PAIRING-BASED CRYPTOGRAPHY AT HIGH SECURITY LEVELS

NEAL KOBLITZ AND ALFRED MENEZES

Abstract. In recent years cryptographic protocols based on the Weil and Tate pairings on elliptic curves have attracted much attention. A notable success in this area was the elegant solution by Boneh and Franklin [7] of the problem of efficient identity-based encryption. At the same time, the security standards for public key cryptosystems are expected to increase, so that in the future they will be capable of providing security equivalent to 128-, 192-, or 256-bit AES keys. In this paper we examine the implications of heightened security needs for pairing-based cryptosystems. We first describe three different reasons why high-security users might have concerns about the long-term viability of these systems. However, in our view none of the risks inherent in pairing-based systems are sufficiently serious to warrant pulling them from the shelves.

We next discuss two families of elliptic curves E for use in pairing-based cryptosystems. The first has the property that the pairing takes values in the prime field $\mathbb{F}_p$ over which the curve is defined; the second family consists of supersingular curves with embedding degree $k = 2$. Finally, we examine the efficiency of the Weil pairing as opposed to the Tate pairing and compare a range of choices of embedding degree k, including $k = 1$ and $k = 24$.

1. Introduction

Let E be the elliptic curve

$$y^2 = x^3 + ax + b \tag{1}$$

defined over a finite field $\mathbb{F}_q$, and let P be a basepoint having prime order n dividing $\#E(\mathbb{F}_q)$, where we assume that n does not divide q. Let k be the multiplicative order of q modulo n; in other words, it is the smallest positive k such that $n \mid q^k - 1$. The number k, which is called the *embedding degree*, has been of interest to cryptographers ever since it was shown in [35] how to use the Weil pairing to transfer the discrete log problem in the group $\langle P \rangle \subset E(\mathbb{F}_q)$ to the discrete log problem in the finite field $\mathbb{F}_{q^k}$.

In recent years, the Tate pairing (introduced to cryptographers by Frey-Rück [16]) and the Weil pairing have been used to construct a number of different cryptosystems. These systems were the first elliptic curve cryptosystems not constructed by analogy with earlier versions that used the

Date: March 14, 2005; updated April 7, 2005; May 1, 2005; and May 11, 2005.

Key words and phrases. public key cryptography, elliptic curve, Weil pairing, Tate pairing.

multiplicative group of a finite field. Rather, pairing-based cryptosystems use properties of elliptic curves in an essential way, and so they cannot be constructed in simpler settings (such as finite fields or the integers modulo N). In the next section we shall describe a particularly elegant example of such a cryptosystem, namely, the solution of Boneh and Franklin [7] of the problem of efficient identity-based encryption.

Meanwhile, it is becoming increasingly apparent that we are approaching a transitional moment in the deployment of cryptography. Calls are going out for heightened security standards for public key cryptosystems, so that in the future they will be capable of providing security equivalent to 128-, 192-, or 256-bit AES keys. In this paper we examine the implications for pairing-based elliptic curve cryptography of a move to higher levels of security.

Our first purpose is to describe three general questions about efficiency and security that arise. These concerns are not new to people working in the area, but they are rarely mentioned explicitly in print. By calling the reader's attention to these issues we have no intention of sounding alarmist or of discouraging deployment of these systems. On the contrary, in our view none of the considerations discussed below are sufficiently worrisome to justify abandoning pairing-based cryptography.

Our second purpose is to describe two very simple families of elliptic curves defined over a prime field $\mathbb{F}_p$ with embedding degrees $k = 1$ and $k = 2$, respectively, that could be used in pairing-based cryptosystems. The main advantage of these families is the flexibility one has in choosing the two most important parameters of the system — the field size p and the prime order n of the basepoint $P \in E(\mathbb{F}_p)$. One can easily get n and p both to have optimal bitlengths and at the same time to be Solinas primes [49] (that is, the sum or difference of a small number of powers of 2). In earlier papers on parameter selection for pairing-based systems we have not found any discussion of the advantages and disadvantages of low-Hamming-weight p.

On the negative side, when $k = 1$ one does not have any of the speedups that come from working in a subfield at various places in the pairing computations. When $k = 2$ our curves are supersingular, and so one must anticipate some resistance to their use because of the traditional stigma attached to the word "supersingular" by implementers of elliptic curve cryptography. Moreover, in both cases the use of a Solinas prime p could possibly enable an attacker to use a special form of the number field sieve. It remains to be seen whether the increased field sizes which would then be necessary offset the efficiency advantage provided by the use of such a prime.

Our third purpose is to compare different choices of k, ranging from 1 to 24, for different security levels. Our comparisons are simple but realistic, and incorporate most of the important speedups that have been discovered so far. Although much depends on the implementation details, it appears that for nonsupersingular curves the choice $k = 2$ that is recommended by some authors [46] is probably less efficient than higher values of k. We also find that for very high security levels, such as 192 or 256 bits, the Weil pairing computation is sometimes faster than the Tate pairing.

Earlier work in this area has focused on providing 80 bits of security, which is sufficient for most current applications. In contrast, we are particularly interested in how the choice of parameters will be affected by the move to the higher AES standard of 128, 192, or 256 bits of security that is anticipated in the coming years.

2. Identity-Based Encryption

One of the most important applications of the Weil (or Tate) pairing is to identity-based encryption [7]. Let's recall how the basic version of the Boneh–Franklin scheme works. Suppose that E over $\mathbb{F}_q$ is an elliptic curve on which (a) the Diffie–Hellman problem is intractable and (b) the Weil pairing $\widehat{e}(P, Q) \in \mathbb{F}_{q^k}$ can be efficiently computed. (For an excellent treatment of the Weil and Tate pairings, see [17].) Here P and Q are $\mathbb{F}_{q^k}$-points of prime order n, where $n \mid \#E(\mathbb{F}_q)$, and the embedding degree k (for which $E(\mathbb{F}_{q^k})$ contains all n^2 points of order n) must be small.

Bob wants to send Alice a message m, which we suppose is an element of $\mathbb{F}_{q^k}$, and he wants to do this using nothing other than her identity, which we suppose is hashed and then embedded in some way as a point I_A of order n in $E(\mathbb{F}_q)$. In addition to the field $\mathbb{F}_q$ and the curve E, the system-wide parameters include a basepoint P of order n in $E(\mathbb{F}_{q^k})$ and another point $K \in \langle P \rangle$ that is the public key of the Trusted Authority. The TA's secret key is the integer s that it used to generate the key $K = sP$.

To send the message m, Bob first chooses a random r and computes the point rP and the pairing $\widehat{e}(K, I_A)^r = \widehat{e}(rK, I_A)$. He sends Alice both the point rP and the field element $u = m + \widehat{e}(rK, I_A)$. In order to decrypt the message, Alice must get her decryption key D_A from the Trusted Authority; this is the point $D_A = sI_A \in E(\mathbb{F}_q)$ that the TA computes using its secret key s. Finally, Alice can now decrypt by subtracting $\widehat{e}(rP, D_A)$ from u (note that, by bilinearity, we have $\widehat{e}(rP, D_A) = \widehat{e}(rK, I_A)$).

3. Clouds on the Horizon?

The first reservation that a high-security user might have about pairing-based systems relates to efficiency. A necessary condition for security of any pairing-based protocol is that discrete logarithms cannot be feasibly found in the finite field $\mathbb{F}_{q^k}$. In practice, q is either a prime or a power of 2 or 3, in which case the number field sieve [19, 43] or function field sieve [14, 1, 44] will find a discrete log in time of order $L(1/3)$; this means that the bitlength of q^k must be comparable to that of an RSA modulus offering the same security. In both cases the bitlength should be, for example, at least 15360 to provide security equivalent to a 256-bit AES key [29, 40].[1]

As in the case of RSA, the loss of efficiency compared to non-pairing-based elliptic curve cryptography (ECC) increases steeply as the security level grows. Unlike RSA, pairing-based systems can achieve certain cryptographic objectives — notably, identity-based encryption — that no one has been able to achieve using ordinary ECC. So one has to ask how badly one wants the features that only pairing-based methods can provide. As the security requirements increase, the price one has to pay for the extra functionality will increase sharply.

It should be noted that in certain applications bandwidth can be a reason for using pairing-based systems (see, for example, [5, 6, 8]). We shall not consider bandwidth in this paper, except briefly in §4.1 for Boneh–Lynn–Shacham signatures.

The other two concerns about pairing-based systems are more theoretical, and both relate to security. In the first place, in most pairing-based protocols security depends upon the assumed intractability of the following problem, which Boneh and Franklin [7] called the Bilinear Diffie–Hellman Problem (BDHP): Given $P, rP, sP, Q \in E(\mathbb{F}_{q^k})$ such that $\zeta = \widehat{e}(P,Q) \neq 1$, compute ζ^{rs}.

The BDHP is a new problem that has not been widely studied. It is closely related to the Diffie–Hellman Problem (DHP) in the elliptic curve group $E(\mathbb{F}_{q^k})$, which is the problem, given P, rP, and sP, of computing rsP. Since $\zeta^{rs} = \widehat{e}(rsP, Q)$, it follows that if one has an algorithm for the DHP on the curve, one can immediately solve the BDHP as well. But the converse is not known, and it is possible that the BDHP is an easier problem than the DHP on the curve.

