Contents lists available at ScienceDirect

Journal of Number Theory

www.elsevier.com/locate/jnt

Special Issue: Elliptic Curve Cryptography

Elliptic curve cryptography: The serpentine course of a paradigm shift

Ann Hibner Koblitz[a], Neal Koblitz[b,*], Alfred Menezes[c]

[a] *Women and Gender Studies Program, Arizona State University, Tempe, AZ 85287, USA*
[b] *Dept. of Mathematics, Box 354350, Univ. of Washington, Seattle, WA 98195, USA*
[c] *Dept. of Combinatorics & Optimization, Univ. of Waterloo, Waterloo, Ontario N2L 3G1, Canada*

ARTICLE INFO

Article history:
Received 2 January 2009
Revised 6 January 2009
Available online 17 March 2009
Communicated by David Goss

MSC:
94A60
11T71
14G50
68P25
01A80

Keywords:
Cryptography
Public key
Elliptic curve
Social construction of science
Social construction of technology

ABSTRACT

Text. Over a period of sixteen years elliptic curve cryptography went from being an approach that many people mistrusted or misunderstood to being a public key technology that enjoys almost unquestioned acceptance. We describe the sometimes surprising twists and turns in this paradigm shift, and compare this story with the commonly accepted Ideal Model of how research and development function in cryptography. We also discuss to what extent the ideas in the literature on "social construction of technology" can contribute to a better understanding of this history.

Video. For a video summary of this paper, please visit http://www.youtube.com/watch?v=HHFFvfDoTK4.

1. Introduction

Research into number theoretic questions concerning elliptic curves was originally pursued mainly for aesthetic reasons. But in recent decades such questions have become important in several applied areas, including coding theory, pseudorandom number generation, and especially cryptography.

[*] Corresponding author.
E-mail addresses: koblitz@asu.edu (A.H. Koblitz), koblitz@math.washington.edu (N. Koblitz), ajmeneze@uwaterloo.ca (A. Menezes).

0022-314X/$ – see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jnt.2009.01.006

The first use of elliptic curves in cryptography was H.W. Lenstra's elliptic curve factoring algorithm [70]. Inspired by this unexpected application of elliptic curves, in 1985 N. Koblitz [54] and V. Miller [80] independently proposed using the group of points on an elliptic curve defined over a finite field in discrete log cryptosystems. The primary advantage that elliptic curve systems have over systems based on either integer factorization or the discrete log problem in the multiplicative group of a finite field is the absence of a subexponential-time algorithm (such as those of index calculus type) that could find discrete logs in these groups, provided that the curve and the underlying field are suitably chosen. Consequently, one can use an elliptic curve group that is smaller in size while maintaining the same level of security. In many situations the result is smaller key sizes, bandwidth savings, and faster implementations, features which are especially attractive for security applications in devices where computational power and integrated circuit space are limited, such as smart cards and cell phones.

In 2005 the U.S. National Security Agency posted a paper [87] titled "The Case for Elliptic Curve Cryptography," in which they recommended that industry "take advantage of the past 30 years of public key research and analysis and move from first generation public key algorithms and on to elliptic curves." The NSA commented:

> The best assured group of new public key techniques is built on the arithmetic of elliptic curves. This paper will outline a case for moving to elliptic curves as a foundation for future Internet security. This case will be based on both the relative security offered by elliptic curves... and the relative performance of these algorithms. While at current security levels elliptic curves do not offer significant benefits over existing public key algorithms, as one scales security upwards over time to meet the evolving threat posed by eavesdroppers and hackers with access to greater computing resources, elliptic curves begin to offer dramatic savings over the old, first generation techniques.

In the present paper we give an overview of the history of elliptic curve cryptography (ECC), focusing on the controversies over the security of ECC. This story can be seen as a case study in the history of technology. We start by describing what we call the Ideal Model of research and development in cryptography. The subsequent sections examine to what extent our observations and experiences conform to or contradict that Ideal Model. We then summarize some of the viewpoints found in a subfield of history and sociology of science called Social Construction of Technology,[1] and ask whether those ideas can contribute to a better understanding of the history of ECC.

