A brief history of pairings

Razvan Barbulescu

A brief history of pairings

Razvan Barbulescu

CNRS, Univ Paris 6, Univ Paris 7, France
razvan.barbulescu@imj-prg.fr

Abstract. Pairings are a relatively new tool in cryptography. Recent progress on the attack algorithms have changed the security estimations. We make a list of pairing families and explain their advantages but also their weaknesses.

1 Introduction

Pairings are a mathematical tool which has been known to cryptographers for a long time and which switched sides during its history. If in the early 90's it was on the attacker's side, it is now used to create secure cryptologic protocols.

Let E be an elliptic curve defined over a finite field $\mathbb{F}_q$, r an integer number, P a point of order r and μ an rth root of unity in the algebraic closure $\overline{\mathbb{F}_q}$. The Weil pairing (restricted to the subgroup generated by P) is the map

$$\begin{array}{cccc} & e : \mathbb{Z}/r\mathbb{Z}P \times \mathbb{Z}/r\mathbb{Z}P & \to & \mu^{\mathbb{Z}/r\mathbb{Z}} \\ \forall (a,b) \in (\mathbb{Z}/r\mathbb{Z})^2 & ([a]P, [b]P) & \mapsto & \mu^{ab}. \end{array} \tag{1}$$

Two properties of the Weil pairing are direct:

- bilinearity: for all a, a', b, b' we have

$$\begin{aligned} e([a+a']P, [b]P) &= e([a]P, [b]P) \cdot e([a']P, [b]P) \\ e([a]P, [b+b']P) &= e([a]P, [b]P) \cdot e([a]P, [b']P) \end{aligned}$$

- non-degeneracy: for all $a \neq 0$ there exists b so that

$$e([a]P, [b]P) \neq 1,$$

  and similarly with the roles of a and b exchanged.

The Weil pairing owes its name to André Weil who gave an equivalent definition in 1940 [Wei40]. More precisely, Weil defined the map

$$e_W(S,T) = \frac{g(X+S)}{g(X)}, \tag{2}$$

where g is a function so that $\mathrm{div}(g) = rT - r\mathcal{O}_E$ and $X \in E \backslash E[r^2]$. This map is bilinear and non-degenerate (see Proposition III.8.1 in [Sil07]) and, since there is a unique map with these two properties (up to a multiplicative constant), we

conclude that Equations (2) and (1) are alternative definitions of the same object. In 1985 Miller [Mil04] invented an algorithm based on this equivalent definition to compute e in polynomial time with respect to the bit sizes of q and r. Frey and Rück [FR94] created an alternative manner to compute Weil pairings using results of Tate and Lichtenbaum.

From attacker's point of view, pairings are a tool to reduce hard problems to easier ones. Given a cyclic group G of known order, a generator P of G and an other point $[a]P$ for $a \in \{0, 1, \ldots, \#G - 1\}$, the discrete logarithm problem (DLP) consists in finding a. In 1992 Menezes, Okamoto and Vanstone [MOV93] showed that the Weil pairing associated to an elliptic curve over $\mathbb{F}_q$, an integer r, a point P of order r and an rth root of unity in $\overline{\mathbb{F}_q}$ allows to reduce the DLP on E to the DLP in the multiplicative group of $\mathbb{F}_q(\mu)$, the smallest subfield of $\overline{\mathbb{F}_q}$ which contains μ. The embedding degree of E with respect to r is the degree of $\mathbb{F}_q(\mu)$.

From a constructive point of view, pairings are a tool to combine two encrypted secrets into a common encrypted secret, without decrypting them at any time. In 2001 Joux [Jou00] proposed a three-party Diffie-Hellman key exchange which requires only one round of communications. If Alice, Bob and Carol want to agree on a common key they need to agree on an elliptic curve and on a point P of order r. Then they proceed in two steps

1. each participant generates a random integer, raises P to that power and broadcasts the result:
   - Alice generates a and computes $[a]P$ and broadcasts it,
   - Bob generates b and computes $[b]P$ and broadcasts it,
   - Carol generates c and computes $[c]P$ and broadcasts it;
2. each participant computes the Weil pairing of the received points and raises it to its own secret number:
   - Alice computes $e([b]P, [c]P)^a$,
   - Bob computes $e([c]P, [a]P)^b$,
   - Carol computes $e([a]P, [b]P)^c$.

Due to Equation (1) all participants have computed μ^{abc}.

This protocol has inspired alternative solutions which are based on lattices and therefore belong to the exponential cryptography [GGH13].

The three party Diffie-Hellman protocol can be broken by solving the DLP in the subgroup of E generated by P or by solving the DLP in the multiplicative group of $\mathbb{F}_{q^k}$. This is true for other applications of pairings but we stick to this example for simplicity.

2 Known attacks against pairings

2.1 Attacks on the curve side

Pollard rho In the three-party Diffie-Hellman protocol an attacker can compute the discrete logarithm of $[a]P$ and obtain the secret information a. The state-of-the-art algorithm to solve DLP in elliptic curves over prime fields is Pollard's

rho [Pol78] which has a cost of $O(\sqrt{r})$ operations. Hence, for a given security level one has to set $\log_2 r = 2s$ and therefore $\log_2 \#E(\mathbb{F}_q) \geq 2s$. Due to Hasse's theorem, q and $\#E(\mathbb{F}_q)$ have the same bit size up to an error of 3 bits, so we have $\log_2 q \geq \log_2 r = 2s$.