In the early discussions of discrete-log-based cryptosystems it was a source of concern that security depended on the presumed intractability of the Diffie–Hellman Problem rather than the more natural and more extensively studied Discrete Log Problem (DLP). That is why cryptographers were very pleased when a series of papers by den Boer, Maurer, Wolf, Boneh, Lipton and others (see [33] for a survey) developed strong evidence for the equivalence of the Diffie–Hellman and Discrete Log Problems on elliptic curves. But unfortunately, no such evidence has been found for hardness of the Bilinear Diffie–Hellman Problem. Of course, no one knows of any way to solve the BDHP except by finding discrete logs, so perhaps it is reasonable to proceed as if the BDHP is equivalent to the DHP and the DLP on elliptic curves — despite the absence of theoretical results supporting such a supposition.

The BDHP is also closely related to the Diffie–Hellman Problem in the finite field $\mathbb{F}_{q^k}$, and any algorithm for the DHP in the field will immediately

[1] For fields of small characteristic, q^k should be significantly larger than for q a prime. In [29] the bitlengths 4700, 12300, and 24800 are recommended for security levels 128, 192, and 256 bits, respectively.

enable us to solve the BDHP too. But it is possible that the BDHP is strictly easier than the DHP in the field. In the DHP we are given only the values ζ, ζ^r, and ζ^s, whereas in the BDHP the input also includes the inverse images of these n-th roots of unity under the Menezes-Okamoto-Vanstone [35] embedding from $\langle P\rangle \subset E(\mathbb{F}_{q^k})$ to the finite field given by $X \mapsto \widehat{e}(X,Q)$ for $X \in \langle P\rangle$.

This brings us to the third major concern with pairing-based cryptosystems, namely, Verheul's theorem [51].

Even if one is willing to suppose that the Bilinear Diffie–Hellman Problem on a low-embedding-degree curve is equivalent to the DHP and the DLP on the curve, in practice one really considers the DHP and DLP in the multiplicative group of a finite field, because it is there that the problem has been extensively studied and index-calculus algorithms with carefully analyzed running times have been developed. Using the MOV embedding, the DHP and DLP on the low-embedding-degree curve reduce to the corresponding problems in the finite field. At first it seems that it would be nice to have reductions in the other direction as well. That is, a homomorphism in the opposite direction to the MOV embedding would show that the problems on the curve and in the field are provably equivalent. Indeed, in special cases construction of such a homomorphism was posed as an open problem in [28] and [36]. However, in [51] Verheul dashed anyone's hopes of ever strengthening one's confidence in the security of pairing-based systems by constructing such a reduction.

Verheul proved the following striking result. Let μ_n denote the n-th roots of unity in $\mathbb{F}_{p^6}$, where $n|(p^2-p+1)$, and hence μ_n is not contained in a proper subfield; this is called an XTR group [30]. Suppose that an efficiently computable nontrivial homomorphism is found from μ_n to $\langle P\rangle \subset E(\mathbb{F}_{p^2})$, where E is an elliptic curve defined over $\mathbb{F}_{p^2}$ with $\#E(\mathbb{F}_{p^2}) = p^2-p+1$. Here we are assuming, as before, that P is a point of prime order n. Then Verheul's theorem states that the DHP is efficiently solvable in both μ_n and $\langle P\rangle$.

A generalization of Verheul's theorem, which was conjectured but not proved in [51], would give the same result whenever a group $\mu_n \subset \mathbb{F}_{q^k}$ can be efficiently mapped to a supersingular curve $E(\mathbb{F}_q)$. (Note that $q = p^2$ and $k = 3$ for the XTR group.) It is this generalized version that prompted Verheul to suggest that his results "provide evidence that the multiplicative group of a finite field provides essentially more...security than the group of points of a supersingular elliptic curve of comparable size."

The following observation, which was not made in [51], seems to give further support for Verheul's point of view. Given an arbitrary finite field $\mathbb{F}_q$, suppose that one can efficiently construct a trace-zero elliptic curve E over $\mathbb{F}_q$, that is, a curve for which $\#E(\mathbb{F}_q) = q+1$. (If $q \equiv -1 \pmod 4$ or $q \equiv -1 \pmod 6$, then the curve (4) or (5) in §7 has this property; more generally, see §7.6 and Exercise 2 in Chapter 7 of [13] for the prime field case.) We then have the following theorem about the so-called class-VI supersingular curves, which can be viewed as curves of embedding degree $k = 1/2$.

Theorem 1. *Let $\mathbb{F}_q$ be an arbitrary finite field, and let E be a trace-zero elliptic curve over $\mathbb{F}_q$. Suppose that E has equation $y^2 = f(x)$ for odd q and $y^2 + y = f(x)$ for q a power of 2. Let $\beta \in \mathbb{F}_{q^2}$ be a nonsquare in $\mathbb{F}_{q^2}$ for odd q and an element of absolute trace 1 for q a power of 2 (that is, $\mathrm{Tr}_{\mathbb{F}_{q^2}/\mathbb{F}_2}(\beta) = 1$). Let $\widetilde{E}$ be the "twisted" curve over $\mathbb{F}_{q^2}$ with equation $\beta y^2 = f(x)$ for odd q and $y^2 + y + \beta = f(x)$ for q a power of 2. Then $\widetilde{E}(\mathbb{F}_{q^2})$ is a product of two cyclic groups of order $q - 1$, each of which is isomorphic to the multiplicative group of $\mathbb{F}_q$ under the MOV embedding.*

This theorem is an immediate consequence of the classification of supersingular elliptic curves (see Table 5.2 in [34]). Notice that for a trace-zero curve E we have $\#E(\mathbb{F}_q) = q + 1 = q + 1 - \alpha - \overline{\alpha}$ with $\alpha^2 = -q$, and hence $\#E(\mathbb{F}_{q^2}) = q^2 + 1 - \alpha^2 - \overline{\alpha}^2 = q^2 + 1 + 2q$. Thus, for the twist we have $\#\widetilde{E}(\mathbb{F}_{q^2}) = q^2 + 1 - 2q$.

It is reasonable to think that Verheul's theorem can be generalized to the curves in the above theorem. One would need to describe algorithms for obtaining a trace-0 curve for arbitrary q and a "distortion" map in the twisted curve over $\mathbb{F}_{q^2}$. In that case the construction of a Verheul homomorphism would make the DHP easy in *all* finite fields.

Thus, there are two possible interpretations of Verheul's theorem in its (conjectured) general form. The "optimistic" interpretation is that a Verheul homomorphism will never be constructed, because to do so would be tantamount to making the Diffie–Hellman problem easy in all finite fields. Under this interpretation we are forced to conclude that the DHP that arises in pairing-based cryptography is not likely to be provably equivalent to the DHP in finite fields. The "pessimistic" interpretation is that a Verheul homomorphism might some day be constructed. Even if it were constructed just for the class-VI supersingular elliptic curves, that would be enough to render all pairing-based cryptosystems (and also many XTR protocols) completely insecure.

Remark 1. This issue does not arise in the usual non-pairing-based elliptic curve cryptography (ECC). In ECC protocols one uses nonsupersingular curves having large embedding degree k. In fact, k is generally of size comparable to n itself (see [2]), in which case even the input to the Verheul inversion function would have exponential size. Thus, the danger posed by such a map — if it could be efficiently computed — applies only to small k.

Remark 2. The third concern with pairing-based systems — that the problem that their security relies on is not likely to be provably equivalent to a standard problem that is thought to be hard unless both problems are easy

— is analogous to a similar concern with RSA. In [10] Boneh and Venkatesan proved that an "algebraic" reduction from factoring to the RSA problem with small encryption exponent is not possible unless both problems are easy.

4. Parameter Sizes

For the remainder of this paper, unless stated otherwise, we shall suppose that $\mathbb{F}_q$ is a prime field, and we set $q = p$. As mentioned in the last section, in order for a pairing-based cryptosystem to be secure, the field $\mathbb{F}_{p^k}$ must be large enough so that discrete logs cannot feasibly be found using the best available algorithms (the number field and function field sieves). It is also necessary for the prime order n of the basepoint P to be large enough to withstand the Pollard-ρ attack on discrete logs in the group $\langle P \rangle$. Table 1 (see [29, 40]) shows the minimum bitlengths of n and p^k as a function of the desired security level.

security level (in bits)	80	128	192	256
b_n (min. bits of prime subgroup)	160	256	384	512
b_{p^k} (min. bits of big field)	1024	3072	8192	15360
$\gamma =$ the ratio b_{p^k}/b_n	6.4	12	$21\frac{1}{3}$	30

Table 1. Minimum bitlengths of n and p^k

4.1. **Short signatures.** One of the best known uses of pairings is to produce short signatures [9]. Without using pairing methods, the shortest signatures available are the ECDSA, where the length is roughly $2b_n$ bits, and the Pintsov–Vanstone [41] and Naccache–Stern [39] schemes, where the length is roughly $1.5b_n$. The pairing-based Boneh–Lynn–Shacham signatures have length approximately equal to the bitlength of p, which is ρb_n, where $\rho = \log p / \log n$.