2. The ideal model

Although not everyone working in cryptography necessarily believes in the validity of what we shall call the Ideal Model of research and development, the general outline given below is a fair representation of the image that cryptographers hope to project to the outside world — especially to laypeople, business customers, and scientists and engineers in related fields.

1. Security always at center stage. The most fundamental feature of any type of cryptographic technology is its security — its resistance to being compromised by an adversary. Although functionality and efficiency are also important — and, for example, users may choose to make do with smaller parameters for increased efficiency if they need only short-term security — the desire to speed up encryption and signature and improve user interface is never a valid reason to lose sight of the basic question of security. In addition, there is a broad realization that complacency is the enemy of security. Hence, the security of the protocols and the underlying mathematical problems is constantly reevaluated in light of new research.

[1] Often referred to by the acronym SCOT, not to be confused with SCOS (Social Construction of Science) or STOC (one of the most prestigious annual conferences in computer science). In this paper in the interest of readability we shall eschew the use of abbreviations and acronyms, with only a few exceptions, such as RSA and ECC.

2. From an art to a science. Cryptographic research and development have largely left behind the days when they depended on the intuition of artisans. Rather than a craft or art, cryptography has truly become a science. The techniques of "provable security" allow marketers of cryptographic protocols to give ironclad guarantees that broad classes of attacks — and these include even attacks that no one has yet imagined — are impossible provided that certain widely believed mathematical assumptions are correct. In addition, the increasing use of automatic software-checkers and theorem-provers gives further reason to expect that human mistakes and failings will play an ever-diminishing role in the evaluation and selection of cryptographic products.

3. Tradition of vigorous debate. The cryptographic community expects and welcomes vigorous debate on the merits of competing systems and methods of analysis. Because of the large interests at stake, these discussions might be heated at times, but the participants understand the need for sharp debate, and so do not take disagreements personally.

4. Special institutions to ensure careful vetting. Although, as in other branches of science, cryptographers have the usual peer review system for academic journals, the most important guarantors of quality control are the program committees that choose papers for presentation at major conferences[2] and the accredited industrial standards bodies that evaluate specific systems and recommend their deployment with suggested parameters.

5. Survival of the fittest. As a result of the checks and balances that are part and parcel of cryptographic research and development, the technology that emerges as the "winner" has passed a stringent series of tests leading to "survival of the fittest." In that sense it can be regarded as intrinsically the best of the alternatives available at the time.

3. The mid-1980s: Discrete logs and factoring

At the heart of any type of public key cryptography is a "one-way" mathematical process or function for which the inverse cannot feasibly be computed. In the famous RSA system the process is to take two very large randomly-generated prime numbers and multiply them together. In a classical Diffie–Hellman system the operation is exponentiation in a finite field. In the former case the inverse process is integer factorization. In the latter type of system the inverse is called the *discrete logarithm* in the finite field.

More precisely, let G be a subgroup of prime order n in the multiplicative group of the field of q elements $\mathbb{F}_q$, where $q = p^f$ is a prime power. (For simplicity we shall generally assume that G has prime order; this is usually the case in cryptographic applications.) Given a generator $g \in G$ (i.e., a non-identity element), the *discrete log problem in* G is the problem, given $y \in G$, of finding an integer x (mod n) such that $y = g^x$.

The simplest example of a Diffie–Hellman system is a basic key agreement scheme that works as follows. Suppose that Alice and Bob wish to agree on a shared key, which will be a random element of G. Alice chooses a secret integer a (mod n) and sends Bob the group element $A = g^a$; Bob chooses a secret integer b (mod n) and sends Alice the group element $B = g^b$. The shared key is then g^{ab}, which Alice can compute as B^a and Bob can compute as A^b. An eavesdropper who monitors the exchange of information has the task of computing g^{ab} knowing g, g^a, and g^b. This problem is known as the *Diffie–Hellman problem* in the group G. The Diffie–Hellman problem can be immediately solved if one knows how to find discrete logs in G, and it is thought to be essentially equivalent to the discrete log problem.

