Updating key size estimations for pairings

Razvan Barbulescu[1] and Sylvain Duquesne[2]

[1] IMJ-PRG, UMR CNRS 7586, Univ Paris 6, Univ Paris 7
razvan.barbulescu@imj-prg.fr
[2] IRMAR, UMR CNRS 6625, Univ Rennes 1
sylvain.duquesne@univ-rennes1.fr

Abstract. Recent progress on NFS imposed a new estimation of the security of pairings. In this work we study the best attacks against some of the most popular pairings. It allows us to propose new pairing-friendly curves of 128 bits and 192 bits of security.

Introduction

Pairing based cryptography has now many practical applications such as short signature schemes [BLS04], identity based cryptography [BF01] or broadcast encryption [BGW05]. A pairing is a non degenerate bilinear map

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_3.$$

It is usually realized thanks to elliptic curves. More precisely, the groups $\mathbb{G}_1$ and $\mathbb{G}_2$ are subgroups of an elliptic curve defined over a finite field $\mathbb{F}_q$ or one of its extensions and $\mathbb{G}_3$ is a subgroup of $\mathbb{F}_{q^k}^*$ where k is called the embedding degree. Of course a suitable pairing for cryptographic applications requires that the discrete logarithm problem is sufficiently difficult on these 3 groups. Because of recent attacks on the discrete logarithm problem in small characteristic finite fields [Jou14,BGJT14], it is now clear that prime base fields should be used. In this case, ordinary elliptic curves are the only ones allowing large embedding degrees.

The security of pairings defined over $\mathbb{F}_p$ having embedding degree k and group order r is determined by:

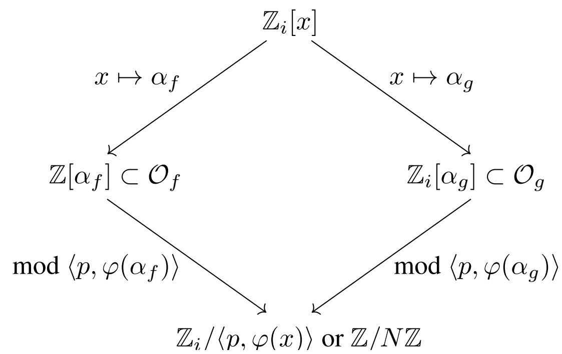
1. the cost of the discrete logarithm problem (DLP) on an order r subgroup of an elliptic curve defined over $\mathbb{F}_p$ (the curve side);
2. the cost of DLP in the multiplicative group of $\mathbb{F}_{p^k}$ (the finite field side).

The security evaluation on the curve side is simple: if s is the desired level of security, we select r such that $\log_2 r \geq 2s$ because of Pollard's rho algorithm (and by consequence $\log_2 p \geq 2s$). Attacks on the field side however are harder to estimate since the best algorithms belong to the Index Calculus family and their complexity is hard to write down explicitly. The goal of this paper is to give a precise evaluation of these algorithms in the pairing context, namely with parametrized parameters, and then to propose new parameters for pairing based cryptography ensuring the right security level.

Roadmap. After explaining the necessity of a new and more precise evaluation of key sizes in Section 1 we recall the most popular families of pairings in Section 2 and identify the best variant of NFS that an attacker can use against these families in Section 3. The proposition of new curves is done in three steps: first we study NFS and find what are the field sizes which correspond to 128 bits of security (Section 4), then we search for curves of this size (Section 6) and finally we do an analysis even more precise than before for each of the curves we propose (section 7). We end with a comparison of the estimated complexity of an optimal Ate pairing for these new curves (Section 8) and we present curves for higher security (192-bits security level, subgroup secure curves) in Section 9.

1 Big lines of NFS and a simple estimation of complexity

Whether the goal is to factor a composite integer N or to compute discrete logarithms in a field of p^n elements, NFS works in a similar manner. We select a number ring $\mathbb{Z}_i$, which is simply $\mathbb{Z}$ when factoring and is such that p is inert for discrete logarithms. Then we select two polynomials $f, g \in \mathbb{Z}_i[x]$ having a common factor φ modulo q, where $q = N$ for factoring and $q = p$ for discrete logarithms. This allows to draw a commutative diagram which is the core of NFS:



where α_f and α_g are roots of f and g in their number fields and where $\mathcal{O}_f$ and $\mathcal{O}_g$ are the rings of integers of these same number fields.

The algorithm starts with a stage in which small polynomials $\phi(x)$ are enumerated and put in the top of the diagram. What a small polynomial is changes from variant to variant but the degree and the coefficients are small, the simplest example being $\phi(x) = a - bx$ with integers a, b smaller in absolute value than some parameter. If $\phi(\alpha_f)$ and $\phi(\alpha_g)$ are B-smooth for a parameter B (factor into ideals of norm less than B) then we obtain a multiplicative relation in $\mathbb{Z}_i[x]/\langle q, \varphi\rangle$. At this step the two variants of NFS split: either one transforms multiplicative relations into linear equations and computes a right kernel to obtain a large number of discrete logarithms or one writes a matrix of valuations and computes a left kernel to obtain a non-trivial solution to the equation $x^2 \equiv 1 \bmod N$. In both cases one finishes with a step of negligible cost.

The classical variant of NFS has complexity $L_Q[64]^{1+o(1)}$ where $Q = N$ or p^n and

$$L_Q[c] = \exp\left((c/9)^{\frac{1}{3}} (\log Q)^{\frac{1}{3}} (\log \log Q)^{\frac{2}{3}}\right).$$

Each of the variants of NFS requires its own complexity analysis but it is always of the form $L_Q[c]^{1+o(1)}$ for some constant. Joux and Pierrot [JP13] invented a method of polynomial selection which obtains $c = 32$ for some finite fields where the characteristic p has a special form. Barbulescu, Gaudry, Guillevic and Morain [BGGM15] proposed new methods of polynomial selection which achieve $c = 48$ in some cases intractable with the previous method. Later Barbulescu, Gaudry and Kleinjung [BGK15] proposed to replace $\mathbb{Z}$ by a larger number ring $\mathbb{Z}_i$ and also obtained $c = 32$ for some finite fields, in particular proving that a popular pairings curve estimated to 128 bits can be the target of this variant. Finally, Kim and Barbulescu [KB16] showed how to use the new methods of polynomial selection together with the new choices of $\mathbb{Z}_i$ and obtained $c = 32$ for a very large range of finite fields.

o(1)-less estimation. What is the impact of these new constants in the complexity on the real-life security ? To get a first idea one can start by dropping the $o(1)$ term, so that the cost of each variant of NFS is $2^\kappa L_Q[c]$ where κ and c are two constants. We use the same convention as in [Len01a, Section 2.4.6] and count a clock cycle as one operation. Thanks to real-life record computations we have a relatively good estimation of κ as summarized in Table 1 and we conclude on the security estimations in Figure 1. For those fields where the fastest variant applies it seems that we have to use 5008 bit fields for 128 bits of security and 12871 for 192 bits of security.

variant	classical NFS	classical MNFS	composite n NFS	composite n MNFS	SNFS
c	64	61.93	48	45.00	32
κ	-8[KDL+16]	-8 [KDL+16]	-7[BGGM15]	-7 [BGGM15]	-7[AFK+07]

Table 1: Value of κ to match the formula cost(NFS)=$2^\kappa L_Q[c]$

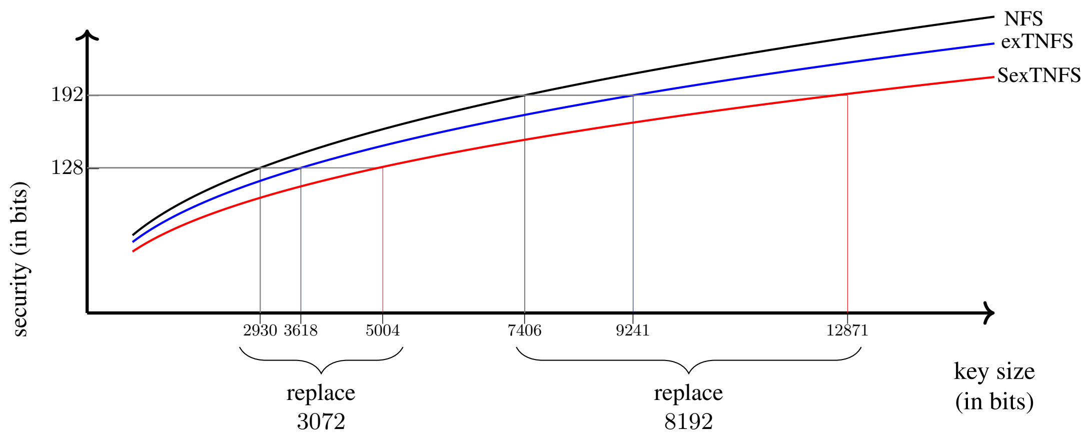


Fig. 1: Modification of key sizes according to the o(1)-less formula.

The goal of this article is to go beyond the $o(1)$-less estimation and to study in each case what is the best variant of NFS which applies, concluding on new key sizes. This type of estimations seem to be rare but we can note the works of Lenstra [Len01b] and of Bos et al. [BKK$^+$09] who evaluate the security of RSA, DSA and DH. In a recent article Menezes et al. [MSS16] made a precise estimation for pairings. In our analysis we consider elements which are not included by Menezes et al. making our work complementary.

2 Families of pairing-friendly cruves

Depending on the required embedding degree, some families of curves have been built [FST10]. We recall here the most popular ones.

2.1 BN curves

A BN curve [BN05] is an elliptic curve E defined over a finite field $\mathbb{F}_p$, $p \geq 5$, such that its order r and p are prime numbers parametrized by

$$p = 36u^4 + 36u^3 + 24u^2 + 6u + 1,$$
$$r = 36u^4 + 36u^3 + 18u^2 + 6u + 1,$$

for some well chosen u in $\mathbb{Z}$. It has an equation of the form $y^2 = x^3 + b$, where $b \in \mathbb{F}_p^*$. BN curves have an embedding degree equal to 12. They were widely used for the 128-bit security level till the recent results on the discrete logarithm problem in $\mathbb{F}_{p^{12}}^*$. Indeed, a 256-bits prime p leads to a 256-bits curve and to pairings taking values in $\mathbb{F}_{p^{12}}^*$, which is a 3072-bits multiplicative group. Both groups involved are then supposed to match the 128-bit security level according to the NIST recommendations [Nat12] (which are however now invalidate by [KB16]). By the way, BN curves have been the object of numerous recent publications ([DSD,AKL$^+$11,CDF$^+$11,GJNB11,NNS10,GAL$^+$13,UW14]).

Finally, BN curves always have order 6 twists. If ξ is an element which is neither a square nor a cube in $\mathbb{F}_{p^2}$, the twisted curve E' of E is defined over $\mathbb{F}_{p^2}$ by the equation $y^2 = x^3 + b'$ with $b' = b/\xi$ or $b' = b\xi$. In order to simplify the computations, the element ξ should also be used to represent $\mathbb{F}_{p^{12}}$ as a degree 6 extension of $\mathbb{F}_{p^2}$ ($\mathbb{F}_{p^{12}} = \mathbb{F}_{p^2}[\gamma]$ with $\gamma^6 = \xi$) [DSD], [LN97].