Faults on the twist curve Biehl, Meyer and Müller [BMM00] explained that, since some implementations of the scalar multiplication use only the x coordinate of the points on the elliptic curve $E : y^2 = x^3 + ax + b$, by error injection one can transfer the DLP from E to its twisted curve $E' : \epsilon y^2 = x^3 + ax + b$, where ϵ is a non-square of $\mathbb{F}_q$. As a counter-measure we require that the elliptic curves used in cryptography are twist-safe, i.e. that both $\#E(\mathbb{F}_q)$ and $2(q+1) - \#E(\mathbb{F}_q)$ have large prime factors.

Faults in Miller's algorithm Page and Vercauteren [PV06] studied the fault attacks which concern precisely the evaluation of the pairings and are independent on the protocol in which this primitive is used.

2.2 Attacks on the finite field side

In the three-party Diffie-Hellman protocol an attacker, who has access to the public information $[a]P$, can compute $\mu^a = e([a]P, P)$ using solely public information. By solving the DLP in the group generated by μ one can obtain the secret information a. Hence a safe pairing requires that the DLP in the multiplicative group of $\mathbb{F}_{q^k}$ is hard.

The best algorithms to solve DLP in finite fields inherited the main traits from Index Calculus [Adl79] and have a complexity inferior to any exponential function. A suitable notation to express their complexity is

$$L_Q(\alpha, c) = \exp((c + o(1))(\log Q)^{\alpha}(\log\log Q)^{1-\alpha}),$$

where Q is the cardinality of the target finite field and α and c are two constants such that $0 < \alpha < 1$. When the constant c is not important we simply write $L_Q(\alpha)$. By extension we use a similar notation when α is a function.

The state-of-the-art algorithms depend on the size of the characteristic p with respect to $Q = p^n$ (we switch notations from q^k to p^n to show that p is not necessarily prime). When $p = L_Q(l_p, c_p)$ we have the following complexities:

- $L_Q(\frac{1}{3}, \sqrt[3]{\frac{64}{9}})$ when the field has large characteristic, i.e. if $l_p > \frac{2}{3}$, [JLSV06];
- $L_Q(\frac{1}{3}, c)$ with $c \in [\sqrt[3]{\frac{48}{9}}, \sqrt[3]{\frac{96}{9}}]$ in the boundary case, i.e. if $l_p = \frac{2}{3}$; the constant $c = \sqrt[3]{\frac{48}{9}}$ is obtained when $c_p = 12^{\frac{1}{3}}$, [SS16];
- $L_Q(\frac{1}{3}, \sqrt[3]{\frac{48}{9}})$ when the field has medium characteristic, i.e. if $\frac{1}{3} < l_p < \frac{2}{3}$, and n has a factor of size $12^{-\frac{1}{3}}(\frac{\log Q}{\log\log Q})^{\frac{1}{3}}$; and $L_Q(\frac{1}{3}, \sqrt[3]{\frac{96}{9}})$ if n has no factor of the suitable size (e.g. if n is prime), [BGGM15b];

- $L_Q(\frac{1}{3}, c)$ with $c \in [\sqrt[3]{\frac{8}{9}}, \sqrt[3]{\frac{96}{9}}]$ when the field has a characteristic at the boundary between medium and small, i.e. if $l_p = \frac{1}{3}$; the complexity $c = \sqrt[3]{\frac{8}{9}}$ is obtained when $c_p = 3^{-\frac{1}{3}}$ [Jou13]; one has a better complexity in the case of Kummer extensions;
- $L_Q(l_p + o(1))$ when the field has small characteristic, i.e. $l_p < \frac{1}{3}$; the best complexity corresponds to $\exp(O(1)(\log\log Q)^2) = L_Q(o(1))$ when $p = (\log Q)^{O(1)}$, [BGJT14].

When the characteristic is non-small, i.e. $l_p \geq 1/3$, the best complexities are all obtained with the same algorithm, presented below.

Number field sieve (NFS) The main steps of NFS [JL03] are similar to those of Index calculus and the key ingredient is smoothness: an integer is B-smooth if all its prime factors are less than B.

Polynomial selection One selects two polynomials f and g with integer coefficients which, when seen as elements of $\mathbb{F}_p[x]$, have a common factor φ which has degree n and is irreducible. The performance of the algorithm depends strongly on the degrees of the two polynomials as well as on their norms, i.e. larges coefficient in absolute value.

Relation collection Given two polynomials $f = \sum_{i=0}^{\deg f} f_i x^i$ and $g = \sum_{i=0}^{\deg g} g_i x^i$ we collect all the pairs (a, b) of integers (or equivalently linear polynomials $a - bx \in \mathbb{Z}[x]$) such that $\max(|a|, |b|) \leq E$ for a parameter E, $\gcd(a, b) = 1$ and the two norms $N_f(a, b) = \sum_{i=0}^{\deg f} f_i a^i b^{\deg f - i}$ and $N_g(a, b) = \sum_{i=0}^{\deg g} g_i a^i b^{\deg g - i}$ are B-smooth. This stage is usually done using a technique called sieve.

Linear algebra For each pair (a, b) yielded by the sieve one can write a linear equation whose unknowns are in bijection with set of prime ideals of degree one in the number fields of f and g of norm less than B. The square matrix has less than B unknowns and less than $\log_2 p^n$ non-zero entry per row so that one can use sparse-matrix algorithms like Wiedemann [Wie86].