3.1. Index calculus

The most efficient algorithms to solve both the problem of factoring the product of two large primes and the problem of finding the discrete log in a finite field were — and still are — of "index

[2] The vast majority of papers in cryptography are published in refereed conference proceedings, not in journals.

calculus" type. We will illustrate how index calculus works using a simplified version for the discrete log in a prime field. For ease of exposition we will temporarily suppose that G is the entire group $\mathbb{F}_p^*$ rather than a prime order subgroup, so that $n = p - 1$. Let g be a generator of $\mathbb{F}_p^*$. Given $y \in \mathbb{F}_p^*$, we want to find x such that $y \equiv g^x \pmod{p}$.

To do that we first choose a "factor base" consisting of the first s primes, where s is chosen in a certain optimal way so as to minimize running time. The first part of the algorithm, which does not depend on y, consists in finding the discrete logs of the factor base. We choose some random value u_i less than n and compute the least positive residue of $g^{u_i} \pmod{p}$. If that residue has a prime factor greater than the sth prime, we make another choice of u_i. Finally we get a "smooth" residue that has no large prime factor, at which point we can write

$$g^{u_i} \equiv \prod_{j=1}^{s} p_j^{\alpha_{ij}} \pmod{p},$$

and hence

$$u_i \equiv \sum_{j=1}^{s} \alpha_{ij} x_j \pmod{n},$$

where x_j is the discrete log of p_j. When we get more than s such congruences we can find the unknowns x_j by linear algebra over $\mathbb{Z}/n\mathbb{Z}$. Once we have the discrete logs of the p_j, the rest of the algorithm proceeds quickly. We choose random values of u until we get one for which the least positive residue of $g^u y$ has no prime factor greater than p_s, so that we can write

$$g^u y \equiv \prod_{j=1}^{s} p_j^{\beta_j} \pmod{p}.$$

We conclude that the desired discrete log is

$$x \equiv \left(\sum_{j=1}^{s} \beta_j x_j \right) - u \pmod{n}.$$

In the early 1980s the best index calculus algorithms for either factorization or discrete log in a finite field had asymptotic running time of the form $\exp(k^{1/2+\epsilon})$, where k is, respectively, the bitlength of the number to be factored or the bitlength of the size of the finite field. An important exception — which turned out to be a harbinger of things to come — was Don Coppersmith's algorithm [18] for finding discrete logs in the finite fields $\mathbb{F}_{2^k}$. His algorithm had running time of the form $\exp(k^{1/3+\epsilon})$.[3]

After the demise of the early knapsack cryptosystems (which were proposed in the late 1970s and broken within a few years), most cryptographic protocols were based on either factorization or discrete logs in a finite field. This was a little disconcerting, because it appeared that, despite the superficial dissimilarity between the two problems, the most efficient algorithms were very similar. In such circumstances one might speculate that a major advance in solving one of the two supposedly "hard" problems would soon be followed by a similar improvement in methods to solve the other one. (And in fact a few years later, when the number field sieve was developed for factoring, it was soon followed by a version that finds discrete logs in a prime field [36].) In that sense the two problems are not really independent, and it might have seemed that cryptographers were putting all their security eggs in one basket.

[3] In discussing running times of attacks on number theoretic problems, we shall not distinguish between heuristic and proven time estimates.

3.2. Elliptic curve cryptography (ECC)

In 1984 Hendrik Lenstra circulated a preprint describing a new factorization method. Like the index calculus algorithms available at the time, it also has running time $\exp(k^{1/2+\epsilon})$ to factor a k-bit integer, but it has several features that mark a radical departure from the other algorithms with that running time. First, it is not an index calculus algorithm, and it seems that no algorithm similar to Lenstra's can be developed for the discrete log problem in a finite field. Second, although it is not more efficient than index calculus for factoring an RSA-type integer — that is, a product of two primes of roughly the same size — it has the advantage that its running time depends on the size of the smallest prime factor, not on the size of the number itself. This was later put to use in factoring other types of numbers that arise in cryptography.