2.2 BLS curves

BLS curves were introduced in [BLS03]. They are also defined over a parametrized prime field $\mathbb{F}_p$ by an equation of the form $y^2 = x^3 + b$ and have a twist of order 6 defined in the same way than BN curves. Contrary to BN curves they do not have prime order but their order is divisible by a large parametrized prime r and the pairing will be defined on the r-torsions points. They are available for different embedding degrees but we are only interested here by the BLS12 and BLS24 families having embedding degrees 12 and 24 with respect to r. Till now, they were used for the 192-bis security level [AFCK+13]. The parametrizations are given by

BLS12	BLS24
$p = (u-1)^2(u^4 - u^2 + 1)/3 + u$	$p = (u-1)^2(u^8 - u^4 + 1)/3 + u$
$r = u^4 - u^2 + 1$	$r = u^8 - u^4 + 1$

2.3 KSS curves

KSS curves are also available for different embedding degrees [KSS08]. If the required embedding degree is 18, this is very similar to BLS curves (same defining equation, degree 6 twist, parametrized primes p and $r|\#E(\mathbb{F}_p)$). In this case, the parametrization is given by

$$p = (u^8 + 5u^7 + 7u^6 + 37u^5 + 188u^4 + 259u^3 + 343u^2 + 1763u + 2401)/21$$
$$r = (u^6 + 37u^3 + 343)/343$$

If the required embedding degree is 16, the KSS16 curves are defined over a parametrized prime field $\mathbb{F}_p$ by an equation of the form $y^2 = x^3 + ax$ and have a twist of order only 4. Again they do not have a prime order but it is divisible by a parametrized prime r and the pairing will be defined on the r-torsions points. In this case, the parametrization is

$$p = (u^{10} + 2u^9 + 5u^8 + 48u^6 + 152u^5 + 240u^4 + 625u^2 + 2398u + 3125)/980$$
$$r = (u^8 + 48u^4 + 625)/61250$$

Whatever the family, a curve is always obtained by finding a parameter u such that both p and r are prime numbers. The curve and its twist are generated by finding suitable coefficients which can usually be chosen small. More details on the generation process are given in Section 6.

2.4 Optimal Ate pairing

Their are several available pairings (Weil, Tate, Ate, R-Ate, ...) but the most efficient pairing is always the so-called optimal Ate pairing [Ver09]. Let us recall this pairing in the context of ordinary elliptic curves defined over prime fields and more precisely in the case of the considered families.
Let E be an elliptic curve defined over the prime field $\mathbb{F}_p$. Let r be a prime divisor of $\#E(\mathbb{F}_p)$ and k the embedding degree relatively to r. Let $\tilde{E}$ be a degree d twist of E defined over $\mathbb{F}_{p^e}$ where $e = k/d$ [HSV06]. The optimal Ate pairing is defined over $\mathbb{G}_1 \times \mathbb{G}_2$ and takes its values in $\mathbb{G}_3$ where

- $\mathbb{G}_1$ is the set of rational points on E of order r.
- $\mathbb{G}_2$ is the image of $\tilde{E}(\mathbb{F}_{p^e})[r]$ in $E(\mathbb{F}_{p^k})$ by the twisting isomorphism.
- $\mathbb{G}_3$ is the order r subgroup of $\mathbb{F}_{p^k}^*$

For the considered parametrized curves, the optimal Ate pairing of P and Q is mainly made of 2 parts. The first one (usually called the Miller loop) is the computation of $f_{u,Q}(P)$ where u is the family parameter and the second one is an exponentiation to the power $\frac{p^k-1}{r}$. Assuming $\ell_{A,B}$ denotes the line through points A and B, the precise pairing are given in Table 2 [HSV06,Ver09].

Curve	Miller loop of P and Q	final exponent
BN	$f_{6u+2,Q}(P) \cdot \ell_{[6u+2]Q,[p]Q}(P) \cdot \ell_{[6u+2+p]Q,[p^2]Q}(P)$	$(p^{12}-1)/r$
BLS12	$f_{u,Q}(P)$	$(p^{12}-1)/r$
KSS16	$\left(f_{u,Q}(P) \cdot \ell_{[u]Q,[p]Q}(P)\right)^{p^3} \cdot \ell_{Q,Q}(P)$	$(p^{16}-1)/r$
KSS18	$f_{u,Q}(P) \cdot f_{3,Q}(P)^p \cdot \ell_{[u]Q,[3p]Q}(P)$	$(p^{18}-1)/r$

Table 2: Optimal Ate pairings

3 The spectrum of possibilities for an attack on the field side

An attacker who uses an algorithm of Index calculus type can make a series of choices : decide which algorithm and variant to use, improve implementation, select polynomials, and optimize the main parameters. In this section we explain what are the reasonable choices for an attacker and give arguments to eliminate other choices.

3.1 Choice of algorithm

Let us make a list of the algorithms which can be implemented on a classical computer.

We discard the FFS algorithm [Adl94,AH99,JL02,JL06] and its pinpointing variant [Jou13] by estimating the size of the factor base. Indeed, when the target is $\mathbb{F}_{p^n}$, the factor base of FFS is formed of all the monic polynomials $\mathbb{F}_p[x]$ of degree less than a parameter b. This has been confirmed by implementations of FFS [JL06,HSW+10,HSST12] and pinpointing [Jou13,SS16a]. Hence the factor base has at least p elements and then the linear algebra step has a cost of at least p^2 operations, which is more than the security on the curve side evaluated to $p^{\frac{1}{2}}$ operations.

We also discard the MNFS variants, i.e. the variants of NFS in which more than two sides are used. Indeed, the asymptotic complexity is close to that of NFS ([KB16, Table 2] so the "o(1)-less" extrapolation leads us to results which are similar to those of the classical case (see Figure 1). Detrey [Det14] and Lenstra and al. [KBL14] made proof-of-concept implementations of FFS and NFS for factoring, which are similar to NFS for discrete logarithms. Their results seem to show that the crossing point between classical and MNFS variants of NFS is around 1000 bits, but the gain is small, say less than 2 bits of security, so that we can ignore it in this article.

The three variants of NFS, classical [Gor93,Sch93,JL03], TNFS [Sch00,BGK15] and JLSV [JLSV06], can be seen as particular cases of exTNFS [KB16], which remains the only algorithm to consider.

When p can be written as $P(u)/v$, for some polynomial $P \in \mathbb{Z}[x]$ and some integers u and v (as it is the case for pairing applications), the polynomial selection is done differently and one of f and g has small coefficients. To emphasize this difference we give a different name to the algorithm by adding the letter S: the "special" variant of NFS is called SNFS, the special variant of exTNFS is called SexTNFS, the corresponding variant of TNFS is STNFS and the special variant of JLSV will be called S-JLSV or simply Joux-Pierrot.

In order to fix notations we recall the **(S)exTNFS algorithm** [KB16]:

1. Polynomial selection. Given a parameter η, chosen among the divisors of n, one selects a polynomial $h \in \mathbb{Z}[x]$ of degree η which is irreducible modulo p. Then one selects two polynomials f and g in $\mathbb{Z}[t,x]$ so that $f \bmod \langle h(t), p\rangle$ and $g \bmod \langle h(t), p\rangle$, seen as elements of $\mathbb{F}_{p^\eta}[x]$, have a common factor $k(x)$ which is irreducible of degree $\kappa := n/\eta$. In the particular case $\gcd(\eta,\kappa) = 1$ we can take $f, g \in \mathbb{Z}[x]$ which share an irreducible factor of degree κ, whereas in the case $\gcd(\eta,\kappa) \neq 1$ we have to guarantee that f and g are not defined over a proper subfield of the number field of h.
2. Sieve. Given two parameters A and B, one collects all (up to sign) the degree 1 polynomials in $\mathbb{F}_{p^n}[x]$ or equivalently tuples in the set $\{(a_0,\ldots,a_{\eta-1},b_0,\ldots,b_{\eta-1} \in [-A,A]^{2\eta} \mid a_0 \geq 0\}$, called sieving domain, so that N_f and N_g are B-smooth (all prime factors are less than B), where

$$N_f = \mathrm{Res}_t\left(\mathrm{Res}_x\left(\sum_{i=0}^{\eta-1} a_i t^i - x\sum_{i=0}^{\eta-1} b_i t^i, f(t,x)\right), h(t)\right)$$

is the norm on the f side, and similarly for g instead of f. In order to emphasize the analogy with the simpler variants of NFS, we put $E = A^\eta$ which is a good approximation of the square root of the cardinality of the sieving domain.

3. Filtering. Unknowns which occur in a single relation are called singletons and are deleted together with the corresponding equation. Additionally, using elementary transformations of the matrix one can create new singletons. This leads to a smaller matrix and hence a faster resolution of the linear system.
4. Linear algebra step. One computes the right kernel of the sparse matrix obtained after the filtering using the Wiedemann or Lanczos algorithm or their block variants. The coordinates of the kernel vector are called virtual logarithms.
5. Individual logarithms. Given a generator g of $\mathbb{F}_{p^n}$ and an element h, compute the discrete logarithm $\log_g h$ using the virtual logarithms.

3.2 Improving the implementation

Although the complexity of NFS for DLP in $\mathbb{F}_p$ hasn't changed for almost 30 years, its real-life speed was improved continuously. In the jargon of the NFS community an improvement which changes the $o(1)$ term in the complexity is called a practical improvement and is presented as an implementation detail.

Filtering If an ideal occurs in a single relation then we can erase this ideal and its relation from the matrix. Thanks to the exceeding number of relations compared to the cardinality of the factor base, one can erase rows and do linear operations on the rows in order to create new singletons [Cav02, Ch 3]. Table 3 summarizes how does the filtering behave in practice. It is hard to compare the different rows of the table because the authors of different records made different choices, some of which collected much more relations than needed (oversieved) and hence helped the filtering step reduce considerably the matrix.

record	rows before filtering	rows after filtering	size reduction	$\log_2 B$
SNFS-1039 (factor)	13.8G	82.8M	167	38
NFS-768 (factor)	47.7G	192.8M	247	40
FFS-809	67.4M	3.6M	19	28
SNFS-1024 (DLP)	249M	28M	9	31
NFS-768 (DLP)	9.0G	23.5M	382	36

Table 3: Behavior of filtering in practice

We made an asymptotic estimation of the number of ideals which might be used to reduce the matrix and we obtained the following statement.

Conjecture 1. In the filtering step of NFS one reduces the matrix by a factor $(\log B)^{1+o(1)}$, where B is the smoothness bound.

Justification: Let $\mathfrak{q}$ be an ideal in the factor base of NFS lying above a prime q and let N denote the size of the norms product and B the smoothness bound. We shall argue that the following statements are true:

1. If $q < B/(\log B)^{1+\epsilon}$ with $\epsilon > 0$ then $\mathfrak{q}$ occurs in a number of relations which tends to infinity as B and N go to infinity.
2. If $q > B/(\log B)^{1-\epsilon}$ with $\epsilon > 0$ then $\mathfrak{q}$ will occur in a number of relations which tends to 0 as B and N go to infinity.

The sieving domain has B^2 elements (parameter tuning in NFS implies $E = B$ where E is the square root of the number of sieved pairs [BLP93]) and a proportion of $1/q$ are divisible by $\mathfrak{q}$. They produce relations if the cofactor of size N/q is B-smooth, for which we have no proven formula, but which is approximated by the proportion of integers in the interval $[1, N/q]$ which are B-smooth. Due to the theorem of Canfield, Erdös and Pomerance [CEP83] this proportion is $\rho\left(\frac{\log(N/q)}{\log B}\right)$ where ρ is Dickman's function, i.e. the function such that $\rho(v) = 1$ for $v \leq 1$ and $\rho'(v) = -\rho(v-1)/v$ for $v > 1$.

$$\text{number of relations where } \mathfrak{q} \text{ occurs} \approx B^2/q \cdot \rho\left(\frac{\log(N/q)}{\log B}\right).$$

Recall that in NFS we set B so that $\rho\left(\frac{\log N}{\log B}\right)^{-1} = B$. (once again see [BLP93]). We put $v = \frac{\log N}{\log B}$, so that we have $\log B = v \log v$, $\log N = v^2 \log v$ and $q > B/v^{1+2\epsilon}$ (resp. $q < B/v^{1-2\epsilon}$). We replace all variables in the the right hand side member by their expressions in terms of v and obtain that its logarithm is equivalent to $v^{1+\epsilon} - v$. It tends to ∞ if $\epsilon > 0$ so the ideals of norm $q < B/(\log B)$ occur in a very large number of relations and are unlikely to create singletons, so they are not erased during filtering. The right member tends to $-\infty$ if $\epsilon < 0$ so the ideals of norm $q > B/\log(B)$ occur in almost no relations, and are very likely to be used during filtering.