Individual logarithm The unknowns obtained after the linear algebra stage, called virtual logarithms, allow to compute any discrete logarithm. This stage takes a negligible amount of time compared to the other stages.

When p has a special form, e.g. a low Hamming weight, a variant of NFS has a better asymptotic complexity.

The special number field sieve (SNFS) Given an integer d, an integer p is d-SNFS if there exists a polynomial $P \in \mathbb{Z}[x]$ and an integer u so that $\|P\| \leq 50$ (or other absolute constant) and $p = P(u)$. Semaev [Sem02] proved that the DLP is easier in prime finite fields $\mathbb{F}_p$ when p is d-SNFS with $d = (\frac{9}{2})^{\frac{1}{3}}(\frac{\log p}{\log\log p})^{\frac{1}{3}}$. One doesn't have to change anything in the NFS algorithm except for the choice

of polynomials: $f = P(x)$ and $g = x - u$. In practice d is the value of $\deg f$ in the record computations using NFS and goes from 5 for fields of about 500 bits to 8 for fields of about 1200 bits. Experiments conducted with SNFS in the case of discrete logarithm [HT11] as well as of factorization [KBL14] confirm the efficiency of the algorithm for d-SNFS numbers with $d \geq 3$.

2.3 The LogJam attack

A simple remark about the algorithms of the Index Calculus family is that they have two types of input data: a group G and a generator g of G which are used in the costly stages of the algorithm, relation collection and linear algebra, and an element h of G which isn't used before the individual logarithm stage. An attacker can therefore perform the expensive computations which depend on G and g once for all and use then to compute many secrete keys by solving many instances of individual logarithm with respect to that group.

Adrian et al. [ABD+15] conducted real life attacks in this manner. They estimated that 82% of the scanned servers use the same group and therefore can be attacked with one stone. One can easily imagine a situation where this is unacceptable: 80 bits of security are enough to protect bits of one minute for a pay-TV channel whereas it might be unacceptable for the whole program.

Consequences. Whenever the security of a cryptosystem is measured using Index Calculus attacks, as NFS, one is vulnerable to the LogJam attack. In this case one might either use a stronger level of security or generate on-the-fly the group used in the cryptosystem. For example in the case of pairings one should be able to generate on-the-fly pairing-friendly curves. However in the case of hardware implementation of cryptosystems, where parameters have to be hard-coded, the only option is to use larger key sizes.

3 Recent progress of the NFS attack

The first estimations of security of pairings have been done at a time when NFS could only be used for prime fields, and one had to make the hypothesis that the DLP in the general case is as hard as in prime fields [Len01]. Since then the NFS was adapted to the case $\mathbb{F}_{p^n}$ of non-small characteristic and in some cases the complexity is smaller than in the prime case, as we present below.

3.1 New methods of polynomial selection

The first manner to go from $\mathbb{F}_p$ to $\mathbb{F}_{p^n}$ is to create new methods of polynomial selection whose result is a pair $(f, g) \in \mathbb{Z}[x]$ not necessarily irreducible which have a common irreducible factor φ in $\mathbb{F}_p[x]$.

For any pair (p, φ) formed of a prime p and a monic polynomial with integer coefficients φ which is irreducible in $\mathbb{F}_p[x]$ and any parameter $D \geq \deg \varphi$ one

defines the lattice

$$\mathcal{L}(p,\varphi,D) = \{(a_0,\ldots,a_D) \in \mathbb{Z}^{D+1} \mid \sum_{i=0}^{D} a_i x^i \in p\mathbb{Z}[x] + \varphi\mathbb{Z}[x])\}.$$

A naive method of polynomial selection would be to pick a random monic irreducible $\varphi \in \mathbb{F}_p[x]$ of degree n and to make f and g from the shortest two vectors b_1 and b_2 in an LLL-reduced basis of $\mathcal{L}(p,\varphi,D)$. By the Lenstra-Lenstra-Lovasz theorem [LLL82] we know that $\|b_1\|_2 \leq c_1 \mathrm{Vol}(\mathcal{L})^{\frac{1}{\dim \mathcal{L}}}$ where $c_1 = 2^{\frac{\dim \mathcal{L}}{4}}$. Heuristically we expect b_1 and b_2 to have no non-zero coordinates (random vectors) so that $\deg f = \deg g = D$ and $\|b_1\| \approx \|b_2\| \approx \mathrm{Vol}(\mathcal{L})^{\frac{1}{\dim \mathcal{L}}}$.

JLSV$_2$ In [JLSV06] Joux, Lercier, Smart and Vercauteren take φ of degree $n < D$ such that $\|\varphi\|_2 = 1 + c_1 \mathrm{Vol}(\mathcal{L})^{\frac{1}{\dim \mathcal{L}}}$. Then one can make f from the coordinates of the shortest vector of $\mathcal{L}(p,\varphi,D)$ and set $g = \varphi$. By the Lenstra-Lenstra-Lovasz theorem $\|f\| \leq c_1 \mathrm{Vol}(\mathcal{L})^{\frac{1}{\dim \mathcal{L}}} < \|g\|$ so the two polynomials are distinct. The advantage is that $\deg g = n$ which is smaller than D whereas $\deg f$, $\|f\|$ and $\|g\|$ are the same as in the naive method.