But the most striking feature of Lenstra's factoring algorithm [70] was that it used elliptic curves. This was the first application of elliptic curves in cryptography, and it set in motion a process of finding cryptographic uses for many types of "pure" mathematics — especially arithmetic algebraic geometry — that had never before been studied for this purpose.

In 1985 V. Miller [80] and N. Koblitz [54] proposed a completely different cryptographic use of elliptic curves: constructing Diffie–Hellman type protocols using the group of points of an elliptic curve defined over a finite field rather than the multiplicative group of a finite field. Let E be given by a Weierstrass equation

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,$$

with $a_i \in \mathbb{F}_q$. The groups used in ECC are the prime order subgroups G of the $\mathbb{F}_q$-points of E. In the setting of the elliptic curve group law, which is customarily written in additive notation, the discrete log problem asks: given $P, Q \in G$, find $x \pmod{n}$ such that $Q = xP$.

The most important reason for considering ECC was that it seemed unlikely that index calculus could be adapted for use in an elliptic curve group. The reason is that in order to apply the idea of index calculus, one needs a set of "small" elements (the "factor base") such that a reasonable proportion of the remaining elements can be efficiently written in terms of the factor base. In [80] Miller made an argument using the Néron–Tate height function (see also [99], which contains a much more detailed discussion) that if one tries to use the most natural notion of "smallness" one will find that there are very few points of bounded size, not nearly enough to form a factor base for index calculus.

In the early years of ECC a popular choice of curves for expository purposes was the equation $y^2 = x^3 - x$ defined over a prime field $\mathbb{F}_p$. If $p \equiv 3 \pmod 4$ — this is known as the *supersingular* case — it is an easy exercise to show that the group order is $p + 1$. One can then quickly find a p such that this group has a very large prime order subgroup. The procedure is similar to the following method for finding a prime field whose multiplicative group has a prime order subgroup of smallest possible index 2. Namely, let n be a Sophie Germain prime, and set $p = 2n + 1$. Then $\mathbb{F}_p^*$ has a subgroup of prime order $n = (p - 1)/2$. In the elliptic curve case choose a prime n for which $p = 4n - 1$ is prime; then the group of $\mathbb{F}_p$-points on $y^2 = x^3 - x$ is the product of the group of 4 points of order 2 and a subgroup of prime order $n = (p + 1)/4$.

In characteristics 2 and 3 the supersingular curves had another convenient feature: point doubling on a supersingular curve in characteristic 2 and point tripling on a supersingular curve in characteristic 3 take negligible time.

Convenient as these parameter choices were, the early writers on ECC later regretted having used them, because in 1991 we learned that the discrete log problem on a supersingular curve is much easier to solve than on most curves (see Section 4). Among all elliptic curves defined over $\mathbb{F}_p$ the supersingular ones are a tiny proportion — a randomly selected curve has probability only $O(1/\sqrt{p})$ of being supersingular — but the frequent use of supersingular curves for ease of exposition gave some people an exaggerated impression of their importance.

Avoiding supersingular curves does not, however, mean avoiding curves that are very easy to compute with. For example, one can use the very same curve $y^2 = x^3 - x$ but choose $p \equiv 1 \pmod 4$. In that case a formula for the group order goes back to Gauss. If p is expressed as a sum of two squares, $p = a^2 + b^2$ with $a \equiv 1 \pmod 4$, then $\#E(\mathbb{F}_p) = p + 1 - 2a$. (A similar example is given by $y^2 = x^3 + 1$ with $p \equiv 1 \pmod 3$.)

The two curves in the last paragraph are obtained by reduction mod p of an elliptic curve defined over $\mathbb{Q}$ that has *complex multiplication* by, respectively, the fourth roots and the third roots of unity. Namely, on the curve $y^2 = x^3 - x$ we have the automorphism $(x, y) \mapsto (-x, iy)$, and on the curve $y^2 = x^3 + 1$ we have $(x, y) \mapsto (\zeta x, y)$, where $\zeta = \exp(2\pi i/3)$.