Hence the filtering erases most of the ideals of norm larger than $B/(\log B)^{1+o(1)}$ and keeps all but a negligible fraction of the others, so that the matrix size is reduced by a factor $(\log B)^{1+o(1)}$. ☐

It seems then plausible that the filtering gain is a constant times $\log(B)$, and by comparing it with Table 3 we model the gain by $\log_2 B$.

Exploiting automorphisms. Record computations with FFS [HSW⁺10,HSST12] and NFS [BGGM15] showed that if the target field is of the form $p^{\kappa\eta}$ for two integers η and κ so that κ is small, then one can gain a factor κ in the sieve and a factor κ^2 in the linear algebra.

Kim and Barbulescu [KB16] explained that one has a similar gain in SexTNFS, where κ is to be replaced by $\mathcal{A}$, the number of automorphisms of h which fix g times the number of automorphisms of g. If $\kappa = 1$ and h has η automorphisms then the exact number of automorphisms is $\mathcal{A} = \eta$, e.g. $\mathcal{A} = \ell - 1$ if $h = \Phi_\ell$ for some prime ℓ. If $\kappa = 2$ one doubles the number of automorphisms thanks to the automorphisms of g. For example if $h = \Phi_7$ and $g = x^2 + \alpha x + \beta + t^4 + t^2 + t - u$ for some integers α, β then $\mathcal{A} = 6$ because any automorphism in the set $\{\tau^i\sigma^j, 0 \leq i \leq 1, 0 \leq j \leq 2\}$ can be used (here $\sigma : t \mapsto t^2$ and $\tau : x \mapsto -\alpha - x$). Finally, if $\kappa = 3$ and $\eta = 4$ an attacker might use $h = \phi_8$ and find polynomials g which have 3 automorphisms, so for a worst case analysis we count $\mathcal{A} = 12$.

3.3 Selection of polynomials

The polynomial selection consists of selecting h, f and g.

Choice of h. The polynomial $h \in \mathbb{Z}[x]$ has two constraints, its degree is η and it is irreducible modulo p. Among the possible choices we select those having small norms for N_f and N_g, which generally corresponds to the case when h has small coefficients. In all examples we could select h with coefficients in $\{0, -1, 1\}$ and experiments confirmed that the best choice is never much better than $h = t^\eta - t - 1$.

In Section 3.2 we saw that in order to use the Galois automorphisms the attacker has to find a polynomial h with non-trivial automorphisms. We ran an exhaustive search of integer polynomials with coefficients less than 10 and obtained a very small set of such polynomials which have no automorphism of order different from 2, that we summarize in Table 4.

degree	6	12	16	18
polynomials	Φ_7, Φ_9, Φ_{14}, Φ_{18}	Φ_{13}, Φ_{26}	Φ_{17}, Φ_{34}	Φ_{19}, Φ_{38}

Table 4: List of all monic polynomials $h \in \mathbb{Z}[x]$ of degree between 6 and 20 such that $\|h\| \leq 10$ and $Aut(h)$ has elements of order larger than 2.

Construction of f and g. One produces a large number of pairs of polynomials using one of the following methods: base-m [BLP93], base-m-SNFS [LLJMP90], Joux-Pierrot [JP13], Conjugation [BGGM15], JLSV1 [JLSV06, Section 2.3], GJL [BGGM15,Mat06], algorithms A,B,C or D of Sarkar and Singh [SS16c,SS16d,SS16b].

In this article we focus on families of pairings where p is parametrized, then one choice of polynomials is by far the most natural. Let $P(x) \in \mathbb{Z}[x]$ and the integers u, v be such that $p = P(u)/v$. Then one can take $f = P(x^\kappa + S(t,x))$ and $g = x^\kappa + S(t,x) - u$ for some $S \in \mathbb{Z}[t,x]$ of degree in x less than κ so that g is irreducible in $(\mathbb{F}_p[t]/h)[x]$. In most cases this is the only choice but for instance in the case of KSS 18 one can also take $f = P(x-2)$ and $g = x - 2 - u$, with a non negligible effect on the complexity estimation.

How can we be sure that the attacker cannot find choices of f that we could not predict ? See [FGHT16] for a discussion about the consequences of this question on discrete logarithms in $\mathbb{F}_p$. The attacker cannot use the fastest versions of NFS (SNFS, STNFS, SexTNFS, Joux-Pierrot) unless he finds three polynomials, $T(x,y) \in \mathbb{Z}[x,y]$ and $U, V \in \mathbb{Z}(x)$ whose coefficients are bounded by an absolute constant, so that $p = T(U(u), V(u))$ for some integer u, in which case he sets

$$f = T(x) \text{ and } g = V(u)x - U(u).$$

In the case of SexTNFS, the coefficients of f occur at large powers in the norms and hence we can restrict the search to very small constants. We ran the exhaustive search and obtained that the only alternative choices are $f = P(x-1)$ for KSS 16, $f = P(x-2)$ for KSS 18 and $f = 4x^4 - 4x^3 + 12x^2 - 10x + 7$ and $g = x - (3u+1)$ for BN. In the rest of the security evaluation we considered the alternative choices together with the natural ones.

Optimization. Murphy [Mur98] introduced a map $\alpha : \mathbb{Q}[x] \to \mathbb{R}$ which allows to decide which are the best polynomials for NFS. Barbulescu and Lachand [BL16] proved, when f is quadratic of fundamental negative discriminant, that for a random pair of relatively prime integers the norm $N = \mathrm{Res}_x(a - bx, f)$ has the same probability to be B-smooth (for a parameter B) as a random integer less than $e^{\alpha(f)}N$. Because of the uncertainty on α we cannot predict the exact cost of a DLP computation with NFS. In the previous paragraph we saw that in the case of parametrized pairings we only have one or two choices of f and g. For each choice we verify directly that $\alpha(f) \approx 0$ whereas for linear polynomials the value of α is constant equal to $0, 56...$ which is also the average value of α on all polynomials [BL16].

4 Optimization of parameters

Given a field $\mathbb{F}_{p^n}$ where the characteristic is parametrized by a polynomial $P(u)/v$ of degree d, we decided to use SexTNFS with $f = P(x^\kappa + S(t,x))$ and $g = x^\kappa + S(t,x) - u$ for some polynomial S of degree in x less than n. We also decided to use, if possible, h from Table 4 and otherwise $h = t^{n/\kappa} - t - 1$ because it is the simplest one and then the one providing the smallest norms. This choice is the best possible one for the attacker. At this point we need to decide which value of κ to use and to optimize parameters A and B.

Choice of κ According to [KB16, Section 4.1] the parameter κ is chosen to minimize the norms product $N_f N_g \approx E^{(d+1)\kappa} Q^{\frac{1}{d\kappa}}$, where E is the square root of the cardinality of the sieve space and Q is p^n. This corresponds to

$$\kappa \approx \sqrt{\frac{\log_2 Q}{d(d+1)\log_2 E}}.$$

It was useful for us to guess the optimal value of κ, which received the most attention, even though we had to test all possible values. Our method was to approximate $\log_2 Q$ from Figure 1 and to take $E^2 = 2^s$ where s is the security level, which leads to Table 5. We verified that in every case the best value is in this table.

security level	$d=4$	$d=6$	$d=8$	$d=10$
128	2 or 3	1 or 2	1 or 2	1
192	2 or 3	1 or 2	1 or 2	1 or 2

Table 5: Rule of thumb values for κ.

Optimization of the bounds A and B. As before B denotes the smoothness bound and A the bound on the coefficients of the sieved polynomials. A pair of values is valid if the sieve produces enough relations, so we need to estimate the number of relations. The sieving space is formed of the pairs $a(t), b(t)$ in $\mathbb{Z}(t)/h$ so that $\deg a, \deg b \leq \eta - 1$. If $\mu(t)$ is a root of unity of the number field of h then the pairs $(\mu a, \mu b)$ and (a, b) give the same multiplicative relation. In Section 3.1 we restricted a_0 to positive values to account for the unit -1, here the sieving space shrinks further by the number of roots of unity divided by two.

$$\text{sieving space} = (2A+1)^{2\eta}/(2w),$$

where w is the index of $\{1, -1\}$ in the group of roots of unity. By a Monte Carlo integration we estimate the bit size of the norms: we considered random tuples $(a_0, \ldots, a_{\eta-1}, b_0, \ldots, b_{\eta-1})$ each of the components being uniformly chosen in the interval $[-A, A]$. We call bit size of the norms the arithmetic mean of the bits sizes of the norms for each tuple in a sample of 25600 tuples (see Appendix A for more details). We emphasize that we average $\log_2(N_f)$ and $\log_2(N_g)$ and not N_f and N_g because this value is used to compute the smoothness probabilities $p_f = \rho\left(\frac{\log_2 N_f}{\log_2 B}\right)$ and $p_g = \rho\left(\frac{\log_2 N_g}{\log_2 B}\right)$. This gives us the total number of relations which is

$$\text{relations} = (\text{sieving space}) \cdot p_f \cdot p_g.$$

The factor base is formed of the prime ideals of norm less than B in the number fields of f and g, so the cardinality of the factor base is asymptotically equal to $2B/\log(B)$. Due to the Galois speed-up the factor base can be reduced by a factor equal to the number of automorphisms of h times the number of automorphisms of $\mathbb{F}_{p^n}/\mathbb{F}_{p^\eta}$ which fix g seen as an element of $\mathbb{F}_{p^\eta}[x]$, factor that we denote $\mathcal{A}$. In some record computations the number of relations is less than the cardinality of the factor base, e.g. 68% in [AFK+07], but for simplicity and without changing the complexity results by more than one bit, we consider that the attacker must collect at least as many relations as elements in the factor base. Hence the validity condition is

$$\frac{(2A+1)^{2\eta}}{2w} \cdot p_f \cdot p_g \geq \frac{2B}{\mathcal{A}\log(B)}. \tag{1}$$

The sieve ends when enough relations have been found so the number of enumerated pairs is $2B/\mathcal{A}\log(B)p_f p_g$. The ratio beetween the real cost of the sieve and the number of tuples enumerated in the sieve is hard to evaluate so we call it c_{sieve}. According to Table 6, c_{sieve} is almost constant in various computations realized with various variants of NFS. We stay on the safe side and model c_{sieve} to be a constant equal to 1 even though we believe its value should increase slowly with the input size and the first implementations on SexTNFS will be much worst than the implementations of NFS which obtained these costs. Finally

$$\text{sieve cost} = \frac{2B}{\mathcal{A}\log(B) \cdot p_f \cdot p_g}$$

The size of the matrix sent to filtering is $2B/\mathcal{A}\log(B)$. As explained in Section 3.2 it is reduced by a factor $\log_2 B$. The number of non-zero entries per row in the reduced matrix varies between 100 and 200 in all records that we consider and we will approximate it by 128. Let then $c_{\text{lin.alg}}$ be such that the cost of the linear algebra is $c_{\text{lin.alg}} 2^7 B^2/(\mathcal{A}\log(B)\log_2(B))^2$, as it is expected to be using Wiedemann's algorithm. The factor $c_{\text{lin.alg}}$ accounts for the cost of a multiplication in $\mathbb{F}_r$, where r is the order of the pairings group. Since $\log_2 r$ varies by at most a factor 2 between various types of pairings and various security levels between 128 and 256, we expect $c_{\text{lin.alg}}$ to be a constant. The records we summarize in Table 6 confirm that $c_{\text{lin.alg}}$ is a constant close to 1.

We conclude this section with a model of the cost:

$$\text{cost} = \frac{2B}{\mathcal{A}\log B}\,\rho\left(\frac{\log_2(N_f)}{\log_2 B}\right)^{-1} \rho\left(\frac{\log_2(N_g)}{\log_2 B}\right)^{-1} + 2^7 \frac{B^2}{\mathcal{A}^2(\log B)^2(\log_2 B)^2}, \tag{2}$$

where $\mathcal{A}$ can be upper bounded by $\eta\kappa/\gcd(\eta, \kappa)$.