GJL In [JL03] Joux and Lercier proposed a method of polynomial for $\mathbb{F}_p$ which was generalized [Mat06],[BGGM15b] to $\mathbb{F}_{p^n}$ with $n > 1$ (generalized Joux Lercier). One takes f to be a polynomial of degree $D+1$ with $\|f\| = 1$ which has an irreducible factor $\varphi \in \mathbb{F}_p[x]$ of degree n, and then one makes g from the shortest vector of $\mathcal{L}(p,\varphi,D)$. The advantage in this case is that f has coefficients of size $O(1)$ instead of $c_1(p^n)^{\frac{1}{D+1}}$ for the small cost of increasing the degree of f from D to $D+1$.

JLSV$_1$ Also in [JLSV06] Joux, Lercier, Smart and Vercauteren proposed to take f equal to a polynomial of degree n which is irreducible in $\mathbb{F}_p[x]$ with $\|f\| \leq 1$ and to set $g = f + p$. An additional improvement, which doesn't change the asymptotic complexity, consists in selecting polynomials such that $\deg f = \deg g$ and $\|f\| = \|g\|$. We can obtain this if we apply the JLSV$_2$ method with $D = 2n$, when $\|f\| \approx \|g\| \approx c_1(p^n)^{\frac{1}{2n}} = c_1\sqrt{p}$. However, one can obtain polynomials of the same characteristics by reducing a lattice of dimension 2 instead of $2n$. Indeed one takes two polynomials $f_0, f_1 \in \mathbb{Z}[x]$ of degree n respectively $\leq n-1$ so that, for all integers a, $f_0 + af_1$ has degree n. Next one LLL-reduces the lattice generated by $M(a,p) = \begin{pmatrix} 0 & p \\ 1 & a \end{pmatrix}$ and obtains a vector (u,v) of norm $\leq 2^{\frac{1}{4}}\sqrt{p}$. Finally one sets $f = f_0 + af_1$ and set $g = vf_0 + uf_1$, which is a multiple of f in $\mathbb{F}_p[x]$.

Conjugation method This method, presented in [BGGM15b], is similar to JLSV$_1$. First we select f_0 and f_1 so that, for all integer a, $f_0 + af_1$ has degree n. Next we select m as small as possible so that $x^2 - m$ has a root $a \in \mathbb{Z}$ modulo p

and $f_0 + af_1$ is irreducible in $\mathbb{F}_p[x]$. We finish as in JLSV$_1$ by reducing $M(a,p)$ and setting $g = vf_0 + uf_1$.

At this point one would like to set $f = f_0 + \sqrt{m}f_1$ but this polynomial belongs to $\mathbb{Z}[\sqrt{m}][x]$ instead of $\mathbb{Z}[x]$. We overcome this difficulty by setting $f = (f_0 + \sqrt{m}f_1)(f_0 - \sqrt{m}f_1) = f_0^2 - mf_1^2$ which has integer coefficients and is a multiple of g in $\mathbb{F}_p[x]$.

Methods for composite n Sarkar and Singh [SS16] proposed a method which improves the asymptotic complexity of NFS when $p = L_Q(2/3, c_p)$ with $c_p \in [1.12, 1.45] \bigcup [3.15, 20.91]$. The authors made a precise estimation of efficiency in the case of finite fields of cryptographic sizes $n = 4$ and $n = 6$.

Practical efficiency of the new methods The new methods have been tested in practice and one concluded that the DLP in non-prime finite fields can be easier than in the prime case. In Table 1 we compare the cases $n = 2$ and $n = 3$ using the Conjugation method (Conj) to the prime case ($n = 1$). For this we converted the computation time into GIPS years (1GIPS year = the number of instructions done in one year by a CPU core of 1GHz) and made the convention that 1 GPU hour = 10 CPU hours.

bit size of p^n	160 dd ($\approx$ 532 bits)	180 dd ($\approx$ 600 bits)
n=1	55.5 [Kle07]	260 [BGI+14]
n=2 (Conj)	0.5 [BGGM14]	1 [BGGM15b]
n=3 (Conj)	34 [BGGM15a]	46 [GGM16]

Table 1: Time of discrete logarithms computations in $\mathbb{F}_{p^n}$ measured in GIPS years.

3.2 The tower number field sieve

A second method to go from $\mathbb{F}_p$ to $\mathbb{F}_{p^n}$ with $n > 1$ has been proposed by Schirokauer [Sch00] and revised in [BGK15]. One selects $h \in \mathbb{Z}[x]$ of degree n which is irreducible in $\mathbb{F}_p[x]$ and call ι a root of h in its number field. Then one selects f and g in $\mathbb{Z}[x]$ which have a common root in $\mathbb{F}_p$ using one of the methods for $\mathbb{F}_p$ and calls α_f (resp. α_g) a root of f (resp. g) in its number field and set $K_f = \mathbb{Q}(\iota, \alpha_f)$ (resp. $K_g = \mathbb{Q}(\iota, \alpha_g)$) and compute θ_f (resp. θ_g) a primitive element of K_f (resp. K_g).

One sets the parameters E, B and d at the same value as when computing discrete logarithms in a prime field of same bit size as $\mathbb{F}_{p^n}$. The factor base is formed of the prime ideals of K_f and K_g whose norm is less than B and whose inertia degree over $\mathbb{Q}(\iota)$ is one, together with all the prime ideals dividing the leading coefficients of f and g. The algorithm continues as follows.

