www.loria.fr/~zimmerma/records/top50.html`.