For each pairing curve and choice of polynomials one has to solve an optimization problem: find the values of $\log_2 A$ and $\log_2 B$ which minimize the cost in equation 2 under the condition in Equation 1.

record	$\log_2 E$	$\log_2$(cost of sieve)	$\log_2 B$	$\log_2$(cost of lin.alg)	$\log_2(c_{\text{sieve}})$	$\log_2(c_{\text{lin.alg}})$
SNFS-1039 (factor)	31.0	63.0	38	63.0	1	1
NFS-768 (factor)	33.0	66.5	40	64.5	0.5	-2
FFS-809	27.0	57.5	28	55.0	3.5	2
SNFS-1024 (DLP)	31.5	64.5	31	63.5	1.5	2
NFS-768 (DLP)	35.0	68.0	36	66.0	-2	-4

Table 6: A list of records and their parameters.

5 Estimating SexTNFS complexity

In this section, we use the previous result to estimate the security level provided by a given finite field $\mathbb{F}_{p^n}$.

5.1 Recapitulative process for computing SexTNFS cost

Let us first summarize the way to estimate the complexity of the SexTNFS algorithm. It is made of 4 steps.

- **Step 1: Parameter selection.** The first choice to be made is the one of the κ value according to Table 7 and/or trying few values and only keep the one given the best results in the following steps. Then one has to choose the polynomial h such that $\mathcal{A}$ is as large as possible and h is as simple as possible (small and few coefficients) and the polynomials f and g to define the commutative diagram given in the introduction. The details on the ways to choose these polynomials are given in Section 3.3. In this step, we also determine the values w and $\mathcal{A}$
- **Step 2: Choice of the bounds A and B.** These bounds will define the number of enumerated relations and the size of the factor basis so they have a direct impact on the complexity. As already explained they must be chosen to minimize the cost in Equation 2 under the condition in Equation 1 (these are steps 3 and 4 below). This optimization problem will be solve by brute force because we do not need a very high accuracy. We first enumerate only integer values of $\log_2 A \in [1, \frac{150}{\eta}]$ and $\log_2 B \in [1, 150]$ because the cost is lower bounded by $(A^{2\eta} + B^2)/1000$ which is more than 2^{192} for larger values of parameters. We call $\log_2 A_0$ and $\log_2 B_0$ the optimum of this integer search. In a second time we test all values of $\log_2 A$ in the set $\{\log_2 A_0 + i/100 \mid i \text{ integer in } [-100, 100]\}$ and all values of $\log_2 B$ in the set $\{\log_2 B_0 + j/5 \mid j \text{ integer in } [-25, 25]\}$. For small values of A, one can even enumerate A one by one.
- **Step 3: Sieving.** For each choice of A, we use a Monte Carlo integration to estimate the average bitsize of the norms N_f and N_g. Hopefully, these norms are essentially proportional to $A^{\deg f \deg h}$ so we do not need to recompute it each time for a first approximation. They allow to estimate the number of enumerated pairs necessary to get enough relations.
- **Step 4: Final cost.** We can deduce from Step 3 the sieving cost (which is the number of enumerated pairs, assuming $c_{\text{sieve}} = 1$) and from B the cost of linear algebra which is $2^7 B^2/(\mathcal{A}\log(B)\log_2(B))^2$. The overall complexity is the sum of these 2 costs.

5.2 Example: a BN curve where the finite field has 3072 bits

One of the most popular BN curves is the one associated to $u = -2^{62} - 2^{55} - 1$ which was evaluated to 128 bits of security before the recent development on NFS. Let us follow Section 5.1 to estimate its real security level.

- **Step 1: Parameter selection.** We decide to use the SexTNFS algorithm with $\kappa = 2$ and $\eta = 6$ because it gives the best result from the viewpoint of the attacker. The intermediate field will be defined by $h = t^6 - t^3 - t - 1$ which is irreducible modulo p. Indeed the cyclotomic polynomials $\Phi_7, \Phi_9, \Phi_{14}$ and Φ_{18} are not irreducible in this case and h is the "smallest" irreducible polynomial (in the sense that he has only 4 non-zero coefficients which moreover equal ± 1). We tried several polynomials and find that $x^2 + t - u$ is irreducible in $\mathbb{F}_{p^6} = \mathbb{F}_p[t]/h(t)$ so that $\mathbb{F}_{p^{12}} = \mathbb{F}_{p^6}[x]/(x^2 + t - u)$. Hence we can take $f = P(x^2 + t)$ (where P is the polynomial parametrizing p given in Section 2.1) and $g = x^2 + t - u$. In this case, we have no non-trivial roots if unity ($w = 1$) and $\mathcal{A} = 2$ because.

– **Step 2: Choice of the bounds A and B.** As explained in Section 5.1, we applied Steps 3 and 4 for many values of A and B to find that $\log_2 A = 7.36$ and $\log_2(B) = 57$ are minimizing the cost given by the Equation 2.
– **Step 3: Sieving.** The total number of tuples in the sieving space is $(2A+1)^{2\eta}/(2w)$, where $w = 1$ is the number of automorphisms of h, so the size of the sieving space is $2^{99.45}$. By Monte Carlo integration we estimates the norms on the two sides of the commutative diagram and then one can approximate the smoothness probability using Dickman's function

$$\log_2(N_f) \approx 414.7 \Rightarrow \rho\left(\frac{\log_2(N_f)}{\log_2(B)}\right) \approx 2^{-21.41} \text{ and } \log_2(N_g) \approx 460.8 \Rightarrow \rho\left(\frac{\log_2(N_g)}{\log_2(B)}\right) \approx 2^{-25.30}$$

Hence the number of relations is approximatively $2^{99.45-21.41-25.30} \approx 2^{52.74}$.
On the other hand, the cardinality of the factor base is approximatively $2B/\mathcal{A}\log(B) \approx 2^{51.70}$, which is less than the number of relations, so we have enough relations (Equation 1 is satisfied).
– **Step 4: Final cost.** The number of relations we need to collect is $2^{51.70}$ and each relation is obtained after testing on average $2^{21.41+25.30} = 2^{46.71}$ pairs (a, b). Hence the cost of the sieve is $c_{\text{sieve}} 2^{51.70+46.71} \approx 2^{98.41}$ assuming $c_{\text{sieve}} \approx 1$. On the other hand, Filtering allows to reduce the matrix size by a factor around $\log_2 B = 57$, its new size being $N = 2^{51.70}/57 \approx 2^{46.87}$. The cost of the algorithms of sparse linear algebra is given by $2^5 N^2 = 2^{98.73}$ times the cost of an addition modulo p, which counts here for an elementary operation. Finally, we get the overall cost by adding the cost of the relation collection and the one of the linear algebra :$2^{98.65} + 2^{98.73} = 2^{99.69}$ which means that the BN curve used in most of the existing implementations ensures no more than the 100-bits security level.

5.3 General results and recommendations

The goal of this section is to determine the required size of the finite field involved in the pairings given in Section 2 to ensure the 128 and 192-bits security levels. For this, we follow the strategy given in Section 5.1 for each family of curves making at each step the most favorable choice (for the attacker). For example we assumed that the number of automorphisms $\mathcal{A}$ is maximal and the number of roots of unity w divided by ± 1 is 1. If the parameter u (and therefore p) is selected such that the attacker cannot use the best polynomials listed in Table 7 then we observed a considerable increase in security. However, for the purpose of general recommendations, we consider that the attacker can use the best polynomials. The results are given in Tables 8 and 9 which then contain our recommendations for the size of p^k where k is the embedding degree.

family	η	h	g	w	$\mathcal{A}$
BN, BLS 12	6	Φ_7	$x^2 - u + t$	7	6
KSS 16	16	Φ_{17}	$x - u$	17	16
KSS 18	18	Φ_{19}	$x - u$	19	18

Table 7: Best choices of h and g at 128 bits of security

family	$\log_2(p^k)$	κ	$\log_2 A$	$\log_2 B$
BN	5534	2	10.16	74.00
BLS 12	5530	2	10.10	73.65
KSS16	3840	1	3.06	71.75
KSS18	4096	1	2.57	71.50

Table 8: Size of finite fields associated to pairing-friendly curves which have a cost of 2^{128} operations.

family	$\log_2 u$	$\log_2(p^k)$	κ	$\log_2 A$	$\log_2 B$
KSS18	85	12200	1	9.85	105.0
BLS 24	56	13300	1	3.06	109.4

Table 9: Recommended parameters for pairings of 192 bits of security

6 New parameters for the 128-bits security level

The goal of this section is to propose new parameters for the 128-bits security level for the main families of curves given in Section 2 (BN, BLS12, KSS16 and KSS18). This is done in 2 steps. The first one consists in finding the size of the extension field ensuring this security level in the general case which means that we assume that the all the improvements of the NFS-like algorithms can be used. This is done in Section 5.3 and the results are given in Table 8. We must also take care that the r-torsion subgroup of the elliptic curve involved in the pairing computation ensures the 128-bits security level. For example, this is the limiting factor in the KSS cases. Then, for each family, we know the size of the curve parameter u that should be used to ensure the 128-bits security level (Table 10) in the general case.

Curve	BN	BLS12	KSS16	KSS18
$\log_2(u)$	114	77	34	44

Table 10: Bitsize of the parameter u ensuring the 128-bits security level

The second step is to generate the best possible parameter u satisfying this condition. Note that since the recommendations correspond to the weakest curve we could have considered slightly smaller values of u for which the specialized security analysis would conclude that they ensure the 128-bits security level. We decided not to do it to keep a small security margin in order to eventually balance the few simplifying assumptions we made on the complexity estimation of SexTNFS. Let us start with the generation of a BN curve.

6.1 New BN parameter

The way to build the parameter u is detailed in [DMHR15]: it should be chosen sparse and congruent to 7 or 11 mod 12 so that building $\mathbb{F}_{p^{12}}$ can be done via $Y^6 - (1+\mathbf{i})$ over $\mathbb{F}_{p^2} = \mathbb{F}_p[\mathbf{i}]$. We also impose the condition that the curve obtained is twist-secure [VLFR08] which means that $p+1+t$ should have a 256-bits prime factor (where t is the trace of the Frobenius as usual). We then performed an exhaustive search on u having increasing Hamming weight. There are no result of weight 2. We found some values having Hamming weight 3 but not satisfying the congruence. More precisely, the extension tower should be build using $\sqrt{-5}$ which is much less interesting in terms of $\mathbb{F}_{p^{12}}$ arithmetic. Finally, we found the value $u = 2^{114} + 2^{101} - 2^{14} - 1$ which is satisfying all the required conditions. The curve E defined over $\mathbb{F}_p$ by

$$E : y^2 = x^3 - 4$$

is twist-secure ($p+1+t$ has a 280-bits prime factor) and $u = 7 \bmod 12$ so that $\mathbb{F}_{p^2}$ is defined by X^2+1 and $\mathbb{F}_{p^{12}}$ by $Y^6 - (1+\mathbf{i})$. The twisted curve E' is defined over $\mathbb{F}_{p^2}$ by

$$E' : y^2 = x^3 - 4(1+\mathbf{i})$$

6.2 New BLS12 parameter

Most of the results of [DMHR15] can be used for BLS curves because the extension degree is also 12. Again, we performed an exhaustive search on the parameter u having increasing Hamming weight. We do not find any value of weight 2 but we found two having Hamming weight 3, $-2^{77} + 2^{50} + 2^{33}$ and $-2^{77} - 2^{59} + 2^9$. In both cases $\mathbb{F}_{p^{12}}$ can be build via $Y^6 - (1+\mathbf{i})$ over $\mathbb{F}_{p^2} = \mathbb{F}_p[\mathbf{i}]$ which provides the best possible $\mathbb{F}_{p^{12}}$ arithmetic. We recommend to use the first one because if the second one is used, the cyclotomic polynomial Φ_7 is irreducible and can be used for h which improves the algorithm. Then, for $u = -2^{77} + 2^{50} + 2^{33}$, the elliptic curve E (resp. its twist E') is defined over $\mathbb{F}_p$ (resp. $\mathbb{F}_{p^2}$) by

$$E : y^2 = x^3 + 4, \qquad E' : y^2 = x^3 + 4 * (1 + \mathbf{i})$$

E is of course twist-secure (thanks to a 273 prime factor).