1. Enumerate all pairs $a, b \in \mathbb{Z}[t]$ of degree $n-1$ with $\|a\|, \|b\| \leq E^{\frac{1}{n}}$ and collect those such that $\mathrm{Res}_t(F(a,b), h(t))$ and $\mathrm{Res}_t(G(a,b), h(t))$ are B-smooth.

2. Consider each element $a(\iota) + \alpha_f b(\iota)$ (resp. $a(\iota) + \alpha_g b(\iota)$) and compute the corresponding linear equations, as in the case of the classical version of NFS. Then solve the linear system to obtain the virtual logarithms of the factor base.
3. Compute the desired discrete logarithm in a similar manner to the classical case.

The practical efficiency of the TNFS has not been tested. Indeed, the relation collection consists of sieving on pairs $(a, b) \in \mathbb{Z}[t]$ of degree less than n which is equivalent to sieving on pairs of $2n$-tuples of integers. Several teams [Zaj10],[HAKT15],[GGV16] made experiments in the case of 3-tuples and concluded that this does not represent a major practical obstacle. This might be a starting point for future experiments in the case of 4-tuples so that TNFS in $\mathbb{F}_{p^2}$ can be tested.

3.3 The extended tower number field sieve

The extended number field sieve (exTNFS), presented in [KB16], consists in combining the two ideas of the previous sections: new methods of polynomial selection and tower number fields. One writes $n = \eta\kappa$ with $\eta, \kappa \in \mathbb{Z}$ but not necessarily different from 1 and n and selects polynomials:

1. f and g as in Section 3.1 with κ instead of n;
2. h as in Section 3.2 with η instead of n.

When $\eta = 1$ we obtain the variant of NFS in Section 3.1, when $\eta = n$ we obtain TNFS (Section 3.2), but when n is composite and η is a proper factor of n we obtain a new algorithm. When $\gcd(\eta, \kappa) \neq 1$ one has to use a special method of polynomial selection which is due to Jeong and Kim [JK16]. The advantage of exTNFS is that, in a similar manner in which in TNFS one has the same size of norms as in classical NFS, in exTNFS one has the same size of the norms when attacking $\mathbb{F}_{p^{\eta\kappa}}$ as when attacking $\mathbb{F}_{P^\kappa}$ for a prime P of the same bit size as p^η.

The case of general primes In order to analyze the efficiency of exTNFS we estimate the bit size of the norms product. Using Lemma 1 in [KB16] we find that, when the Conjugation method is used to select f and g, the two upper bound on the norms bit size is:

$$\text{norms bit size(exTNFS-Conj)} \leq 3\kappa \log_2 E + \frac{1}{2\kappa} \log_2 Q + o(1). \tag{3}$$

where $o(1)$ is a negligible term when $\log_2 Q$ goes to infinity. The $o(1)$ term is indeed negligible in cryptographic examples, e.g. Example 1 in [KB16]. Hence exTNFS has the same efficiency as NFS with the difference that now we can tune the parameter κ and make it equal to any factor of n.

The right hand member of Equation (3) has its minimum when $\kappa \approx \sqrt{\frac{\log_2 Q}{6 \log_2 E}}$. Although the bit size of the parameter E depends on the size of the norms it

doesn't vary of more than a factor 2 among variants of NFS when one attacks the same size of finite fields. In [KDL+16] one has $\log_2 Q = 768$ and $\log_2 E \approx 43$ so that the optimal value of $\kappa \approx 1.72$. We conclude that if one selects f and g using the Conjugation method then for target fields of approximatively 1000 bits with $n \leq 24$ composite the best options are $\kappa = 2$ if n is even and $\kappa = 3$ if n is odd. This would allow to obtain similar practical results as in Table 1.

The case of primes of special form The exTNFS variant for SNFS numbers, abbreviated SexTNFS, consists in writing $n = \eta\kappa$ for two integers κ and η not necessarily different from 1 and n, in selecting h as in Section 3.2 with η instead of n and in selecting f and g using the Joux-Pierrot method [JP13], that we describe below, with κ instead of n.

One selects a monic polynomial $S \in \mathbb{Z}[x]$ of degree n such that $f = P(S(x))$ is irreducible in $\mathbb{F}_p[x]$ and then sets $g = S(x) - u$. The method is correct due to the following equation:

$$f(x) - p = P(S(x)) - P(u) \equiv 0 \bmod (S(x) - u) \text{ in } \mathbb{F}_p[x].$$

Once again we evaluate the practical efficiency using the estimation of the bit size of the norms product, which from [KB16, Section 5.2] is:

$$\text{norms bit size(SexTNFS)} \leq (d+1)\kappa \log_2 E + \frac{1}{\kappa d} \log_2 Q + o(1),$$

where $o(1)$ is negligible when Q goes to infinity. The advantage of SexTNFS is that we have the possibility to set κ equal to any divisors of n.

4 Pairings families and their security

In the light of the recent progress, a perfect pairing family needs to contain a large number of curves for each security level that can be rapidly generated. Each curve of a perfect family has an embedding degree k which can be set as desired to any prime of desired size. The characteristic p is large and is not d-SNFS with $d \geq 3$. Finally for efficiency reasons the parameter r has the same bit size as q.