Thus, in order to have short Boneh–Lynn–Shacham signatures, one must choose the parameters so that $\rho = \log p / \log n$ is close to 1 and hence $k = \gamma/\rho$ is nearly equal to $\gamma = b_{p^k}/b_n$ (see Table 1). Starting with [38], techniques have been developed to do this with nonsupersingular curves when k can be taken equal to 2, 3, 4, or 6. In those cases the k-th cyclotomic polynomial is linear or quadratic, and the resulting Diophantine equations are computationally tractable. For larger k — notably, for $k = 24$ — the best results are due to Brezing–Weng [11], who obtain $\rho = 1.25$. For example, at the 256-bit security level with 512-bit n they can produce 640-bit signatures, compared to 768 bits for Pintsov–Vanstone and Naccache–Stern and 1024 bits for ECDSA.

It should also be noted that at very high security levels the Boneh–Lynn–Shacham public keys are much larger than in the Pintsov–Vanstone, Naccache–Stern and ECDSA schemes. For instance, at the 256-bit level the latter public keys are roughly 512 bits long, whereas in the pairing-based short signature scheme the public key is a point of $E(\mathbb{F}_{p^k})$, where p^k has about 15360 bits. It suffices to give the x-coordinate of the public-key point, and for even k we may assume that this coordinate is in the smaller field $\mathbb{F}_{p^{k/2}}$ (see the end of §8.2). But even then the public key is about 7680 bits.

4.2. **Changes as security requirements increase.** As our security needs increase, the gap between the desired sizes of n and of p^k increases (see Table 1). At the same time, the optimal choices of algorithms in implementations — and hence the decisions about what families of curves provide greatest efficiency — are likely to be affected. That is, certain tricks that were useful at lower security levels may become less important than other considerations, such as the ability to choose parameters of a special form.

Our first observation is that as b_n and b_{p^k} increase for greater security, the parameter selection methods proposed with nonsupersingular curves of embedding degree $k \geq 2$ do not seem to yield values of n (the prime order of the basepoint) and p (the size of the prime field) that are both Solinas primes. In the literature we have found one construction, due to Scott and Barreto [48], that comes close to solving this problem at the 128-bit security level. Namely, one can apply their construction for $k = 6$ in §5 of [48] with $x = 12Dz^2+1$, where D is a small power of 2 and z is a roughly 80- to 90-bit power of 2 or sum or difference of two powers of 2 that is chosen so that both $n = x^2 - x + 1$ and $p = (x^3 - 2x^2 + 14x - 1)/12$ are primes. Then the bitlengths of n and p^6 are roughly equal to the optimal values in Table 1; moreover, n and p are each equal to a sum or difference of a relatively small number of powers of 2. However, for higher security levels and $k \geq 2$ we do not know of any similar method to achieve nearly-optimal characteristics.

Example 1. Set $D = 1$, $z = 2^{81} + 2^{55}$. Then n is a 332-bit prime with Hamming weight 19, and p is a 494-bit prime with Hamming weight 44.

Our second observation is that for $k > 2$ at the higher security levels it is probably not possible to find suitable supersingular elliptic curves with n having the optimal bitlength that one uses for nonsupersingular curves. The greatest value of k that one can get is $k = 6$, and there are only two supersingular elliptic curves E, both defined over $\mathbb{F}_3$, that have embedding degree 6. Because of the efficiency of the function field sieve in finding discrete logs in characteristic-3 fields, it would be advisable to choose fields $\mathbb{F}_{3^m}$ such that the bitlength of 3^{6m} is larger than the value of b_{p^k} in Table 1. But even using the values in Table 1, there is a serious question of whether one can find a field extension degree $m \approx b_{p^k}/(6 \log_2 3)$ such that $\#E(\mathbb{F}_{3^m})$ has a prime factor n of the appropriate size. There are a relatively small number of possible choices of extension degree, so an elliptic curve group whose order has such a factor n might simply not exist. Moreover, even if it does exist, to find it one needs to factor $\#E(\mathbb{F}_q)$, $q = 3^m$, which cannot feasibly be done unless one is lucky and this number is fairly smooth. For example, at the 256-bit security level we would want the 2560-bit integer $\#E(\mathbb{F}_q)$ to be the product of a roughly 512-bit prime and a 2048-bit cofactor made up of primes that are small enough to be factored out of $\#E(\mathbb{F}_q)$ by the Lenstra elliptic curve factorization method [31]. This is not very likely; in fact, standard estimates from analytic number theory imply that the probability of a random 2048-bit integer being 2^{150}-smooth is less than 2^{-50}.

Very recently, however, techniques have been developed to speed up the pairing computations in the low-characteristic supersingular case to make them virtually independent of the bitlength of n (see [3]). A detailed analysis has not yet been done, so it is still unclear how these supersingular implementations compare with the nonsupersingular ones as the security level increases. In particular, the field operations in the former case are in characteristic 2 or 3, and in the latter case they are in a large prime field.

Our third observation is that as n and p^k increase, one should look more closely at the possibility of switching back to use the Weil pairing rather than the Tate pairing. We shall examine this question when we study efficiency comparisons in §8.

5. Pairing-Friendly Fields

Suppose that we have an elliptic curve E defined over $\mathbb{F}_p$ with even embedding degree k. We shall say that the field $\mathbb{F}_{p^k}$ is *pairing-friendly* if $p \equiv 1 \pmod{12}$ and k is of the form $2^i 3^j$.[2] The following theorem is a special case of Theorem 3.75 of [32]:

Theorem 2. *Let $\mathbb{F}_{p^k}$ be a pairing-friendly field, and let β be an element of $\mathbb{F}_p$ that is neither a square nor a cube in $\mathbb{F}_p$.*[3] *Then the polynomial $X^k - \beta$ is irreducible over $\mathbb{F}_p$.*

The field $\mathbb{F}_{p^k}$ can thus be constructed from $\mathbb{F}_p$ as a tower of quadratic and cubic extensions by successively adjoining the squareroot or cuberoot of β, then the squareroot or cuberoot of that, and so on (see Figure 1). It is easy to see that, if an element of $\mathbb{F}_{p^k} = \mathbb{F}_p[X]/(X^k - \beta)$ is written as a polynomial $\sum_{\ell<k} a_\ell X^\ell$, then it belongs to a subfield $\mathbb{F}_{p^{k'}}$, where $k' = 2^{i'}3^{j'}$, if and only if ℓ is a multiple of $k/k' = 2^{i-i'}3^{j-j'}$ in all of the nonzero terms. Namely, if we set $\mathbb{F}_{p^{k'}} = \mathbb{F}_p[Y]/(Y^{k'} - \beta)$, then the map $Y \mapsto X^{k/k'}$ gives an embedding of the elements of $\mathbb{F}_{p^{k'}}$ (regarded as polynomials in Y) into $\mathbb{F}_{p^k}$. Thus, when we do arithmetic in the field $\mathbb{F}_{p^k}$, we can easily work with the tower of quadratic and cubic field extensions used to construct it.

In practice, it is easy to find a small value of β that satisfies the conditions of the theorem. In that case multiplication by β in $\mathbb{F}_p$ is much faster than a

[2] If $j = 0$, we only need $p \equiv 1 \pmod 4$.

[3] If $j = 0$, it is enough for β to be a nonsquare.

$$\begin{tikzcd}
& N = M[X]/(X^2 - Y) \arrow[d, dash, "2"] \\
& M = L[Y]/(Y^2 - Z) \arrow[d, dash, "2"] \\
& L = K[Z]/(Z^2 - T) \arrow[d, dash, "2"] \\
K' = \mathbb{F}_p[T']/(T'^2 - \beta) \arrow[dr, dash, "2"'] & K = \mathbb{F}_p[T]/(T^3 - \beta) \arrow[d, dash, "3"] \\
& \mathbb{F}_p
\end{tikzcd}$$

Figure 1. Tower of pairing-friendly fields

general multiplication in that field, and so can be neglected in our count of field multiplications. Then the Karatsuba method reduces a multiplication in a quadratic extension to 3 (rather than 4) multiplications in the smaller field; and the Toom–Cook method reduces a multiplication in a cubic extension to 5 (rather than 9) small field multiplications (see §4.3.3 of [25]). This means that we can expect to perform a field operation in $\mathbb{F}_{p^k}$ in time

$$\nu(k)m, \quad \text{where} \quad \nu(k) = 3^i 5^j \text{ for } k = 2^i 3^j,$$

and m denotes the time to perform a multiplication in $\mathbb{F}_p$.

In what follows we shall occasionally perform multiplications in a quadratic subfield $\mathbb{F}_{p^{k/2}} \subset \mathbb{F}_{p^k}$. Because of the Karatsuba technique, we suppose that an $\mathbb{F}_{p^{k/2}}$-operation is equivalent to 1/3 of an $\mathbb{F}_{p^k}$-operation.

Another nice feature of $k = 2^i 3^j$ is that many of the best examples of families of curves for pairing-based cryptography have embedding degree 2, 6, 12, or 24. For instance, we noted in §4.1 that examples with $\rho = \log p / \log n = 1.25$ were constructed in [11] with $k = 24$.

6. Curves with Embedding Degree 1

Let $p > 2$ be a prime of the form $A^2 + 1$. If $4 \mid A$, let E be the elliptic curve defined over $\mathbb{F}_p$ with equation

$$y^2 = x^3 - x. \tag{2}$$

If, on the other hand, $A \equiv 2 \pmod 4$, then let E be the curve

$$y^2 = x^3 - 4x. \tag{3}$$

Theorem 3. *The elliptic curve group $E(\mathbb{F}_p)$ is isomorphic to $\mathbb{Z}/A\mathbb{Z} \oplus \mathbb{Z}/A\mathbb{Z}$. In addition, the map $(x, y) \mapsto (-x, Ay)$ is a "distortion map" on this group in the sense of §4.2 of* [51].