6.3 New KSS16 parameter

In this case, the parameter u should have at least 34 bits to ensure the 128-bits security level on the elliptic curve side. Unfortunately, an exhaustive search does not provide any suitable value of the parameter having Hamming weight less than or equal to 5. The sparser parameter we found is $-2^{34} + 2^{27} - 2^{23} + 2^{20} - 2^{11} + 1$. In this case, the extension field is defined by $X^{16} - 2$ which provides the best possible$\mathbb{F}_{p^{16}}$ arithmetic. The elliptic curve E (resp. its twist E') is defined over $\mathbb{F}_p$ (resp. $\mathbb{F}_{p^2}$) by

$$E : y^2 = x^3 + x, \qquad E' : y^2 = x^3 + 2^{\frac{1}{4}} x$$

And again, E is twist-secure (thanks to a 318-bits prime factor). However we found a suitable 35-bits parameter having Hamming weight 5. Such a parameter will of course involve an additional doubling/squaring step in the exponentiation algorithms but it will also involve one addition/multiplication step less. The impact on the Miller loop is negligible, but in the final exponentiation this means that a $\mathbb{F}_{p^{12}}$ multiplication is replaced by a cyclotomic squaring and this happens 9 times since 9 exponentiations by u are performed (see Section 8 for details). Since a cyclotomic squaring is more than twice faster than a $\mathbb{F}_{p^{12}}$ multiplication, it is better to use the 35-bits parameter as long as $\mathbb{F}_p$ arithmetic is not impacted. For example, p has 330 bits for the 34-bits value of u and 340 for the 35-bits value. Hence, if a 32-bits device is used, both values of p require 11 words so the $\mathbb{F}_p$ arithmetic is not impacted. On the contrary, if a 16-bits device is used, choosing the 35-bits value of u implies that p requires 22 words instead of 21. Then the 34-bits value may be preferred in this case. This parameter is $u = 2^{35} - 2^{32} - 2^{18} + 2^8 + 1$, $\mathbb{F}_{p^{16}}$ is also defined by $X^{16} - 2$ and the elliptic curve E (resp. its twist E') is defined over $\mathbb{F}_p$ (resp. $\mathbb{F}_{p^2}$) by

$$E : y^2 = x^3 + x, \qquad E' : y^2 = x^3 + 2^{-\frac{1}{4}} x$$

E is of course twist-secure (thanks to a 281-bits prime factor).

6.4 New KSS18 parameter

Again, the limiting factor for the security level is the elliptic curve size so that u should have at least 44 bits. Our exhaustive search provides no values having weight 2 or 3 and only one having weight 4. It is $u = 2^{44} + 2^{22} - 2^9 + 2$. In this case, $\mathbb{F}_{p^{18}}$ cannot be defined by $X^{18} - 2$ but by $X^{18} - 3$. The elliptic curves are defined by

$$E : y^2 = x^3 + 3, \qquad E' : y^2 = x^3 + 3.3^{\frac{1}{3}} \text{ (or } 3.3^{-\frac{1}{3}}\text{)}$$

The curve E is twist-secure (thanks to a 333-bits prime factor).

7 Effective security of selected curves

Let us now apply the strategy given in Section 5.1 to evaluate the real security of the proposed curves

7.1 BN

We study the curve BN proposed in the previous section, which has parameter $u = 2^{114} + 2^{101} - 2^{14} - 1$.

- **Step1.** The best results are obtained with $\kappa = 2$ and $\eta = 6$. The best choices for the polynomials are $h = t^6 - t^4 + t^2 + 1$, $g = x^2 - t^3 - u$ and $f = P(x^2 - t^3)$. In this case, we have $w = 1$ and $\mathcal{A} = 2$ as in Section 5.2. As a consequence we will find a higher security level than the general case.
- **Step2.** $A = 1134$ and $B = 2^{74.2}$ are minimizing Equation 2 and satisfying Equation 1.

– **Step3.** The size of the sieving space is $(2A+1)^{12}/2 \approx 2^{132.78}$. The Monte Carlo integration gives $\log_2(N_f) \approx 566$ and $\log_2(N_g) \approx 811$. Then the smoothness probabilities are approximatively equal to $\rho\left(\frac{\log_2(N_f)}{\log_2(B)}\right) \approx 2^{-23.08}$ and $\rho\left(\frac{\log_2(N_g)}{\log_2(B)}\right) \approx 2^{-40.07}$. Hence we expect a number of $2^{132.78-23.08-40.07} \approx 2^{69.55}$ relations which is larger than the cardinality of the factor base which is around $2^{69.51}$.
– **Step4.** Evaluating Equation 2 with these datas finally gives an overall complexity of $2^{132.63}$.

7.2 BLS 12

The recommended parameter is $u = -2^{77} + 2^{50} + 2^{33}$.

– **Step 1.** We chose $\kappa = 2$ and $\eta = 6$. The best polynomials are $h = t^6 - t - 1$, $f = P(x^2 + t)$ where $P(x) = (x-1)^2(x^4 - x^2 + 1) + 3x$ and $g = x^2 + t - u$. In this case, we have $w = 7$ and $\mathcal{A} = 2$ (because g is quadratic).
– **Step 2.** $A = 1169$ and $\log_2 B = 73.50$
– **Step 3.**
  • $\log_2(\text{sieve space}) = 133.30$
  • $\log_2(N_f) = 791.2 \Rightarrow \log_2(\text{smoothness probability on the } f \text{ side}) = -39.17$
  • $\log_2(N_g) = 584.8 \Rightarrow \log_2(\text{smoothness probability on the } g \text{ side}) = -24.67$
  • $\log_2(\text{relations}) = 69.46$
  • $\log_2(\text{reduced factor base}) = 67.83$ (enough relations)
– **Step 4.** security=131.8

7.3 KSS 16

The recommended parameter is $u = 2^{35} - 2^{32} - 2^{18} + 2^8 + 1$.

– **Step 1.** We chose $\kappa = 1$ and $\eta = 16$. The best polynomials are $h = \Phi_{17}$, $f = P(x-1)$ and $g = x - u - 1$. In this case, we have $w = 17$ and $\mathcal{A} = 16$.
– **Step 2.** $A = 12$ and $\log_2 B = 80$
– **Step 3.**
  • $\log_2(\text{sieve space}) = 143.52$
  • $\log_2(N_f) = 920.4 \Rightarrow \log_2(\text{smoothness probability on the } f \text{ side}) = -43.23$
  • $\log_2(N_g) = 628.9 \Rightarrow \log_2(\text{smoothness probability on the } g \text{ side}) = -24.21$
  • $\log_2(\text{relations}) = 76.08$
  • $\log_2(\text{reduced factor base}) = 71.04$ (enough relations)
– **Step 4.** security=138.79. Note that this is the security only on the finite field side. The security on the elliptic curve side is 128 as required.

7.4 KSS 18

The recommended parameter is $u = 2^{44} + 2^{22} - 2^9 + 2$.

– **Step 1.** We chose $\kappa = 1$ and $\eta = 18$. The best polynomials are $h = t^{18} - t^4 - t^2 - t - 1$, $f = P(x-2)$ and $g = x - u - 2$. In this case, we have $w = 1$ and $\mathcal{A} = 1$.
– **Step 2.** $A = 11$ and $\log_2 B = 82.5$
– **Step 3.**
  • $\log_2(\text{sieve space}) = 161.85$
  • $\log_2(N_f) = 920.4 \Rightarrow \log_2(\text{smoothness probability on the } f \text{ side}) = -36.21$
  • $\log_2(N_g) = 628.9 \Rightarrow \log_2(\text{smoothness probability on the } g \text{ side}) = -38.33$
  • $\log_2(\text{relations}) = 87.31$
  • $\log_2(\text{reduced factor base}) = 77.66$ (enough relations)
– **Step 4.** security=152.41. Note that this is the security only on the finite field side. The security on the elliptic curve side is 128 as required.

8 Complexity estimations and comparisons for the 128 bits security level

The goal of this section is to compare the pairing computation cost for the curves given in section 6 at the 128 bits security level. For this, we evaluate the cost of an optimal pairing computation in each case (BN, BLS12, KSS16 and KSS18). Let us first recall the steps of the computation

8.1 Optimal Ate pairing computation

We do not give here the detailed algorithm to compute pairings but only what is necessary to analyze its complexity. More details can be found for example in [MJ17]

8.1.1 The Miller loop Miller explains how to compute $f_{u,Q}$ in [Mil04]. The algorithm is based on the computation of $[u]Q$ using the double and add algorithm. At each step of this algorithm, f is updated with the line function involved in the elliptic curve operation. This algorithm has been improved by many authors in particular using the twisted curve to eliminate denominators and replace $\mathbb{F}_{p^k}$ multiplications by sparse ones. The best known complexity for each step are obtained using projective coordinates [CLN10]. They are given below

- If $d = 6$, the doubling step requires one squaring in $\mathbb{F}_{p^k}$ denoted S_k, one sparse multiplication in $\mathbb{F}_{p^k}$ denoted sM_k (for updating f) together with 2 multiplications in $\mathbb{F}_{p^e}$, denoted M_e, 7 squarings in $\mathbb{F}_{p^e}$ and $2e$ multiplications in $\mathbb{F}_p$, denoted M (for doubling on the curve and computing the line involved in this doubling). If $d = 4$, the curve side requires one additional S_e.
- If $d = 6$, the mixed addition step requires one sM_k for updating f together with $11M_e$, $2S_e$ and $2e\,M$ (or $9M_e$, $5S_e$ and $2e\,M$ if $d = 4$).
- Additional lines in the pairing given in Table 2 are nothing but extra addition steps. In term of complexity, the last one is usually less expensive ($4M_e$ and $2e\,M$ for the curve side) because the resulting point on the curve is useless.
- The computation of points of the form $[p]Q$ is very easy because Q is in the p-eigenspace of the Frobenius map. Then it requires no more than 2 Frobenius mapping in $\mathbb{F}_{p^k}$ (denoted F_k). In practice, it requires even less but there is no interest to get into these kind of details for this comparison work.

8.1.2 The final exponentiation It is usually split in 2 parts, an easy one with the exponent $\frac{p^k-1}{\phi_k(p)}$ (where ϕ_k is the k-th cyclotomic polynomial) and a hard one with the exponent $\frac{\phi_k(p)}{r}$. The easy part is made of an inversion (denoted I_k) and few multiplications and Frobenius mappings in $\mathbb{F}_{p^k}$. The hard part is much more expensive but Scott et al. [SBC+09] reduce this cost by writing the exponent in base p (because p-th powering is only a Frobenius mapping). As p is polynomially parametrized by u, the result is obtained thanks to $\deg_u(p)$ exponentiations by u and some additional $\mathbb{F}_{p^k}$ operations. The number of these additional operation can be reduce by considering powers of the pairing [FKR11]. Note also that, thanks to the easy part of the final exponentiation, the squaring operations (which are widely used during the hard part) can be simplified. We can either use cyclotomic squarings [GS10], denoted cS_k or compressed squarings [Kar13,AKL+11], denoted s_k. Compressed squarings are usually more efficient. However, This method have been developed in the case of degree 6 twists [Kar13,AKL+11]. It makes no doubt that it can be adapted to the case of degree 4 twists (and then to KSS16 curves) but we did not find explicit formulas in the literature. Then, for a fairer comparison between the curves, we chose to consider both squaring methods in the following.