Freeman, Scott and Teske [FST10] made a taxonomy of known pairing-friendly families of elliptic curves. Given a bit size and an embedding degree k, most of them are constructed in two steps:

i) one selects a prime power q of prescribed bit size and an integer t so that any elliptic curve over $\mathbb{F}_q$ of trace t has embedding degree k and its cardinality has a large prime factor r;
ii) one uses the CM method [Mor91],[AM93], which, given a prime power q and an integer t, allows to construct elliptic curves over $\mathbb{F}_q$ of trace t.

The CM method has complexity $O(D^{1+\epsilon})$ where D is the unique integer so that $(4q - t^2)/D$ is a perfect square. This imposes that we fix D in advance: it will be either small or will have common factors with q. By definition $\#E(\mathbb{F}_q) = q + 1 - t$

so we ask the existence of a prime r so that $q+1-t \equiv 0 \bmod r$. Finally, the property that k is the embedding degree of the curve is equivalent to $\Phi_k(q) \equiv 0 \pmod r$. We summarize the conditions on the output of the first step as follows:

CM-1. $\Phi_k(t-1) \equiv 0 \pmod r$
CM-2. $q+1-t \equiv 0 \pmod r$
CM-3. $\exists y,\ 4q = Dy^2 + t^2$

4.1 Supersingular curves

When $k = 2$ there is a value of D for which the system is easy to solve. Indeed we set $t = 0$ so that we have $\Phi_2(t-1) = 0$ and therefore the first equation is satisfied independently on r. In Equation CM-3. we take $D = q$ and $y = 2$ so that there is no condition on q. Finally Equation CM-2. states that $q+1$ has a prime factor r which is easy to fulfill by enumerating primes q. Bröker [Brö06] presented the CM method in the case $D = q$, which is fast although D is large.

A natural question is whether this method can be extended to other values of k. The answer is given by the following classical result.

Proposition 1. *If $p \geq 5$ is a prime then any supersingular elliptic curve over $\mathbb{F}_p$ has embedding degree $k = 2$.*

Proof. By the definition of supersingular curves we have $\gcd(t, p) \neq 1$ so p divides t and therefore $t = 0$ or $|t| \geq p$. By Hasse's theorem $|t| \leq 2\sqrt{p}$ which is less than p and therefore $t = 0$. Then $q \equiv t - 1 \equiv -1 \pmod r$ and $q^2 \equiv 1 \pmod r$ which shows that $k = \mathrm{ord}_r(q) = 2$.

Drawback Due to the quasi-polynomial algorithm the cases $p = 2$ and $p = 3$ are forbidden. When $p \geq 5$ the embedding degree $k = 2$ is fixed to a value which is far from the optimal value in Table **??** and has made the object of recent computation records which were faster than the prime case.

4.2 Pinch-Cocks [CP01]

One starts by replacing Equation CM-2. with

$$\text{CM-2}'. \qquad Dy^2 + (t-2)^2 \equiv 0 \pmod r$$

so that we obtain an equivalent system. Then we select r so that $r \equiv 1 \bmod k$ and $(\frac{-D}{r}) = 1$. Then Equation CM-2'. is factorized into

$$(\sqrt{-D}y + (t-2))(\sqrt{-D}y - (t-2)) \equiv 0 \pmod r.$$

The choice of r allows to set t equal to a root of the polynomial $\Phi_k(X-1) \in \mathbb{F}_r[X]$. The same choice allows to solve this Equation CM-2'. for y: $y = (t-2)/\sqrt{-D} \pmod r$. Finally q is set to $(Dy^2+t^2)/4$. Heuristically this is integer in a constant proportion of the cases and has the same probability to be prime as a random integer of the same size, i.e. one succeeds on average after $O(\log q)$ trials.

Drawback. With high probability the integer y has the same bit size as r so that $\log_2 q \approx 2\log_2 r$ which affects the efficiency of pairings.

4.3 Dupont-Enge-Morain [DEM05]

Once again we start by replacing Equation CM-2. by Equation CM-2′. Then we see Equations CM-1 and CM-2′ as a system which has to be solved with $y, t \in \mathbb{F}_r$:

$$\begin{cases} \Phi_k(t-1) = 0 \\ Dy^2 + (t-2)^2 = 0. \end{cases}$$

We solve the system (for a given D and bit size b of q) as follows:

1: $R(y) \leftarrow \mathrm{Res}_t(\Phi_k(t-2), Dy^2 + (t-2)^2)$;
2: **for** $y \leq 2^{\frac{b}{2\varphi(k)}}$ **do**
3: $r \leftarrow$ the largest prime factor of $R(y)$
4: $t = 2 + \sqrt{-Dy^2}$ (if it exists)
5: $q \leftarrow q = (Dy^2 + t^2)/4$
6: $q' \leftarrow q + 1 + t$ (cardinality of the twisted curve to be tested)
7: **if** q and q' are integer primes and $\log_2 r \geq b/2$ **then return** y
8: **end if**
9: **end for**

For example we ran the algorithm for the bit size $b = 256$, embedding degree $k = 16$ and the parameter $D = 3$. The output list was: $y \in \{39193, 61815\}$.

Drawback. The total number of curves which can be constructed for cryptographic sizes is very small if we restrict to twist-safe curves so that this family is vulnerable to the LogJam attack.

4.4 Sparse families (e.g. MNT [MNT01])

The following construction is possible for all integers k so that $\varphi(k) = 2$, i.e. $k = 3, 4$ and 6, but for simplicity we present only the case $k = 3$. We set $r = \Phi_k(t-1)$ so that Equation CM-1 is satisfied. Next we set $q = r + t - 1$, which satisfies CM-2.. The method was generalized by Freeman when $\varphi(k)$ but cannot be generalized further.