Proof. The curve E is the reduction modulo p of a rational elliptic curve of the form $y^2 = x^3 - N^2 x$, where $N = 1$ in (2) and $N = 2$ in (3). This curve has endomorphism ring $\mathbb{Z}[i]$, where i corresponds to the map $(x, y) \mapsto (-x, iy)$; modulo p the endomorphism i corresponds to the map $(x, y) \mapsto (-x, Ay)$ (note that A is a squareroot of -1 in $\mathbb{F}_p$). According to the theorem in §2 of [27], the Frobenius endomorphism of E is the (unique up to complex conjugation) element α of $\mathbb{Z}[i]$ having norm p and satisfying the congruence $\alpha \equiv \left(\frac{N}{p}\right) \pmod{2 + 2i}$, where $\left(\frac{N}{p}\right)$ denotes the Legendre symbol. When 4 divides A, we see that $\alpha = 1 + Ai \equiv 1 \pmod{2 + 2i}$; when $A \equiv 2 \pmod 4$, we note that $p \equiv 5 \pmod 8$ and hence $\left(\frac{2}{p}\right) = -1$, and so again $\alpha = 1 + Ai \equiv -1 \pmod{2 + 2i}$. Thus, in both cases the number of $\mathbb{F}_p$-points on E is $|\alpha - 1|^2 = A^2$. Moreover, all $\mathbb{F}_p$-points on E are in the kernel of the endomorphism $\alpha - 1 = Ai$, and $E(\mathbb{F}_p)$ is isomorphic as a $\mathbb{Z}[i]$-module to $\mathbb{Z}[i]/Ai\mathbb{Z}[i] \simeq \mathbb{Z}/A\mathbb{Z} \oplus \mathbb{Z}/A\mathbb{Z}$. In the first place, this implies that $E(\mathbb{F}_p)$ is isomorphic as an abelian group to $\mathbb{Z}/A\mathbb{Z} \oplus \mathbb{Z}/A\mathbb{Z}$. In the second place, if $P = (x, y)$ is a point of prime order $n \mid A$, the whole n-torsion group is generated over $\mathbb{Z}$ by P and $iP = (-x, Ay)$; in other words, the endomorphism i is a distortion map. □

Remark 3. As noted in [2] (see Remark 2 of §2), if n is a prime dividing A, then most curves E over $\mathbb{F}_p$ with the property that $n^2 \mid \#E(\mathbb{F}_p)$ have cyclic n-part, that is, they do *not* have n^2 $\mathbb{F}_p$-points of order n, and one has to go to the degree-n extension of $\mathbb{F}_p$ to get all the points of order n. Thus, the property $E(\mathbb{F}_p) \simeq \mathbb{Z}/A\mathbb{Z} \oplus \mathbb{Z}/A\mathbb{Z}$ of the curves (2) and (3) is very unusual, statistically speaking. On the other hand, our elliptic curves are much easier to construct than ones with $n^2 \mid \#E(\mathbb{F}_p)$ and only n points of order n.

Remark 4. The coefficient of x in (2) and (3) can be multiplied by any fourth power N_0^4 in $\mathbb{F}_p$ without changing anything, as one sees by making the substitution $x \mapsto x/N_0^2$, $y \mapsto y/N_0^3$.

Remark 5. In general, a distortion map exists only for supersingular curves; it can exist for a nonsupersingular curve only when $k = 1$ (see Theorem 6 of [51]).

6.1. **History of embedding degree 1.** Although many papers have proposed different families of elliptic curves for use in pairing-based systems, until now no one has seriously considered families with embedding degree $k = 1$. Most authors stipulate from the beginning that $k \geq 2$. We know of only three papers ([21, 23, 51]) that briefly discuss curves E over $\mathbb{F}_p$ with $\#E(\mathbb{F}_p) = p - 1$. In [21], Joux points out that no efficient way is known to generate such curves with $p - 1$ divisible by n but not by n^2, a condition that he wants to have in order to ensure that the Tate pairing value $\langle P, P \rangle$

must always be nontrivial. In [23], Joux and Nguyen repeat this observation. Even though they then show that $\langle P, P\rangle$ is nontrivial for most P even when there are n^2 points of order n, they leave the impression that such curves are less desirable than the supersingular ones that they use in their examples.

In [51], Verheul discusses the nonsupersingular $k = 1$ curves. However, he erroneously states that the discrete logarithm problem in the subgroup $\langle P\rangle$ of prime order n reduces to the discrete log in the field $\mathbb{F}_n$, in which case one needs $b_n \geq 1024$ to achieve 80 bits of security. This mistake leads him also to over-estimate the required bitlength of p, and apparently accounts for his negative view of the practicality of such curves. Thus, the few papers that include the $k = 1$ case quickly dismiss it from serious consideration. No valid reason has been given, however, for excluding such curves.

6.2. **Choice of parameters.** We must choose $A = nh$ such that n and $p = A^2 + 1$ are prime; and, to maximize efficiency, we want

(a) n and p to have respective bitlengths approximately b_n and b_{p^k} corresponding to the desired security level (see Table 1);
(b) n to be a Solinas prime, that is, equal to a sum or difference of a small number of powers of 2;
(c) p also to be a Solinas prime.

The bitlengths of n and p in the following examples are equal to or just slightly more than the minimum values given in Table 1 for the corresponding security level.

Example 2. For 128 bits of security let n be the prime $2^{256}-2^{174}+1$ and let $h = 2^{1345}$. Then $p = (nh)^2+1 = 2^{3202}-2^{3121}+2^{3038}+2^{2947}-2^{2865}+2^{2690}+1$ is prime.

Example 3. For 192 bits of security let n be the prime $2^{386}-2^{342}-1$ and let $h = 2^{3802}$. Then $p = (nh)^2+1 = 2^{8376}-2^{8333}+2^{8288}-2^{7991}+2^{7947}+2^{7604}+1$ is prime.

Example 4. For 256 bits of security let n be the Mersenne prime $n = 2^{521}-1$ and let $h = 2^{7216}$. Then $p = (nh)^2 + 1 = 2^{15474} - 2^{14954} + 2^{14432} + 1$ is prime.

Remark 6. If p is of a certain special form, then discrete logarithms can be found using a special version of the number field sieve (see, for example, [15, 22]). Then the running time for $2b$-bit primes is roughly comparable to the running time of the general number field sieve for b-bit primes. For this reason it is important to avoid the special number field sieve when choosing p. It is clear that certain Solinas primes that one might want to use are of a form that permits the use of a modification of the special number field sieve with running time somewhere between that of the general and the special number field sieves. An analysis of such a modification is currently underway [45]. The results should shed light on the question of whether the advantages of a given Solinas prime are offset by an increased vulnerability to the number field sieve.

Example 5. For the prime $p = 2^{1007}+2^{1006}+2^{1005}+2^{1004}-1 = 240\cdot 2^{1000}-1$ discrete logs in $\mathbb{F}_p$ can be found using the special sieve. The reason is that 2^{200} is a root mod p of the polynomial $f(X) = 240X^5 - 1$, which has small degree and small coefficients.

7. Supersingular Curves with $k = 2$

Suppose that n is a prime and $p = nh - 1$ is also a prime, where $4 \mid h$. If h is not divisible by 3, we let E be the elliptic curve defined over $\mathbb{F}_p$ with equation

$$y^2 = x^3 - 3x; \tag{4}$$

if $12 \mid h$, then we let E be either the curve (4) or else the curve

$$y^2 = x^3 - 1. \tag{5}$$

It is an easy exercise to show that in these cases $\#E(\mathbb{F}_p) = p + 1 = nh$, and so E is a supersingular elliptic curve with embedding degree $k = 2$. Note also that $\beta = -1$ is a nonsquare in $\mathbb{F}_p$, and so $\mathbb{F}_{p^2} = \mathbb{F}_p[X]/(X^2 + 1)$. In addition, the map $(x, y) \mapsto (\zeta x, \varepsilon y)$ is a distortion map in the sense of [51], where

$$\zeta = -1 \quad \text{and} \quad \varepsilon = \text{a squareroot of } -1 \text{ in } \mathbb{F}_{p^2}$$

for the curve (4) and

$$\varepsilon = 1 \quad \text{and} \quad \zeta = \text{a nontrivial cuberoot of 1 in } \mathbb{F}_{p^2}$$

for the curve (5).

7.1. History of embedding degree 2 (supersingular case).

In the early days of elliptic curve cryptography, before the publication of [35] caused people to turn away from such elliptic curves, the supersingular curves (4) with $p \equiv -1 \pmod 4$ and (5) with $p \equiv -1 \pmod 6$ were the most popular examples, because of the simple form of the equation, the trivial determination of the group order, and the easy deterministic coding of integers as points (see Exercise 2 of Chapter 6 of [26]). For similar reasons, Boneh and Franklin used these curves as examples in [7]. On the other hand, authors who study implementation issues tend to shun supersingular curves, perhaps because of the subconscious association of the word "supersingular" with "insecure."