8.2 Finite field arithmetic

In order to compare the different candidates, we need a common base. It cannot be the field $\mathbb{F}_p$ because p has not the same size in all cases. So we have to go to the data-words level. We assume that we work on a 32 bits device (as an average between software, FPGA and embedded devices) and that $\mathbb{F}_p$ arithmetic is quadratic (even if the multiplication complexity can be subquadratic, the reduction usually stays quadratic). For simplicity, we will also assume that $\mathbb{F}_p$ multiplications and squarings have almost the same cost and we will neglect additions. Of course, these assumptions are very dependent on the device so we do not pretend that our result is valid in every case. Anyway, our goal here is not to get an universal comparison (which is not possible) but to have an idea of which curve has to be chosen to get the best efficiency. At the end, nothing will replace practical implementations to ensure that one curve is better than another one in a given context.

Pairing computation makes a large use of $\mathbb{F}_{p^e}$ arithmetic. Let us first recall them in Table 11 for the considered values of e.

	$\mathbb{F}_{p^2}$	$\mathbb{F}_{p^3}$	$\mathbb{F}_{p^4}$
Multiplication	$3M$	$6M$	$9M$
Squaring	$2M$	$5M$	$6M$

Table 11: Complexities of $\mathbb{F}_{p^e}$ arithmetic

Concerning the $\mathbb{F}_{p^k}$ arithmetic, the complexities are given in the literature in the pairing context for extensions of degree 12 [AFCK+13,DMHR15], 16 [ZL12] and 18 [AFCK+13]. They are summarized in Table 12

	$\mathbb{F}_{p^{12}}$	$\mathbb{F}_{p^{16}}$	$\mathbb{F}_{p^{18}}$
Multiplication	$54M$	$81M$	$108M$
Sparse multiplication	$39M$	$63M$	$78M$
Inversion	$I + 97M$	$I + 134M$	$I + 172M$
Frobenius	$11M$	$15M$	$17M$
Squaring	$36M$	$54M$	$66M$
Cyclotomic squaring	$18M$	$36M$	$36M$
Compressed squaring	$12M$	–	$24M$
Simult. decompression of n elements	$I + (24n - 5)M$	–	$I + (51n - 6)M$

Table 12: Complexities of $\mathbb{F}_{p^k}$ arithmetic

We made the simplistic assumption that the cost of Frobenius mapping in $\mathbb{F}_{p^k}$ is always $(k-1)M$ which is not always the case (for example for p^2 or p^3 powering) but this has negligible impact on our comparison (there are few such mapping and this remark holds for all the considered cases).

8.3 $\mathbb{F}_p$ complexities estimations

8.3.1 BN curve In this case, the optimal Ate pairing is given by

$$\left(f_{6u+2,Q}(P).\ell_{[6u+2]Q,[p]Q}(P).\ell_{[6u+2+p]Q,[p^2]Q}(P)\right)^{\frac{p^{12}-1}{r}}$$

It is explained in Section 6 that $u = 2^{114} + 2^{101} - 2^{14} - 1$ should be chosen to ensure the 128-bits security level and the best possible extension field arithmetic. Then $6u + 2$ has length 116 and Hamming weight 7. As a consequence, the Miller loop requires 116 doubling steps and 6 addition steps. Extra lines computations require 4 Frobenius mapping (to compute $[p]Q$ and $[p^2]Q$), one addition step and one incomplete addition step. Then the overall cost is

$$116(2M_2{+}7S_2{+}4M){+}115(S_{12}{+}sM_{12}){+}7(11M_2{+}2S_2{+}4M){+}4M_2{+}4M{+}8sM_{12}{+}4F_{12}$$

Using Tables 12 and 11, this step requires 12068 multiplications in $\mathbb{F}_p$.
There are many ways to compute the final exponentiation for BN curves. The most efficient one is given in [FKR11] and requires $I_{12} + 12M_{12} + 3cS_{12} + 4F_{12}$ in addition to the 3 exponentiation by u (because p has degree 4 in u). As u has length 114 and Hamming weight 4, each of these exponentiations requires 114 squarings and 3 multiplications. If the cyclotomic squaring are used, we need $114cS_{12} + 3M_{12} = 2214M$ according Table 12. If the compressed squaring technique is used, we additionally need the simultaneous decompression of 4 elements. Then, according to Table 12 each exponentiation by u requires $1621M + I$.
The final exponentiation then requires $7485M + I$ or $5706M + 4I$ depending on the way to perform squarings. Finally computing the optimal Ate pairing for BN curve ensuring the 128-bits security level requires $19553M + I$ or $17774M + 4I$ depending on the way to perform squarings.

8.3.2 BLS12 curve The optimal Ate pairing is simpler in this case since it is given by

$$(f_{u,Q}(P))^{\frac{p^{12}-1}{r}}$$

We have seen that the best choice of u is $-2^{77}-2^{59}+2^{9}$ so that the Miller loop is made of 77 doubling steps and 2 addition steps. Then, its cost is

$$77(2M_2{+}7S_2{+}4M){+}76(S_{12}{+}sM_{12}){+}2(11M_2{+}2S_2{+}4M){+}2sM_{12}=7708M$$

According [AFCK+13], the final exponentiation requires $I_{12}+12M_{12}+2cS_{12}+4F_{12}=825M+I$ in addition to the 5 exponentiation by u (because p has degree 6 in u). As u has length 77 and Hamming weight 3, each of these exponentiations requires 77 squarings and 2 multiplications. If the cyclotomic squaring are used, we need $77cS_{12}+2M_{12}=1494M$. If the compressed squaring technique is used, we additionally need the simultaneous decompression of 3 elements so that each exponentiation by u requires $1099M+I$.
The final exponentiation then requires $8295M+I$ or $6320M+6I$ depending on the way to perform squarings. Finally computing the optimal Ate pairing for BLS12 curve ensuring the 128-bits security level requires $16003M+I$ or $14028M+6I$.

8.3.3 KSS16 curve For KSS16 curves, the optimal Ate pairing is given by

$$\left(\left(f_{u,Q}(P).\ell_{[u]Q,[p]Q}(P)\right)^{p^3}.\ell_{Q,Q}(P)\right)^{\frac{p^{16}-1}{r}}$$

and u has been chosen to be $2^{35}-2^{32}-2^{18}+2^{8}+1$ in section 6. Then the Miller loop requires 35 doubling steps and 4 addition steps. According [ZL12], extra lines computations require 3 Frobenius mapping (2 to compute $[p]Q$ and one to raise to p^3) and two incomplete addition steps. The overall cost is then

$$35(2M_4{+}8S_4{+}8M){+}34(S_{16}{+}sM_{16}){+}4(9M_4{+}5S_4{+}8M){+}3F_{16}+5M_4{+}S_4{+}16M{+}6sM_{16}=7534M$$

According [GF16], the final exponentiation requires $I_{16}+32M_{16}+34cS_{16}+24M_4+8F_{16}$ in addition to the 9 exponentiation by u (because p has degree 10 in u). As u has length 35 and Hamming weight 5, each of these exponentiations requires 35 cyclotomic squarings and 4 multiplications. According Table 12, each exponentiation by u then requires $1584M$. Note that we do not find in the literature formulas for compressed squaring in the KSS16 case. The final exponentiation then requires $18542M+I$. Finally computing the optimal Ate pairing for KSS16 curve ensuring the 128-bits security level requires $26076M+I$.

8.3.4 KSS18 curve In this case, the optimal Ate pairing is given by

$$\left(f_{u,Q}(P).f_{3,Q}(P)^p.\ell_{[u]Q,[3p]Q}(P)\right)^{\frac{p^{18}-1}{r}}$$

The best choice of u to ensure the 128-bits security level is $2^{44}+2^{22}-2^{9}+2$ so that the Miller loop is made of 44 doubling steps and 3 addition steps. Extra lines computations requires one addition step and one Frobenius mapping (to compute $f_{3,Q}(P)^p$) together with one $\mathbb{F}_{p^{18}}$ multiplication (to multiply the result by $f_{u,Q}(P)$), 2 Frobenius mappings and one incomplete addition step [AFCK+13]. Then its cost is

$$44(2M_3{+}7S_3{+}6M){+}43(S_{18}{+}sM_{18}){+}4(11M_3{+}2S_3{+}6M){+}4sM_{18}{+}M_{18}{+}3F_{18}{+}4M_3{+}6M{+}sM_{18}=9431M$$

According [FKR11,AFCK+13], the final exponentiation requires $I_{18}+54M_{18}+8cS_{18}+29F_{18}=6785M+I$ in addition to the 7 exponentiation by u (because p has degree 8 in u). As u has length 44 and Hamming weight 4, each of these exponentiations requires 44 squarings and 3 multiplications. If the cyclotomic squaring are used, we need $44cS_{18}+3M_{18}=1908M$. If the compressed squaring technique is used, we additionally need the simultaneous decompression of 4 elements so that each exponentiation by u requires $1578M+I$.
The final exponentiation then requires $20141M+I$ or $17831M+8I$ depending on the way to perform squarings. Finally computing the optimal Ate pairing for KSS18 curve ensuring the 128-bits security level requires $29572M+I$ or $27262M+8I$.

8.4 Comparison

Let us first summarize the complexities obtained in the previous subsections.

	Using cyclotomic squarings	Using compressed squarings	Base field size
BN	$19553M + I$	$17774M + 4I$	461 bits
BLS12	$16003M + I$	$14028M + 6I$	461 bits
KSS16	$26076M + I$	–	340 bits
KSS18	$29572M + I$	$27262M + 8I$	348 bits

Table 13: $\mathbb{F}_p$ complexities of optimal Ate pairing computation

We can obviously conclude that BLS12 curve is more efficient than BN one and that KSS16 is better than KSS18. It is more complicated to compare BLS12 and KSS16 because the base fields are not the same. For this, let us first compare the costs of M which is depending of p. For BN and BLS12 curves, p has 461 bits so that 15 32-bits words are necessary. For the KSS curves, 11 32-bits words are necessary. As a consequence, we can assume that $M = 15^2 = 225$ for BN and BLS12 curves while $M = 11^2 = 121$ for KSS ones. Reporting these values in Table 13, we get the comparative table 14.

	Using cyclotomic squarings	Using compressed squarings
BN	$4399425 + I$	$3999150 + 4I$
BLS12	$3600675 + I$	$3156300 + 6I$
KSS16	$3155196 + I$	–
KSS18	$3578212 + I$	$3298702 + 8I$

Table 14: Comparative complexities of optimal Ate pairing computation

In any case, the KSS16 curve gives the best result which was not expected at the beginning of this work. Of course the complexity for the BLS12 curve using compressed squaring is very close to the complexity of the KSS16 curve with cyclotomic squarings and a practical implementation should be done to confirm the estimated result obtained here. But KSS16 curves have been very few studied compared to BN curves and more generally to curves having a degree 6 twist. Then we are quite confident that optimal pairing on the KSS16 curve given in section 7.3 can be improved for example by computing the formulas for compressed squaring in this case.

9 Higher security

9.1 Some curves for 192 bits of security

In the case of higher levels of security we prefer to be more cautious. Instead of a comparison of the best curves we simply give our own propositions. In terms of security we are once again cautious, our curves having 202 bits of security instead of 192, which is a safety margin in case of theoretic progress on NFS. We give only a KSS18 and a BLS24 curve since it makes no doubt that BN, BLS12 and KSS16 will be less efficient.