Proposition 2. *If $\varphi(k) > 4$ then the system CM-$1, 2, 3$ has a finite set of solutions.*

Proof. When we set $r = \Phi_k(t-1)$ Equation CM-3. becomes

$$y^2 = f(t) \text{ where } f(t) = \frac{1}{D}(4q - t^2) = \frac{1}{D}(4(\Phi_k(t-1) + t - 1) - t^2).$$

By the Riemann-Hurwitz formula the genus of the curve is $\lfloor \frac{\deg f - 1}{2} \rfloor = \lfloor \frac{\varphi(k)-1}{2} \rfloor \geq 2$. By Faltings' theorem the equation has a finitely many solutions in $\mathbb{Q}$.

The integer solutions obtained when setting t equal to a linear polynomial in an additional variable are a subset of the rational solutions, so we have a finite number in total.

Drawback The embedding degree k has a very small set of possibilities all of which are divisible by 2 or 3.

4.5 Complete families (e.g BN [BN05])

Once again we replace Equation CM-2 by CM-2′. Then we set r equal to a polynomial $r(x)$ whose number field contains $\mathbb{Q}(\sqrt{-D}, \zeta_k)$ for a kth root of unity ζ_k. This translates into

1. Φ_k is totally split modulo $r(x)$;
2. $x^2 + D$ is totally split modulo $r(x)$.

Next we take t to be a polynomial $t(x)$ so that $\Phi_k(t(x)) \equiv 0 \bmod r(x)$. Since Equation CM-2′ factors we can set $y(x) = t(x) \cdot \frac{t(x)}{\sqrt{-D}}$ where $\frac{1}{\sqrt{-D}}$ is a polynomial $z(x)$ in $\mathbb{Q}(x]$ so that $Dz^2 + 1 \equiv 0 \bmod r(x)$. Finally set $q(x) = \frac{1}{4}(Dy(x)^2 + t(x)^2)$. The advantage of this method is that pairing-friendly curves can be generated on the fly by evaluating r and q at integer values x.

Drawback The primes constructed by this method are $2\varphi(k)$-SNFS and therefore the NFS attacks have a smaller asymptotic complexity.

4.6 Menezes-Köblitz [KM05]

Not all the pairing constructions are obtained using the CM method. Menezes and Köblitz proposed a family which is not affected by the recent progress: p is not d-SNFS with $d \geq 3$ so that the SNFS attack has no consequences and $k = 1$ so that the security on the finite field side is the same as that of DSA.

Drawback The embedding degree k cannot be tuned as in Table **??**.

5 Conclusion

We have identified a list of properties that a perfect pairing family should have and, by a thorough examination, concluded that in the present state of the art there is no perfect pairing family. In particular there is no clear champion because the Barreto-Naehrig family, long believed to be perfect for 128 bits of security, has a characteristic of a special form and is target to the SNFS attack.

Pairings are subject to two contradictory trends. On the one hand they require more time before being standardized because no perfect family has been proposed. On the other hand, time is running against pairings as they are subject to the NFS attack and therefore belong to the sub-exponential cryptography as RSA and DSA whereas there exist alternative primitives which are based on lattices and belong to the exponential cryptography.

References

ABD+15. David Adrian, Karthikeyan Bhargavan, Zakir Durumeric, Pierrick Gaudry, Matthew Green, J. Alex Halderman, Nadia Heninger, Drew Springall, Emmanuel Thomé, Luke Valenta, Benjamin VanderSloot, Eric Wustrow, Santiago Zanella-Béguelin, and Paul Zimmermann. Imperfect forward secrecy: How diffie-hellman fails in practice. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security–CCS '15*, pages 5–17, New York, NY, USA, 2015. ACM.

Adl79. L. M. Adleman. A subexponential algorithm for the discrete logarithm problem with applications to cryptography. In *20th Annual Symposium on Foundations of Computer Science*, pages 55–60. IEEE, 1979.

AM93. A Oliver L Atkin and François Morain. Elliptic curves and primality proving. *Mathematics of computation*, 61(203):29–68, 1993.

BGGM14. R. Barbulescu, P. Gaudry, A. Guillevic, and F. Morain. Discrete logarithms in GF(p^2) — 160 digits, 2014. Announcement available at the NMBRTHRY archives, item 004706.

BGGM15a. R. Barbulescu, P. Gaudry, A. Guillevic, and F. Morain. New record in $\mathbb{F}_{p^3}$, 2015. Available online at https://webusers.imj-prg.fr/~razvan.barbaud/p3dd52.pdf.

BGGM15b. 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.

BGI+14. C. Bouvier, P. Gaudry, L. Imbert, H. Jeljeli, and E. Thom. Discrete logarithms in GF(p) — 180 digits, 2014. Announcement available at the NMBRTHRY archives, item 004703.

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.

BMM00. Ingrid Biehl, Bernd Meyer, and Volker Müller. Differential fault attacks on elliptic curve cryptosystems. In *Advances in Cryptology–CRYPTO 2000*, volume 1880 of *Lecture Notes in Comput. Sci.*, pages 131–146. Springer, 2000.

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.