Despite the customary preference for nonsupersingular elliptic curves, there is no known reason why a nonsupersingular curve with small embedding degree k would have any security advantage over a supersingular curve with the same embedding degree. Of course, it is not inconceivable that some day someone might find a way to use the special properties of supersingular elliptic curves to attack the security of the system, perhaps by constructing a Verheul homomorphism from μ_n to the curve (see §3). However, one consequence of a generalized version of Verheul's theorem (see the end of §3) is that if supersingular curves were broken in this way, then the Diffie–Hellman problem in *any* finite field would be easy, and hence nonsupersingular curves of low embedding degree would be insecure as well. This means that the only way that supersingular curves could fall without bringing down all low-embedding-degree curves with them is through some special attack unrelated to a Verheul homomorphism.

Thus, on the one hand one has the remote possibility of a vulnerability of supersingular elliptic curves that is not shared by other curves of low embedding degree. On the other hand, one has the very real efficiency advantages of supersingular curves with $k = 2$. Namely, they provide the benefits of both the $k = 1$ case (flexibility in the choice of n and p) and also the $k \geq 2$ case (speedups coming from subfields).

7.2. **Choice of parameters.** It is a simple matter to select n and h so that both n and p are Solinas primes.

Example 6. At the 80-bit security level let n be the prime $2^{160} + 2^3 - 1$, and let $h = 2^{360}$; then $p = nh - 1 = 2^{520} + 2^{363} - 2^{360} - 1$ is prime.

Example 7. At the 128-bit level let $n = 2^{256} + 2^{225} - 1$, $h = 2^{1326}$, $p = nh - 1 = 2^{1582} + 2^{1551} - 2^{1326} - 1$.

Example 8. At the 192-bit level let $n = 2^{384} - 2^{60} + 1$, $h = 2^{3847}$, $p = nh - 1 = 2^{4231} - 2^{3907} + 2^{3847} - 1$.

Example 9. At the 256-bit level let $n = 2^{521} - 1$, $h = 2^{6704}(2^{521} + 1)$, $p = nh - 1 = 2^{7746} - 2^{6704} - 1$. Note that here $12 \mid h$, so we can use either curve (4) or (5).

As in the $k = 1$ case (see Remark 6), certain Solinas primes can be handled by a special version of the number field sieve for $\mathbb{F}_{p^2}$ which is faster than the general algorithm. An investigation is underway [45] to better understand this speedup and its implications for the use of Solinas primes.

8. Efficiency Comparisons

Let's briefly recall the ingredients in pairing computations. According to Proposition 8 of [37] (see also [12]), the Weil pairing $\widehat{e}(P, Q)$ is given by the formula

$$(-1)^n \frac{F_P(Q)}{F_Q(P)}, \qquad P \neq Q,$$

in which F_P and F_Q are functions whose divisors are $n(P) - n(\infty)$ and $n(Q) - n(\infty)$, respectively. Here F_P and F_Q must be normalized so that $F_P(\infty)/F_Q(\infty) = 1$.

In recent years most authors have preferred to use the Tate pairing rather than the Weil pairing. To evaluate the Tate pairing at points P and Q, one first chooses an auxiliary point R (which must not be equal to P, $-Q$, $P-Q$, or ∞). One then evaluates the ratio

$$\frac{F_P(Q + R)}{F_P(R)},$$

with F_P as above. This preliminary value is an element of $\mathbb{F}_{p^k}^*$ that must be raised to the $((p^k-1)/n)$-th power to convert it to an n-th root of unity.

In pairing computations the procedure to compute $F_P(Q)$, $F_Q(P)$, or $F_P(Q+R)/F_P(R)$ resembles the double-and-add method for finding a point multiple. If n is a Solinas prime, then the number of adds/subtracts is negligible compared to the number of doublings. For each bit of n we have to perform a point doubling which leads to two functions $\ell = \ell_1/\ell_2$ and $v = v_1/v_2$ with constant denominators, and then we have a function-evaluation step of the form

$$\frac{f_1}{f_2} \leftarrow \frac{f_1^2}{f_2^2} \cdot \frac{v_2\ell_1(Q)}{\ell_2 v_1(Q)}.$$

for the numerator (or denominator) in the Weil pairing and

$$\frac{f_1}{f_2} \leftarrow \frac{f_1^2}{f_2^2} \cdot \frac{\ell_1(Q+R)v_1(R)}{\ell_1(R)v_1(Q+R)}.$$

for the ratio in the Tate pairing (note that the denominators ℓ_2 and v_2 cancel in the Tate pairing). Such a procedure is called a “Miller operation” [37].

For this type of computation it is usually most efficient to use Jacobian coordinates (see [20], §3.2.2). A point (X, Y, Z) in Jacobian coordinates corresponds to the point (x, y) in affine coordinates with $x = X/Z^2$, $y = X/Z^3$. In Jacobian coordinates the formula for doubling a point $T = (X, Y, Z)$ takes the form $2T = (X_3, Y_3, Z_3)$ with

$$\begin{aligned} X_3 &= (3X^2 + aZ^4)^2 - 8XY^2, \\ Y_3 &= (3X^2 + aZ^4)(4XY^2 - X_3) - 8Y^4, \\ Z_3 &= 2YZ. \end{aligned}$$

The functions ℓ and v correspond, respectively, to the tangent line to the curve at T and the vertical line through the point $2T$:

$$\begin{aligned} v(x) &= v_1(x)/v_2 = (Z_3^2 x - X_3)/Z_3^2; \\ \ell(x,y) &= \ell_1(x,y)/\ell_2 = (Z_3Z^2y - 2Y^2 - (3X^2 + aZ^4)(xZ^2 - X))/(Z_3Z^2). \end{aligned}$$

8.1. **The case** $k = 1$**.** We first examine the case $k = 1$, where E has equation (2) or (3). Using the above formulas, we count the number S of squarings and the number M of multiplications in $\mathbb{F}_p$ that must be performed for each bit of n. In the case of the Weil pairing, after initially setting $T = P$, $f_1 = f_2 = 1$, for each bit of n we do

$$\begin{aligned} T &\leftarrow 2T \\ f_1 &\leftarrow f_1^2 v_2 \ell_1(Q) \qquad (6) \\ f_2 &\leftarrow f_2^2 \ell_2 v_1(Q). \end{aligned}$$

Our field operation count is $9S + 12M$.[4] Since we must go through essentially the same procedure twice — once for $F_P(Q)$ and once for $F_Q(P)$ — the total number of operations per bit of n required to evaluate the Weil pairing is $18S + 24M$.

The Tate pairing has the advantage that the procedure is needed only once. Namely, we choose R to be the point $(0, 0)$ for $k = 1$, and after initially setting $T = P$, $f_1 = f_2 = 1$, for each bit of n we do

$$\begin{aligned} &T \leftarrow 2T \\ &f_1 \leftarrow f_1^2 \ell_1(Q + R) v_1(R) \\ &f_2 \leftarrow f_2^2 \ell_1(R) v_1(Q + R). \end{aligned} \tag{7}$$

When $k = 1$, we have $9S + 13M$ rather than $18S + 24M$ for each bit of n, and in the case $k \geq 2$ one can gain further savings by working in subfields, as we'll see later. On the other hand, in the Tate pairing computation the preliminary result is an element of $\mathbb{F}_{p^k}^*$ that must be raised to the $((p^k - 1)/n)$-th power to convert it to an n-th root of unity. For high security levels the bitlength of p^k is large compared to that of n (the ratio is what we denoted γ in Table 1), and so the time required for this exponentiation is not negligible. If $k = 1$ and $(p - 1)/n$ has sparse binary representation, or if we use window methods, then the exponentiation is essentially $b_{p^k} - b_n = (\gamma - 1)b_n$ squarings in the field. This adds $(\gamma - 1)S$ to our operation count for each bit of n. If we suppose that $S \approx M$, then we see that the Tate method retains its advantage as long as $(\gamma - 1)S < 9S + 11M \approx 20S$. But when $\gamma > 21$ the Weil computation is faster in the case $k = 1$. According to Table 1, the cross-over point when we should switch to the Weil pairing for $k = 1$ occurs just around the 192-bit security level.

8.2. **The case $k \geq 2$.** Now suppose that $k \geq 2$, and k is even. In that case one distinguishes between full field multiplications in $\mathbb{F}_{p^k}$, multiplications in the quadratic subfield $\mathbb{F}_{p^{k/2}}$ (each of which takes one third as long as a multiplication in the full field, see §5), and multiplications where one or both elements are in $\mathbb{F}_p$. We let S and M, as before, denote squaring and multiplication in the large field $\mathbb{F}_{p^k}$, and we let s and m denote squaring and multiplication in $\mathbb{F}_p$; we suppose that a multiplication of an element in $\mathbb{F}_{p^k}$ by an element in $\mathbb{F}_p$ takes time km. When we make efficiency comparisons, we shall further assume that $S \approx M$, $s \approx m$, and $M \approx \nu(k)m$, where $k = 2^i 3^j$ and $\nu(k) = 3^i 5^j$ (see §5).