9.1.1 KSS18 curve We saw in Table 9 that the parameter u should be chosen such that $\log_2(u) \geq 85$. As in the 128-bits case, we perform an exhaustive search of low Hamming weight values for u. The best value we found is $u = -2^{85} - 2^{31} - 2^{26} + 2^6$. In this case, $\mathbb{F}_{p^{18}}$ can be defined by $X^{18} - 2$. The elliptic curves are defined by

$$E : y^2 = x^3 + 2, \qquad E' : y^2 = x^3 + 2.2^{\frac{1}{3}} \text{ (or } 2.2^{-\frac{1}{3}})$$

The curve E is twist-secure (thanks to a 652-bits prime factor). To evaluate its real security, we use the way described in Section 5.1 and we get

- **Step 1.** We chose $\kappa = 1$ and $\eta = 18$. The best polynomials are $h = t^{18} - t^4 - t^2 - t - 1$, $f = P(x-2)$ and $g = x - u - 2$. In this case, we have $w = 1$ and $\mathcal{A} = 1$.
- **Step 2.** $A = 34$ and $\log_2 B = 108.9$
- **Step 3.**
  - $\log_2(\text{sieve space}) = 161.85$
  - $\log_2(N_f) = 1114 \Rightarrow \log_2(\text{smoothness probability on the } f \text{ side}) = -36.29$
  - $\log_2(N_g) = 1642 \Rightarrow \log_2(\text{smoothness probability on the } g \text{ side}) = -63.99$
  - $\log_2(\text{relations}) = 118.62$
  - $\log_2(\text{reduced factor base}) = 103.66$ (enough relations)
- **Step 4.** security $= 204.09$.

9.1.2 BLS 24 curve

We saw in Table 9 that the parameter u should be chosen such that $\log_2(u) \geq 85$. As in the 128-bits case, we perform an exhaustive search of low Hamming weight values for u. The best value we found is $u = -2^{56} - 2^{43} + 2^9 - 2^6$. In this case, $\mathbb{F}_{p^{24}}$ can be build via $Y^{12} - (1 + \mathbf{i})$ over $\mathbb{F}_{p^2} = \mathbb{F}_p[\mathbf{i}]$ which provides the best possible $\mathbb{F}_{p^{24}}$ arithmetic. The elliptic curves are defined by

$$E : y^2 = x^3 - 2, \qquad E' : y^2 = x^3 - 2/(1 + \mathbf{i})$$

E is of course twist-secure (thanks to a 427 prime factor). To evaluate its real security, we use the way described in Section 5.1 and we get

- **Step 1.** We chose $\kappa = 1$ and $\eta = 18$. The best polynomials are $h = t^{24} + t^4 - t^3 - t - 1$, $f = P(x-2)$ and $g = x - u - 2$. In this case, we have $w = 1$ and $\mathcal{A} = 1$.
- **Step 2.** $A = 9$ and $\log_2 B = 109.8$
- **Step 3.**
  - $\log_2(\text{sieve space}) = 202.90$
  - $\log_2(N_f) = 1295 \Rightarrow \log_2(\text{smoothness probability on the } f \text{ side}) = -44.85$
  - $\log_2(N_g) = 1460 \Rightarrow \log_2(\text{smoothness probability on the } g \text{ side}) = -53.42$
  - $\log_2(\text{relations}) = 104.63$
  - $\log_2(\text{reduced factor base}) = 104.55$ (enough relations)
- **Step 4.** security $= 203.72$.

9.2 Subgroup-secure curves

All the curves provided are not protected against the so-called subgroup attacks which use the fact that the 3 groups involved in the pairing may have small cofactors [LL97]. They can be avoided by the use of some (potentially expensive) subgroup membership tests or by choosing resistant parameters. The definition of subgroup security for pairing is given in [BCM+15] and implies that one should be able to find factors of $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_3$. This can be done using the ECM method but it is very costly so one cannot perform an exhaustive search checking subgroup security at each step. As explained in [BCM+15], the most reasonable way to find a subgroup-secure curve for pairing applications is to find a parameter u such that $\#\mathbb{G}_2/r$ and $\#\mathbb{G}_3/r$ are primes. This is of course much easier to check but on the other hand there are much fewer candidates.

According Section 8, we are only interested in BLS12 and KSS16 curves i the case of security level 128. We then made an exhaustive search of increasing Hamming weight values of u satisfying this condition. For BLS12 curves, we find some parameters in weight 7. We give only one here but the other ones are not so difficult to find: $u = -2^{77} - 2^{71} - 2^{64} + 2^{37} + 2^{35} + 2^{22} - 2^5$. In this cases $\mathbb{F}_{p^{12}}$ can be build via $Y^6 - (1 + \mathbf{i})$ over $\mathbb{F}_{p^2} = \mathbb{F}_p[\mathbf{i}]$ which provides the best possible $\mathbb{F}_{p^{12}}$ arithmetic. The elliptic curve E (resp. its twist E') is defined over $\mathbb{F}_p$ (resp. $\mathbb{F}_{p^2}$) by

$$E : y^2 = x^3 - 2, \qquad E' : y^2 = x^3 - 2/(1 + \mathbf{i})$$

E is of course twist-secure (thanks to a 433 prime factor).

The case of KSS16 curves is more complicated. We first remark that $\#\mathbb{G}_2/r$ and $\#\mathbb{G}_3/r$ are always even and often divisible by 17 [GF16] so we have interest to relax the condition. Unfortunately it was not sufficient to find a parameter of Hamming weight less than or equal to 10. This is due to the fact that $\log_2(u) = 34$ implies that there are not enough possibilities for u to have a reasonable probability that all the numbers involved (p, r, $\#\mathbb{G}_2/2r$, $\#\mathbb{G}_3/2r$) are primes together (up to some 17^n factor).

Conclusion

It was already known that the BN curve widely used in the literature for the 128-bits security level does not ensure this security level because of the SexTNFS algorithm. In this paper, we carefully estimate the complexity of this algorithm in the context of most common pairing families. As a consequence, we give the updated security level of this curve which is in fact 100 bits. We also use this complexity estimation to determine the sizes of the finite field extensions that has to be used to ensure the 128 and the 192-bits security level and then give recommendations on the sizes of the parameter to be used depending on the pairing family. According to these recommendations, we generate new pairing parameters especially in the 128-bits security level that are twist-secure (but also some that are twist and subgroup-secure). Finally, we estimate the complexity of the optimal-Ate pairing in each case and conclude that, at the 128-bits security level, BLS12 and more surprisingly KSS16 are the most efficient choices. Then we encourage the community to study more precisely these curves and to propose software or hardware implementation to confirm our conclusions. We also provide some parameters for the 192-bits security level but our study is probably not complete in this case since other families and/or embedding degrees could be more interesting.

A Numerical integration

The size of the norms can by computed via numerical methods. Due to the known upperbounds we can certify that our results are correct up to an error probability of 2^{-128}, so that our chances to be wrong are equal to the chances of an attacker to break the system by pure luck.

Given a polynomial f and a sieve parameter A let $c(f, A)$ be the average of value of $\{\log_2 N_f(e) \mid e \text{ tuple in sieving domain}\}$ and $U(f, A)$ an upper bound on the norms on the f side for pairs in the sieving domain. Let $e_1, \ldots, e_T$ be random tuples in the sieving domain, uniformly and independently chosen. Then the Chernoff theorem applied to the random variables $\frac{\log_2 N_f(e_1)}{\log_2 U(f)}, \ldots, \frac{\log_2 N_f(e_T)}{\log_2 U(f)}$ states that for any constant $\varepsilon > 0$

$$\mathrm{Prob}\left(|c(f,A) - +\frac{1}{T}\sum_{i=1}^{T}\log_2 N_f(e_i)| < \varepsilon \log_2 U(f,A)\right) \leq 2e^{-2\varepsilon^2 T}. \quad (3)$$

For $\varepsilon = 0.05$ we solve the equation $e^{-2\varepsilon^2 T} = 2^{-128}$ and obtain $T = 25600$. When plugged in

References

Adl94. L. M. Adleman. The function field sieve. In *Algorithmic Number Theory–ANTS I*, volume 877 of *Lecture Notes in Comput. Sci.*, pages 108–121. Springer, 1994.

AFCK+13. Diego F. Aranha, Laura Fuentes-Castañeda, Edward Knapp, Alfred Menezes, and Francisco Rodríguez-Henríquez. *Implementing Pairings at the 192-Bit Security Level*, pages 177–195. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

AFK+07. Kazumaro Aoki, Jens Franke, Thorsten Kleinjung, Arjen K Lenstra, and Dag Arne Osvik. A kilobit special number field sieve factorization. In *Advances in Cryptology – ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Comput. Sci.*, pages 1–12, 2007.

AH99. L. M. Adleman and M. D. A. Huang. Function field sieve method for discrete logarithms over finite fields. *Information and Computation*, 151(1):5–16, 1999.

AKL+11. D. Aranha, K. Karabina, P. Longa, C. H. Gebotys, and J López. Faster explicit formulas for computing pairings over ordinary curves. In *Advances in Cryptology EUROCRYPT 2011*, volume 6632 of *LNCS*, pages 48–68. Springer, 2011.

BCM$^+$15. Paulo S. L. M. Barreto, Craig Costello, Rafael Misoczki, Michael Naehrig, Geovandro C. C. F. Pereira, and Gustavo Zanon. *Subgroup Security in Pairing-Based Cryptography*, pages 245–265. Springer International Publishing, Cham, 2015.

BF01. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the Weil pairing. In Joe Kilian, editor, *Advances in Cryptology - CRYPTO 2001*, volume 2139 of *LNCS*, pages 213–229, 2001.

BGGM15. Razvan Barbulescu, Pierrick Gaudry, Aurore Guillevic, and François Morain. Improving NFS for the discrete logarithm problem in non-prime finite fields. In *Advances in Cryptology - EUROCRYPT 2015*, volume 9056 of *Lecture Notes in Comput. Sci.*, pages 129–155, 2015.

BGJT14. Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, and Emmanuel Thomé. A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic. In *Advances in Cryptology - EUROCRYPT 2014*, volume 8441 of *Lecture Notes in Comput. Sci.*, pages 1–16, 2014.

BGK15. Razvan Barbulescu, Pierrick Gaudry, and Thorsten Kleinjung. The Towed Number Field Sieve. In *Advances in Cryptology – ASIACRYPT 2015*, volume 9453 of *Lecture Notes in Comput. Sci.*, pages 31–55, 2015.

BGW05. Dan Boneh, Craig Gentry, and Brent Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In Victor Shoup, editor, *Advances in Cryptology - CRYPTO 2005*, volume 3621 of *LNCS*, pages 258–275, 2005.

BKK$^+$09. Joppe Bos, Marcelo Kaihara, Thorsten Kleinjung, Arjen K Lenstra, and Peter L Montgomery. On the security of 1024-bit rsa and 160-bit elliptic curve cryptography, 2009.

BL16. R. Barbulescu and A. Lachand. Some mathematical remarks on the polynomial selection of nfs. *Math. Comp.*, 2016. Published electronically at http://dx.doi.org/10.1090/mcom/3112.

BLP93. J Buhler, H Lenstra, and Carl Pomerance. Factoring integers with the number field sieve. *The development of the number field sieve*, pages 50–94, 1993.

BLS03. Paulo S. L. M. Barreto, Ben Lynn, and Michael Scott. *Constructing Elliptic Curves with Prescribed Embedding Degrees*, pages 257–267. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003.

BLS04. Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. *Journal of cryptology*, 17(4):297–319, 2004.

BN05. Paulo SLM Barreto and Michael Naehrig. Pairing-friendly elliptic curves of prime order. In *International Workshop on Selected Areas in Cryptography–SAC 2005*, volume 3006 of *Lecture Notes in Comput. Sci.*, pages 319–331. Springer, 2005.

Cav02. Stefania Hedwig Cavallar. *On the number field sieve integer factorisation algorithm*. Universiteit Leiden, 2002.

CDF$^+$11. Ray C. C. Cheung, Sylvain Duquesne, Junfeng Fan, Nicolas Guillermin, Ingrid Verbauwhede, and Gavin Xiaoxu Yao. FPGA implementation of pairings using residue number system and lazy reduction. In *Cryptographic Hardware and Embedded Systems - CHES 2011*, volume 6917 of *LNCS*, pages 421–441. Springer, 2011.