Brö06. R. Bröker. *Constructing elliptic curves of prescribed order*. PhD thesis, Leiden University, 2006. Available at: http://www.math.leidenuniv.nl/~reinier/thesis.pdf.

CP01. Clifford Cocks and RGE Pinch. Identity-based cryptosystems based on the weil pairing. *Unpublished manuscript*, 170, 2001.

DEM05. Régis Dupont, Andreas Enge, and François Morain. Building curves with arbitrary small mov degree over finite prime fields. *Journal of Cryptology*, 18(2):79–89, 2005.

FR94. Gerhard Frey and Hans-Georg Rück. A remark concerning -divisibility and the discrete logarithm in the divisor class group of curves. *Mathematics of computation*, 62(206):865–874, 1994.

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

GGH13. Sanjam Garg, Craig Gentry, and Shai Halevi. Candidate multilinear maps from ideal lattices. In *Advances in Cryptology-EUROCRYPT 2013*, pages 1–17. Springer, 2013.

GGM16. P. Gaudry, A. Guillevic, and F. Morain. Discrete logarithm record in GF(p^3) of 592 bits (180 decimal digits), 2016. Announcement available at the NMBRTHRY archives, item 004706.

GGV16. Pierrick Gaudry, Laurent Grémy, and Marion Videau. Collecting relations for the number field sieve in GF(p^6), 2016. Accepted for publication at ANTS-XII, Kaiserslautern.

HAKT15. Kenichiro Hayasaka, Kazumaro Aoki, Tetsutaro Kobayashi, and Tsuyoshi Takagi. A construction of 3-dimensional lattice sieve for number field sieve over GF(p^n). Cryptology ePrint Archive, Report 2015/1179, 2015. http://eprint.iacr.org/2014/300.

HT11. Kenichiro Hayasaka and Tsuyoshi Takagi. An experiment of number field sieve over GF(p) of low hamming weight characteristic. In *Coding and Cryptology–2011*, volume 6639 of *Lecture Notes in Comput. Sci.*, pages 191–200. Springer, 2011.

JK16. Jinhyuck Jeong and Taechan Kim. Extended tower number field sieve with application to finite fields of arbitrary composite extension degree. Cryptology ePrint Archive, Report 2016/526, 2016. http://eprint.iacr.org/2016/526.

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

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.

Jou00. Antoine Joux. A one round protocol for tripartite diffie–hellman. In *International Algorithmic Number Theory Symposium–ANTS V*, volume 1838 of *Lecture Notes in Comput. Sci.*, pages 385–393. Springer, 2000.

Jou13. Antoine 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.

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.

KB16. T. Kim and R. Barbulescu. Extended tower number field sieve: A new complexity for medium prime case. In *Advances in Cryptology – CRYPTO 2016 (part 1)*, volume 9815 of *Lecture Notes in Comput. Sci.*, pages 543–571, 2016.

KBL14. Thorsten Kleinjung, Joppe W Bos, and Arjen K Lenstra. Mersenne factorization factory. In *Advances in Cryptology–ASIACRYPT 2014*, 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.

Kle07. T. Kleinjung. Discrete logarithms in GF(p) — 160 digits, 2007. Announcement available at the NMBRTHRY archives, item 003269.

KM05. Neal Koblitz and Alfred Menezes. Pairing-based cryptography at high security levels. In *Cryptography and Coding 2005*, volume 3796 of *Lecture Notes in Comput. Sci.*, 2005.

Len01. 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.

LLL82. Arjen Klaas Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.

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.

MNT01. Atsuko Miyaji, Masaki Nakabayashi, and Shunzou Takano. New explicit conditions of elliptic curve traces for fr-reduction. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 84(5):1234–1243, 2001.

Mor91. François Morain. Building cyclic elliptic curves modulo large primes. In *Advances in Cryptology - EUROCRYPT '91*, volume 547 of *Lecture Notes in Comput. Sci.*, pages 328–336, 1991.

MOV93. Alfred J Menezes, Tatsuaki Okamoto, and Scott A Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on information Theory*, 39(5):1639–1646, 1993.

Pol78. John Pollard. Monte carlo methods for index computation (mod p). *Mathematics of Computation*, 32(143):pp. 918–924, 1978.

PV06. Dan Page and Frederik Vercauteren. A fault attack on pairing-based cryptography. *IEEE Transactions on Computers*, 55(9):1075–1080, 2006.

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

Sem02. I. Semaev. Special prime numbers and discrete logs in finite prime fields. *Mathematics of Computation*, 71(237):363–377, 2002.

Sil07. Joseph H Silverman. *The arithmetic of dynamical systems*, volume 241. Springer Science & Business Media, 2007.

SS16. Palash Sarkar and Shashank Singh. New complexity trade-offs for the (multiple) number field sieve algorithm in non-prime fields. In *Advances in Cryptology – EUROCRYPT 2016*, volume 9665 of *Lecture Notes in Comput. Sci.*, pages 429–458, Berlin, Heidelberg, 2016. Springer.

Wei40. André Weil. Sur les fonctions algébriquesa corps de constantes fini. *CR Acad. Sci. Paris*, 210(1940):592–594, 1940.

Wie86. D. Wiedemann. Solving sparse linear equations over finite fields. *IEEE Trans. Inform. Theory*, 32(1):54–62, 1986.

Zaj10. Pavol Zajac. On the use of the lattice sieve in the 3d nfs. *Tatra Mountains Mathematical Publications*, 45(1):161–172, 2010.