[4] In the case $k \geq 2$, without loss of generality we may assume that the coefficient a in the elliptic curve equation $y^2 = x^3 + ax + b$ is equal to -3, in which case in the doubling one saves two squarings. (This is because $3X^2 + aZ^4 = 3(X + Z^2)(X - Z^2)$ when $a = -3$.) When $k = 1$, we suppose that the curve is given by (2) or (3), and so we still have the extra squarings but save one multiplication (by a). If we want to use the equation $y^2 = x^3 - 3x$ instead of (2) or (3), we may do so, provided that $3 \mid h = A/n$ (so that 3 is a quadratic residue in $\mathbb{F}_p$) and 3 is a fourth power in $\mathbb{F}_p$ when $4 \mid A$ but not when $A \equiv 2 \pmod 4$.

In most cryptographic protocols there is some flexibility in the choice of order-n subgroups generated by P and by Q. In particular, one of the two — say, P — can be chosen in $E(\mathbb{F}_p)$. Then $\langle P\rangle$ is the unique subgroup of order n in $E(\mathbb{F}_p)$. In this case the Miller operation for computing $F_P(Q)$ in the Weil pairing is quicker than that for $F_Q(P)$, and so has been dubbed "Miller lite" by Solinas [50].

In addition, in [4] it was pointed out that when the embedding degree k is even, the subgroup $\langle Q\rangle \subset E(\mathbb{F}_{p^k})$ can be chosen so that the x-coordinates of all of its points lie in the quadratic subextension $\mathbb{F}_{p^{k/2}}$ and the y-coordinates are products of elements of $\mathbb{F}_{p^{k/2}}$ with $\sqrt{\beta}$, where β is a fixed nonsquare in $\mathbb{F}_{p^{k/2}}$ and $\sqrt{\beta}$ denotes a fixed squareroot in $\mathbb{F}_{p^k}$. We shall call such values of x and y "real" and "imaginary," respectively, by analogy with the familiar complex plane.

To see that Q can be chosen in this way, we consider the "twisted" elliptic curve $\widetilde{E}$ with equation $\beta y^2 = x^3 + ax + b$. It is easy to show that if E has $p^{k/2}+1-t$ points over the field $\mathbb{F}_{p^{k/2}}$, then $\widetilde{E}$ has $p^{k/2}+1+t$ points over $\mathbb{F}_{p^{k/2}}$. Over the big field $\mathbb{F}_{p^k}$ the number of points on E is equal to the product of the orders of E and its twist $\widetilde{E}$ over $\mathbb{F}_{p^{k/2}}$. Since n^2 divides $\#E(\mathbb{F}_{p^k})$ and only n (but not n^2) divides $\#E(\mathbb{F}_{p^{k/2}})$, it follows that $n \mid \#\widetilde{E}(\mathbb{F}_{p^{k/2}})$.[5] Thus, there is a point $\widetilde{Q} \in \widetilde{E}(\mathbb{F}_{p^{k/2}})$ of order n. The map $(x, y) \mapsto (x, y\sqrt{\beta})$ maps $\widetilde{Q}$ and its multiples to $\mathbb{F}_{p^k}$-points of E (because $(y\sqrt{\beta})^2 = x^3 + ax + b$) that have "real" x and "imaginary" y.

8.3. **Operation count for $k \geq 2$.** When computing the Tate pairing, major savings can be obtained by ignoring terms that are contained in a proper subfield of $\mathbb{F}_{p^k}$ (see [17, 4, 46]). The reason such terms can be ignored is that when raised to the $((p^k-1)/n)$-th power at the end of the Tate pairing computation, they become 1; this is because k is the multiplicative order of p modulo n, and so $(p^k-1)/n$ is a multiple of $p^{k'}-1$ for any proper divisor k' of k. In addition, in Theorem 1 of [4] it is shown that (again because of the exponentiation to the $((p^k-1)/n)$-th power in the Tate pairing) the auxiliary point R can be ignored; that is, the Tate pairing value is $F_P(Q)^{(p^k-1)/n}$. Since the x-coordinate of Q — and hence $v_1(Q)$ — is in $\mathbb{F}_{p^{k/2}}$, it follows that we can drop the entire denominator in (7), and the function-evaluation step becomes simply

$$f_1 \leftarrow f_1^2 \ell_1(Q). \tag{8}$$

A count of the number of operations in a Miller lite point doubling and function evaluation gives $4s+8m+S+M$ for $k=2$ and $4s+(k+7)m+S+M$ for $k \geq 4$ even.

[5] Another way to see this is to note that $n \mid (p^{k/2}+1)$ and also $n \mid (p^{k/2}+1-t)$, from which it follows that $n \mid (p^{k/2}+1+t)$.

The final stage of the Tate pairing computation is the exponentiation. This can be expedited if we use the fact that $n \mid \Phi_k(p)$, where Φ_k is the k-th cyclotomic polynomial; once again, this is a consequence of the assumption that k is the multiplicative order of p modulo n. We then write

$$y^{(p^k-1)/n} = \left(y^{(p^k-1)/\Phi_k(p)}\right)^{\Phi_k(p)/n}.$$

Now raising to the power $(p^k-1)/\Phi_k(p)$ takes very little time (since the p-th power map takes negligible time in extensions $\mathbb{F}_p[X]/(X^k-\beta)$ once $X^{pi} \bmod (X^k-\beta)$ has been precomputed for $i=1,2,\ldots,k-1$). Thus, our estimate for the number of field operations (squarings in $\mathbb{F}_{p^k}$) is the bitlength of $\Phi_k(p)/n$, which is $\frac{\varphi(k)}{k}b_{p^k}-b_n=(\tau_k\gamma-1)b_n$, where we define

$$\tau_k = \frac{\varphi(k)}{k} = \begin{cases} 1/2 & \text{if } k=2^i,\ i\geq 1; \\ 1/3 & \text{if } k=2^i3^j,\ i,j\geq 1. \end{cases}$$

Thus, the operation count for the exponentiation in the Tate pairing is $(\tau_k\gamma-1)S$ for each bit of n.

However, a further speedup is possible because the element that is raised to the $(\Phi_k(p)/n)$-th power has norm 1 over any proper subfield of $\mathbb{F}_{p^k}$. In particular, this element is “unitary” over the quadratic subextension $\mathbb{F}_{p^{k/2}}$. As explained in [47], this means that one need only keep track of the “real” part of powers of the element and can use Lucas sequences to process each bit of the exponent using only one squaring and one multiplication in $\mathbb{F}_{p^{k/2}}$.[6] When $k=2$, this allows us to replace $(\frac{\gamma}{2}-1)S$ by $(\frac{\gamma}{2}-1)(s+m)$ for the exponentiation in the Tate pairing; when $k\geq 4$ is even, the operation count is $(\tau_k\gamma-1)(\widetilde{S}+\widetilde{M})$, where $\widetilde{S}$ denotes a squaring and $\widetilde{M}$ denotes a multiplication in the subfield $\mathbb{F}_{p^{k/2}}$.

If $k>2$ is a multiple of 6 (as it will be for us), then instead of Lucas sequences one could use the trace representations in $\mathbb{F}_{p^{k/3}}$ that Lenstra and Verheul [30] developed in order to make their XTR cryptosystem more efficient (see [47]). This would not necessarily give a better speedup than the Lucas sequences; it is an open question whether the use of the quadratic or cubic subfield is best.

The results so far are summarized in the first two columns of Table 2.

8.4. **Weil or Tate? The case $k\geq 2$.** If we want to compute the Weil pairing rather than the Tate pairing, we need to go through two Miller procedures, one to find $F_P(Q)$ and the other to find $F_Q(P)$. In the case $k\geq 2$, we suppose that $P\in E(\mathbb{F}_p)$, in which case the former is the “Miller lite” part and the latter is the full Miller computation. At first glance it appears that even the Miller lite part is more time-consuming than in the case of the Tate pairing, because we can no longer neglect terms whose

[6] The use of Lucas sequences is closely analogous to computing the n-th power of a complex number on the unit circle; one can use the formula for $\cos(n\theta)$ and work only with the real part. See [47] for details of the Lucas method.

	Exponentiation at end of Tate pairing computation	Miller lite	Full Miller
$k=1$	$(\gamma-1)S$	not applicable	$9S{+}12M$ (Weil) $9S{+}13M$ (Tate)
$k=2$	$(\frac{\gamma}{2}-1)(s+m)$	$4s+8m+S+M$	$4s+8m+S+M$
$k\geq 4$ even	$(\tau_k\gamma-1)(\widetilde{S}+\widetilde{M})$	$4s+(k+7)m+S+M$	$km+4\widetilde{S}+6\widetilde{M}+S+M$

Table 2. Operation counts for each bit of n

$((p^k-1)/n)$-th power equals 1. However, we make the following observation. In any cryptographic application of the pairing it makes no difference if the pairing is replaced by its m-th power, where m is a fixed integer not divisible by n. In particular, for k even we can replace $\widehat{e}$ by its $(1-p^{k/2})$-th power.[7] That means that, just as in the case of the Tate pairing, terms in the Miller computations that lie in $\mathbb{F}_p$ or $\mathbb{F}_{p^{k/2}}$ can be ignored.

In the Miller lite computation the point Q has "real" x-coordinate and "imaginary" y-coordinate; and in the full Miller computation (where we stay with the notation in (6) but with Q now an $\mathbb{F}_p$-point) the point Q has coordinates in $\mathbb{F}_p$. In both the Miller lite and full Miller computations all of the ℓ- and v-terms in (6) except for $\ell_1(Q)$ lie in $\mathbb{F}_{p^{k/2}}$ (or are "purely imaginary"), and so the process (6) again simplifies to (8).