CEP83. E. R. Canfield, P. Erdös, and C. Pomerance. On a problem of Oppenheim concerning "factorisatio numerorum", 1983.

CLN10. Craig Costello, Tanja Lange, and Michael Naehrig. Faster pairing computations on curves with high-degree twists. In Phong Q. Nguyen and David Pointcheval, editors, *Public Key Cryptography - PKC 2010*, volume 6056 of *Lecture Notes in Computer Science*, pages 224–242. Springer Berlin Heidelberg, 2010.

Det14. Jérémie Detrey. Ffs factory: Adapting coppersmith's "factorization factory" to the function field sieve. Cryptology ePrint Archive, Report 2014/419, 2014. http://eprint.iacr.org/2014/419.

DMHR15. Sylvain Duquesne, Nadia El Mrabet, Safia Haloui, and Franck Rondepierre. Choosing and generating parameters for low level pairing implementation on bn curves. Cryptology ePrint Archive, Report 2015/1212, 2015. http://eprint.iacr.org/2015/1212.

DSD. Augusto Jun Devegili, Michael Scott, and Ricardo Dahab. Implementing cryptographic pairings over barreto-naehrig curve. pages 197–207.

FGHT16. Joshua Fried, Pierrick Gaudry, Nadia Heninger, and Emmanuel Thomé. A kilobit hidden snfs discrete logarithm computation. *arXiv preprint arXiv:1610.02874*, 2016.

FKR11. Laura Fuentes-Castañeda, Edward Knapp, and Francisco Rodríguez-Henríquez. Faster hashing to $\mathbb{G}_2$. In *Selected Areas in Cryptography - 18th International Workshop, SAC 2011, Toronto, ON, Canada, August 11-12, 2011, Revised Selected Papers*, Lecture Notes in Computer Science, pages 412–430. Springer, 2011.

FST10. David Freeman, Michael Scott, and Edlyn Teske. A taxonomy of pairing-friendly elliptic curves. *Journal of cryptology*, 23(2):224–280, 2010.

GAL$^+$13. Gurleen Grewal, Reza Azarderakhsh, Patrick Longa, Shi Hu, and David Jao. Efficient implementation of bilinear pairings on arm processors. In LarsR. Knudsen and Huapeng Wu, editors, *Selected Areas in Cryptography*, volume 7707 of *Lecture Notes in Computer Science*, pages 149–165. Springer Berlin Heidelberg, 2013.

GF16. Loubna Ghammam and Emmanuel Fouotsa. *Adequate Elliptic Curves for Computing the Product of n Pairings*, pages 36–53. Springer International Publishing, Cham, 2016.

GJNB11. C. C. F. Pereira Geovandro, Marcos A. Simplício Jr., Michael Naehrig, and Paulo S. L. M. Barreto. A family of implementation-friendly bn elliptic curves. *Journal of Systems and Software*, 84(8):1319–1326, 2011.

Gor93. Daniel M. Gordon. Discrete logarithms in $GF(p)$ using the number field sieve. *SIAM J. Discret. Math.*, 6(1):124–138, February 1993.

GS10. Robert Granger and Michael Scott. Faster squaring in the cyclotomic subgroup of sixth degree extensions. In *Public Key Cryptography - PKC 2010, 13th International Conference on Practice and Theory in Public Key Cryptography, Paris, France, May 26-28, 2010. Proceedings*, pages 209–223, 2010.

HSST12. T. Hayashi, T. Shimoyama, N. Shinohara, and T. Takagi. Breaking pairing-based cryptosystems using η_t pairing over GF(3^{97}). In *Advances in Cryptology–ASIACRYPT 2012*, volume 7658 of *Lecture Notes in Comput. Sci.*, pages 43–60, 2012.

HSV06. Florian Hess, Nigel P. Smart, and Frederik Vercauteren. The eta pairing revisited. *IEEE Transactions on Information Theory*, 52(10):4595–4602, 2006.

HSW$^+$10. T. Hayashi, N. Shinohara, L. Wang, S. Matsuo, M. Shirase, and T. Takagi. Solving a 676-bit discrete logarithm problem in GF(3^{6n}). In *Public Key Cryptography–PKC 2010*, volume 6056 of *Lecture Notes in Comput. Sci.*, pages 351–367. Springer, 2010.

JL02. A. Joux and R. Lercier. The function field sieve is quite special. In *Algorithmic Number Theory–ANTS V*, volume 2369 of *Lecture Notes in Comput. Sci.*, pages 431–445. Springer, 2002.

JL03. A. Joux and R. Lercier. Improvements to the general number field for discrete logarithms in prime fields. *Math. Comp.*, 72(242):953–967, 2003.

JL06. A. Joux and R. Lercier. The function field sieve in the medium prime case. In *Advances in Cryptology–EUROCRYPT 2006*, volume 4005 of *Lecture Notes in Comput. Sci.*, pages 254–270. Springer, 2006.

JLSV06. Antoine Joux, Reynald Lercier, Nigel P. Smart, and Frederik Vercauteren. The number field sieve in the medium prime case. In *Advances in Cryptology - CRYPTO 2006*, volume 4117 of *Lecture Notes in Comput. Sci.*, pages 326–344, 2006.

Jou13. A. Joux. Faster index calculus for the medium prime case application to 1175-bit and 1425-bit finite fields. In *Advances in Cryptology–EUROCRYPT 2013*, volume 7881 of *Lecture Notes in Comput. Sci.*, pages 177–193. Springer, 2013.

Jou14. Antoine Joux. A new index calculus algorithm with complexity l(1/4+o(1)) in small characteristic. In Tanja Lange, Kristin Lauter, and Petr Lisoněk, editors, *Selected Areas in Cryptography – SAC 2013*, volume 8282 of *Lecture Notes in Computer Science*, pages 355–379. Springer Berlin Heidelberg, 2014.

JP13. Antoine Joux and Cécile Pierrot. The special number field sieve in $\mathbb{F}_{p^n}$ – application to pairing-friendly constructions. In *Pairing-Based Cryptography - Pairing 2013*, volume 8365 of *Lecture Notes in Comput. Sci.*, pages 45–61, 2013.

Kar13. Koray Karabina. Squaring in cyclotomic subgroups. *Math. Comput.*, 82(281), 2013.

KB16. Taechan Kim and Razvan Barbulescu. The extended tower number field sieve: A new complexity for the medium prime case. In *Advances in Cryptology – CRYPTO 2016*, volume 9814 of *Lecture Notes in Comput. Sci.*, pages 543–571, 2016.

KBL14. Thorsten Kleinjung, Joppe W Bos, and Arjen K Lenstra. Mersenne factorization factory. In *International Conference on the Theory and Application of Cryptology and Information Security*, volume 8873 of *Lecture Notes in Comput. Sci.*, pages 358–377. Springer, 2014.

KDL$^+$16. Thorsten Kleinjung, Claus Diem, Arjen K. Lenstra, Christine Priplata, and Colin Stahlke. Discrete logarithms in GF(p) — 768 bits, 2016. Announcement available at the NMBRTHRY archives, item 004917.

KSS08. Ezekiel J. Kachisa, Edward F. Schaefer, and Michael Scott. *Constructing Brezing-Weng Pairing-Friendly Elliptic Curves Using Elements in the Cyclotomic Field*, pages 126–135. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

Len01a. Arjen K Lenstra. Unbelievable security matching aes security using public key systems. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 67–86. Springer, 2001.

Len01b. Arjen K Lenstra. Unbelievable security: Matching AES security using public key systems. In *Advances in cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Comput. Sci.*, pages 67–86, 2001.

LL97. Chae Hoon Lim and Pil Joong Lee. *A key recovery attack on discrete log-based schemes using a prime order subgroup*, pages 249–263. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

LLJMP90. Arjen K Lenstra, Hendrik W Lenstra Jr, Mark S Manasse, and John M Pollard. The number field sieve. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 564–572. ACM, 1990.

LN97. Rudolf Lidl and Harald Niederreiter. *Finite Fields*. Cambridge University Press, 1997.

Mat06. D. Matyukhin. Effective version of the number field sieve for discrete logarithms in the field GF(p^k) (in Russian). *Trudy po Discretnoi Matematike*, 9:121–151, 2006.

Mil04. Victor S Miller. The weil pairing, and its efficient calculation. *Journal of Cryptology*, 17(4):235–261, 2004.

MJ17. N.E. Mrabet and M. Joye. *Guide to Pairing-Based Cryptography*. Chapman & Hall/CRC Cryptography and Network Security Series. CRC Press, 2017.

MSS16. Alfred Menezes, Palash Sarkar, and Shashank Singh. Challenges with assessing the impact of nfs advances on the security of pairing-based cryptography. In *Proceedings of Mycrypt*, 2016.

Mur98. Brian Murphy. Modelling the yield of number field sieve polynomials. In *Algorithmic Number Theory–ANTS III*, volume 1423 of *Lecture Notes in Comput. Sci.*, pages 137–150. Springer, 1998.

Nat12. National Institute of Standards and Technology (NIST). Nist special publication 800-57 part 1 (revised) : Recommendation for key management, part 1: General (revised), July 2012. Publication available online at http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57-Part1-revised2_Mar08-2007.pdf.

NNS10. Michael Naehrig, Ruben Niederhagen, and Peter Schwabe. New software speed records for cryptographic pairings. In *LATINCRYPT 2010*, volume 6212 of *LNCS*, pages 109–123. Springer, 2010.

SBC+09. Michael Scott, Naomi Benger, Manuel Charlemagne, Luis J. Dominguez Perez, and Ezekiel J. Kachisa. On the final exponentiation for calculating pairings on ordinary elliptic curves. In *Pairing-Based Cryptography - Pairing 2009, Third International Conference, Palo Alto, CA, USA, August 12-14, 2009, Proceedings*, Lecture Notes in Computer Science, pages 78–88. Springer, 2009.

Sch93. Oliver Schirokauer. Discrete logarithms and local units. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 345(1676):409–423, 1993.

Sch00. Oliver Schirokauer. Using number fields to compute logarithms in finite fields. *Math. Comput.*, 69(231):1267–1283, 2000.

SS16a. Palash Sarkar and Shashank Singh. Fine tuning the function field sieve algorithm for the medium prime case. *IEEE Transactions on Information Theory*, 62(4):2233–2253, 2016.

SS16b. Palash Sarkar and Shashank Singh. A generalisation of the conjugation method for polynomial selection for the extended tower number field sieve algorithm. Cryptology ePrint Archive, Report 2016/537, 2016. http://eprint.iacr.org/2016/537.

SS16c. Palash Sarkar and Shashank Singh. New complexity trade-offs for the (multiple) number field sieve algorithm in non-prime fields. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 429–458. Springer, 2016.

SS16d. Palash Sarkar and Shashank Singh. Tower number field sieve variant of a recent polynomial selection method. Cryptology ePrint Archive, Report 2016/401, 2016. http://eprint.iacr.org/2016/401.

UW14. Thomas Unterluggauer and Erich Wenger. Efficient pairings and ecc for embedded systems. In Lejla Batina and Matthew Robshaw, editors, *Cryptographic Hardware and Embedded Systems - CHES 2014*, volume 8731 of *Lecture Notes in Computer Science*, pages 298–315. Springer Berlin Heidelberg, 2014.

Ver09. Frederik Vercauteren. Optimal pairings. *IEEE Transactions of Information Theory*, 56:455–461, 2009.

VLFR08. Frédéric Valette, Reynald Lercier, Pierre-Alain Fouque, and Denis Réal. Fault attack on elliptic curve montgomery ladder implementation. *2013 Workshop on Fault Diagnosis and Tolerance in Cryptography*, 00:92–98, 2008.

ZL12. Xusheng Zhang and Dongdai Lin. Analysis of optimum pairing products at high security levels. In *Progress in Cryptology - INDOCRYPT 2012, 13th International Conference on Cryptology in India, Kolkata, India, December 9-12, 2012. Proceedings*, Lecture Notes in Computer Science, pages 412–430. Springer, 2012.