If we make a careful count of the number of operations required for each bit of n, we find that the operation count for the full Miller step is $km+4\widetilde{S}+6\widetilde{M}+S+M$, where, as before, $\widetilde{S}$ is a squaring and $\widetilde{M}$ is a multiplication in $\mathbb{F}_{p^{k/2}}$.[8]

We can decide between the Tate and Weil pairings by comparing the exponentiation column in Table 2 with the full Miller column. As before, we assume that $S \approx M$, $s \approx m$, and $M \approx \nu(k)m$; we also suppose that $\widetilde{S} \approx \frac{1}{3}M$ and $\widetilde{M} \approx \frac{1}{3}M$ (see §5). We find that when $k=2$ the Tate pairing is quicker as long as $\gamma < 20$; but for higher values of γ — that is, starting at the 192-bit security level — we should switch to the Weil pairing. When $k \geq 4$ is even, the value of γ after which the advantage shifts to the Weil pairing is 28.8 for $k=6$, 28.2 for $k=12$, and 27.8 for $k=24$. Thus, for those values of k we should switch to the Weil pairing at the 256-bit security level.

[7] In [24] it was noted that the $(1-p^{k/2})$-th power of $\widehat{e}$ is the same as $\widehat{e}^2$; this is because $n \mid (p^{k/2}+1)$, and so the $(1-p^{k/2})$-th power of an n-th root of unity is the same as the $(1-p^{k/2}+p^{k/2}+1)$-th power.

[8] In the supersingular case (5) with $k=2$, where $a=0$ rather than -3, a multiplication can be replaced by a squaring in the point-duplication part of both the Miller lite and full Miller computations. Of course, this has no effect on Table 3.

Remark 7. These conclusions about the relative speed of the Tate and Weil pairing computations are not definitive. Indeed, not nearly as much effort has been put into finding ways to speed up the full Miller operation in the Weil pairing as has been put into speeding up the exponentiation stage of the Tate pairing. So it is possible that further study of the matter will result in an earlier switch to the Weil pairing, which asymptotically at least is the faster method.

8.5. **Time comparison when $k = 1, 2, 6, 12, 24$.** Let $T(b)$ denote the time required for a multiplication in $\mathbb{F}_p$ for general b-bit p, and let $\widetilde{T}(b)$ denote the time required when p is a b-bit Solinas prime. As before, we assume that $s \approx m$, $S \approx M$, $\widetilde{S} \approx \frac{1}{3}M$, $\widetilde{M} \approx \frac{1}{3}M$, $M \approx \nu(k)m$.

For $k = 1$ the operation count is $9S + 13M + \min((\gamma - 1)S, 9S + 11M)$, where the latter minimum determines the choice of Tate versus Weil pairing. For $k = 2$ the operation count is

$$\begin{aligned} 4s + 8m + S + M + \min((\frac{\gamma}{2} - 1)(s + m), 4s + 8m + S + M) \\ \approx \Big(16 + \min(\gamma, 20)\Big)m. \end{aligned}$$

For $k = 6, 12, 24$ the operation count is

$$\approx \Big(k + 11 + \frac{4}{3}\nu(k) + \min(\frac{2}{9}\gamma\nu(k), k + 6\nu(k))\Big)m.$$

Security (bits)	80	128	192	256
bitlength of p^k	1024	3072	8192	15360
$k = 1$	$27\widetilde{T}(1024)$	$33\widetilde{T}(3072)$	$42\widetilde{T}(8192)$	$42\widetilde{T}(15360)$
$k = 2$ (ss)	$22\widetilde{T}(512)$	$28\widetilde{T}(1536)$	$36\widetilde{T}(4096)$	$36\widetilde{T}(7680)$
$k = 2$ (ns)	$22T(512)$	$28T(1536)$	$36T(4096)$	$36T(7680)$
$k = 6$	$58T(171)$	$77T(512)$	$108T(1365)$	$133T(2560)$
$k = 12$		$203T(256)$	$296T(683)$	$365T(1280)$
$k = 24$				$1049T(640)$

TABLE 3. Pairing evaluation time for each bit of n (ss=“supersingular,” ns=“nonsupersingular”)

These formulas give us the time estimates in Table 3. Notice that for nonsupersingular curves Table 3 suggests that even at the 80-bit security level the choice $k = 2$ is less efficient than higher k, and that, more generally, for $k \geq 2$ large k has an advantage. The comparison between $k = 1$ and $k \geq 2$ is harder to make, because it depends on how much of a saving we are able to achieve when multiplying modulo a Solinas prime rather than an arbitrary prime. It is not clear, for example, whether $42\widetilde{T}(15360)$ is greater or less than $36T(7680)$ or $1049T(640)$. The limited experiments we have conducted with integer multiplication packages were inconclusive.

We estimate that $T(512)$ is at least twice $\widetilde{T}(512)$, and so for $k = 2$ supersingular curves are at least twice as fast as nonsupersingular curves at the 80-bit security level.

Finally, we emphasize that the above analysis is imprecise, and definitive conclusions will be possible only after extensive experimentation. In addition, the relative merits of $k = 1$ and $k \geq 2$ depend on the protocol being used and the types of optimization that are desirable in the particular application.

For example, in identity-based encryption suppose that we are very concerned about the time it takes to convert Alice's identity to a public key, which in the Boneh–Franklin system is a point $I_A \in E(\mathbb{F}_p)$. One is then at a disadvantage when $k = 1$. The reason is that after Alice's identity is hashed into the curve, the resulting point must be multiplied by $h = \sqrt{(p-1)/n}$ to get a point I_A of order n. The bitlength of h is $\frac{1}{2}(\gamma - 1)b_n$. In contrast, when $k \geq 2$ the cofactor $h \approx p/n$ is usually small; its bitlength is $(\rho - 1)b_n$, where $\rho = \log p / \log n$ is generally between 1 and 2. In the $k = 1$ case, to avoid the point multiplication by h one might want to use a different identity-based encryption scheme, such as the one in [42] or [52], where Alice's public key is an integer rather than a point.

9. Open Problems

(1) Prove Verheul's theorem for class-VI supersingular elliptic curves, which, as we saw at the end of §3, contain subgroups isomorphic to the multiplicative groups of all finite fields.

(2) To what extent can the special number field sieve be applied to $\mathbb{F}_p$ for Solinas primes p? For what Solinas primes can we be confident that only the general number field sieve and not the special one can be used to find discrete logarithms?

(3) What Solinas primes can be used with embedding degree $k = 2$ without allowing an attacker to use the special number field sieve for $\mathbb{F}_{p^2}$?

(4) At the 80-bit security level with nonsupersingular elliptic curves, is embedding degree 6 faster than embedding degree 2, as suggested by the preliminary results in §8.5?

(5) For higher security levels such as 192 and 256 bits, is it possible to construct nonsupersingular examples with $k \geq 2$ where n and p^k have roughly b_n and b_{p^k} bits and both n and p are Solinas primes?

(6) Try to find ways to speed up the full Miller operation, and then reexamine the relative speed of the Tate and Weil pairing computations.

(7) Determine more precisely the relative efficiency of curves with embedding degree 1.

(8) When k is a multiple of 6, investigate the use of trace methods similar to the one in [30] to speed up the exponentiation stage of the Tate pairing computation.

(9) Compare implementations in large characteristic p with supersingular implementations in characteristic 2 and 3 [3].

(10) More generally, analyze the efficiency of pairing-based protocols at the AES security levels.

10. Conclusions

It is still hard to say whether pairing-based cryptosystems will be able to provide satisfactory security and efficiency as the desired level of security rises. None of the concerns raised in §3 give sufficient cause to avoid these systems, but they certainly point to the need to proceed with caution.

Despite the spate of recent papers on curve selection for pairing-based cryptosystems, the simplest cases — that of embedding degree 1 and that of supersingular curves with embedding degree 2 — have been largely neglected. To be sure, the $k = 1$ case has some drawbacks, since all of the arithmetic must be done in the large field (there being no subfield) and certain simplifications of the pairing computations when $k \geq 2$ are unavailable. On the other hand, the greater flexibility in choosing the pair (n, p) is a compensating advantage. Thus, the embedding degree 1 case should be seriously considered by implementers of pairing-based cryptography.

Similarly, unless someone finds a way to exploit some special properties of supersingular curves to attack the Bilinear Diffie–Hellman Problem — and we see no reason to believe that this will happen — implementers should pay special attention to supersingular curves with $k = 2$. Those curves have the efficiency advantages of both $k = 1$ (flexibility in the choice of n and p) and also $k \geq 2$ (speedups coming from subfields).

When $k = 1$ the Weil pairing rather than the Tate pairing should be used at security levels significantly above 192 bits, such as the 256-bit level. For $k = 2$ the Weil pairing should be used at the 192-bit level and above, and for $k \geq 4$ even the Weil pairing should be used at the 256-bit level.

For nonsupersingular curves with $k \geq 2$ our preliminary results do not seem to support the viewpoint expressed in [46] that $k = 2$ is the embedding degree that leads to the fastest implementation. Rather, at all security levels considered it appears that among the possible values of $k \geq 2$ one should choose $k = 2^i 3^j$ as large as possible.

There is a need for further study of the relative merits of different values of k as our security requirements increase from the present 80 bits to 128, 192, 256 bits and beyond.

11. Acknowledgments

We would like to thank Darrel Hankerson and Arjen Lenstra for answering our questions about efficient finite field arithmetic, Oliver Schirokauer for answering questions about the number field sieve, Paulo Barreto and Michael Scott for calling our attention to the papers [47] and [3], and Steven Galbraith for commenting extensively on an earlier version of the paper.

References

[1] L. Adleman and M. Huang, Function field sieve methods for discrete logarithms over finite fields, *Information and Computation*, **151** (1999), 5-16.

[2] R. Balasubramanian and N. Koblitz, The improbability that an elliptic curve has subexponential discrete log problem under the Menezes–Okamoto–Vanstone algorithm, *J. Cryptology*, **11** (1998), 141-145.

[3] P. Barreto, S. Galbraith, C. Ó hÉigeartaigh, and M. Scott, Efficient pairing computation on supersingular abelian varieties, http://eprint.iacr.org/2004/375/

[4] P. Barreto, B. Lynn, and M. Scott, On the selection of pairing-friendly groups, *Selected Areas in Cryptography – SAC 2003*, LNCS 3006, Springer-Verlag, 2004, 17-25.

[5] D. Boneh, X. Boyen, and E.–J. Goh, Hierarchical identity based encryption with constant size ciphertext, *Advances in Cryptology – Eurocrypt 2005*, to appear; http://eprint.iacr.org/2005/015/

[6] D. Boneh, X. Boyen, and H. Shacham, Short group signatures, *Advances in Cryptology – CRYPTO 2004*, LNCS 3152, Springer-Verlag, 2004, 41-55.

[7] D. Boneh and M. Franklin, Identity-based encryption from the Weil pairing, *Advances in Cryptology – CRYPTO 2001*, LNCS 2139, Springer-Verlag, 2001, 213-229.

[8] D. Boneh, C. Gentry, and B. Waters, Collusion resistant broadcast encryption with short ciphertexts and private keys, http://eprint.iacr.org/2005/018/

[9] D. Boneh, B. Lynn, and H. Shacham, Short signatures from the Weil pairing, *Advances in Cryptology – ASIACRYPT 2001*, LNCS 2248, Springer-Verlag, 2001, 514-532.

[10] D. Boneh and R. Venkatesan, Breaking RSA may not be equivalent to factoring, *Advances in Cryptology – EUROCRYPT '98*, LNCS 1233, Springer-Verlag, 1998, 59-71.

[11] F. Brezing and A. Weng, Elliptic curves suitable for pairing based cryptography, *Designs, Codes and Cryptography*, to appear; http://eprint.iacr.org/2003/143/

[12] L. Charlap and R. Coley, An Elementary Introduction to Elliptic Curves II, CCR Expository Report 34, 1990, available from http://www.idaccr.org/reports/reports.html

[13] H. Cohen, *A Course in Computational Algebraic Number Theory*, Springer-Verlag, 1993.

[14] D. Coppersmith, Fast evaluation of logarithms in fields of characteristic two, *IEEE Transactions on Information Theory*, **30** (1984), 587-594.

[15] T. Denny, O. Schirokauer, and D. Weber, Discrete logarithms: the effectiveness of the index calculus method, *Algorithmic Number Theory Symp. II*, LNCS 1122, Springer-Verlag, 1996.

[16] G. Frey and H. Rück, A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves, *Math. Comp.*, **62** (1994), 865-874.

[17] S. Galbraith, Pairings, Ch. IX of I. F. Blake, G. Seroussi, and N. P. Smart, eds., *Advances in Elliptic Curve Cryptography*, Vol. 2, Cambridge University Press, 2005.

[18] S. Galbraith, J. McKee and P. Valença, Ordinary abelian varieties having small embedding degree, http://eprint.iacr.org/2004/365/

[19] D. Gordon, Discrete logarithms in $GF(p)$ using the number field sieve, *SIAM J. Discrete Math.*, **6** (1993), 124-138.

[20] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*, Springer-Verlag, 2004.

[21] A. Joux, A one round protocol for tripartite Diffie–Hellman, *J. Cryptology*, **17** (2004), 263-276.

[22] A. Joux and R. Lercier, Improvements to the general number field sieve for discrete logarithms in prime fields, *Math. Comp.*, **72** (2003), 953-967.

[23] A. Joux and K. Nguyen, Separating Decision Diffie–Hellman from Computational Diffie–Hellman in cryptographic groups, *J. Cryptology*, **16** (2003), 239-247.

[24] B. Kang and J. Park, On the relationship between squared pairings and plain pairings, http://eprint.iacr.org/2005/112/

[25] D. Knuth, *The Art of Computer Programming*, 3rd ed., Vol. 2, Addison-Wesley, 1997.

[26] N. Koblitz, *A Course in Number Theory and Cryptography*, Springer-Verlag, 1987.

[27] N. Koblitz, *Introduction to Elliptic Curves and Modular Forms*, 2nd ed., Springer-Verlag, 1993.

[28] N. Koblitz, An elliptic curve implementation of the finite field digital signature algorithm, *Advances in Cryptology – CRYPTO '98*, LNCS 1462, Springer-Verlag, 1998, 327-337.

[29] A. Lenstra, Unbelievable security: matching AES security using public key systems, *Advances in Cryptology – ASIACRYPT 2001*, LNCS 2248, Springer-Verlag, 2001, 67-86.

[30] A. Lenstra and E. Verheul, The XTR public key system, *Advances in Cryptology – CRYPTO 2000*, LNCS 1880, Springer-Verlag, 2000, 1-19.

[31] H. W. Lenstra, Jr., Factoring integers with elliptic curves, *Annals Math.*, **126** (1987), 649-673.

[32] R. Lidl and H. Niederreiter, *Finite Fields*, 2nd ed., Cambridge University Press, 1997.

[33] U. Maurer and S. Wolf, The Diffie–Hellman protocol, *Designs, Codes and Cryptography*, **19** (2000), 147-171.

[34] A. Menezes, *Elliptic Curve Public Key Cryptosystems*, Kluwer Academic Publishers, 1993.

[35] A. Menezes, T. Okamoto, and S. Vanstone, Reducing elliptic curve logarithms to logarithms in a finite field, *IEEE Trans. Inform. Theory, IT-39*, 1993, 1639-1646.

[36] A. Menezes and S. Vanstone, ECSTR (XTR): Elliptic Curve Singular Trace Representation, Rump Session of Crypto 2000.

[37] V. Miller, The Weil pairing and its efficient calculation, *J. Cryptology*, **17** (2004), 235-261.

[38] A. Miyaji, M. Nakabayashi, and S. Takano, New explicit conditions of elliptic curve traces for FR-reduction, *IEICE Trans. Fundamentals, E84-A (5)*, 2001.

[39] D. Naccache and J. Stern, Signing on a postcard, *Financial Cryptography – FC 2000*, LNCS 1962, Springer-Verlag, 2001, 121-135.

[40] National Institute of Standards and Technology, Special Publication 800-56: Recommendation on key establishment schemes, Draft 2.0, 2003.

[41] L. Pintsov and S. Vanstone, Postal revenue collection in the digital age, *Financial Cryptography – FC 2000*, LNCS 1962, Springer-Verlag, 2001, 105-120.

[42] R. Sakai and M. Kasahara, ID based cryptosystems with pairing on elliptic curve, http://eprint.iacr.org/2003/054/

[43] O. Schirokauer, Discrete logarithms and local units, *Phil. Trans. Royal Soc. London A*, **345** (1993), 409-423.

[44] O. Schirokauer, The special function field sieve, *SIAM J. Discrete Math.*, **16** (2002), 81-98.

[45] O. Schirokauer, The number field sieve for primes of low hamming weight, in preparation.

[46] M. Scott, Computing the Tate pairing, *Topics in Cryptology — CT-RSA 2005*, LNCS 3376, Springer-Verlag, 2005, 300-312.

[47] M. Scott and P. Barreto, Compressed pairings, *Advances in Cryptology — Crypto 2004*, LNCS 3152, 140-156, http://eprint.iacr.org/2004/032/

[48] M. Scott and P. Barreto, Generating more MNT elliptic curves, *Designs, Codes and Cryptography*, to appear; http://eprint.iacr.org/2004/058/

[49] J. Solinas, Generalized Mersenne numbers, Technical Report CORR 99-39, University of Waterloo, 1999, http://www.cacr.math.uwaterloo.ca/techreports/1999/corr99-39.pdf

[50] J. Solinas, ID-based digital signature algorithms, 2003, http://www.cacr.math.uwaterloo.ca/conferences/2003/ecc2003/solinas.pdf

[51] E. Verheul, Evidence that XTR is more secure than supersingular elliptic curve cryptosystems, *J. Cryptology*, **17** (2004), 277-296.

[52] B. Waters, Efficient identity-based encryption without random oracles, *Advances in Cryptology – Eurocrypt 2005*, to appear; http://eprint.iacr.org/2004/180/

Department of Mathematics, Box 354350, University of Washington, Seattle, WA 98195 U.S.A.
E-mail address: koblitz@math.washington.edu

Department of Combinatorics & Optimization, University of Waterloo, Waterloo, Ontario N2L 3G1 Canada
E-mail address: ajmeneze@uwaterloo.ca