3.3. ECC protocols

By a *protocol* we mean a specific sequence of steps that are carried out in a particular application. Most of the protocols using elliptic curves were obtained by simply repeating the ones that had been developed for finite fields with the obvious modification in notation. Until the advent of pairing-based cryptography (see Section 9), there were no important protocols that exploited any of the rich structure of elliptic curves.

However, it was a little tricky to find a good elliptic curve analogue of the finite field Digital Signature Algorithm that NSA developed in 1991 (see Section 7). We now describe this construction. In the elliptic curve digital signature algorithm (ECDSA) we suppose that Alice wants to sign a message that she has sent to Bob, and both Alice and Bob are using the same elliptic curve defined over $\mathbb{F}_q$ containing a subgroup G of prime order n with generator P. For simplicity we shall suppose that q is a prime, although the construction can easily be adapted to a prime power q as well.

As usual, we suppose that we have a "hash function" that assigns a value H to a message; H plays the role of the message's "fingerprint" in the sense that we assume that it is computationally infeasible to find two different messages with the same hash value.

ECDSA key generation. Each user Alice constructs her keys by selecting a random integer x in the interval $[1, n-1]$ and computing $Q = xP$. Alice's public key is Q; her private key is x.

ECDSA signature generation. To sign a message having hash value H, $0 < H < n$, Alice does the following:

(1) She selects a random integer k in the interval $[1, n-1]$.
(2) She computes $kP = (x_1, y_1)$ and sets r equal to the least non-negative residue of x_1 mod n (where x_1 is regarded as an integer between 0 and $q-1$). (Note: If $r = 0$, then she must go back to step (1) and select another k.)
(3) She computes k^{-1} mod n and sets s equal to the least non-negative residue of $k^{-1}(H + xr)$ mod n. (Note: If $s = 0$, then she must go back to step (1).)

The signature for the message is the pair of integers (r, s).

ECDSA signature verification. To verify Alice's signature (r, s) on a message, Bob does the following:

(1) Obtain an authenticated copy of Alice's public key Q.
(2) Verify that r and s are integers in the interval $[1, n-1]$, and compute the hash value H of the message.
(3) Compute $u_1 = s^{-1}H$ mod n and $u_2 = s^{-1}r$ mod n.
(4) Compute $u_1P + u_2Q = (x_0, y_0)$ and, regarding x_0 as an integer between 0 and $q-1$, set v equal to the least non-negative residue of x_0 mod n.
(5) Accept the signature if and only if $v = r$.

Notice that if Alice generated her signature correctly, then $u_1P + u_2Q = (u_1 + xu_2)P = kP$ because $k \equiv s^{-1}(H + xr) \pmod n$, and so $v = r$.

3.4. Early algorithms for elliptic curve discrete logs

At first the only algorithms known to solve the elliptic curve discrete log problem were *generic* ones, that is, they have nothing to do with the specific structure of the elliptic curve group. The first such algorithm, designed in the setting of finite field discrete logs by Pohlig and Hellman [88], uses the Chinese remainder theorem to reduce the problem to the discrete log problem in the prime order subgroups. This is why groups of prime order are usually chosen for Diffie–Hellman type cryptosystems.

In a group G of prime order n the two best generic algorithms — Shanks' "baby-step/giant-step" and Pollard's rho [89] — each requires time roughly $O(\sqrt{n})$; for this reason they are known as *squareroot attacks* on the discrete log problem. Although Shanks' method has the advantage of being deterministic, it has a very large storage requirement — also of order $\sqrt{n}$ — and so in practice some randomized version of the Pollard-rho method is preferred.

The general idea of Pollard is to take a pseudo-random walk in G (i.e., it is deterministic, but heuristically seems to have a high degree of randomness) using certain combinations of the basepoint P and the point Q with the unknown discrete log. As soon as the walk hits the same place twice, one can immediately solve for the discrete log. The $\sqrt{n}$ estimate comes from the "birthday paradox."

As we shall soon discuss, subsequently faster-than-squareroot algorithms were found for various classes of elliptic curves. However, it still appears — after a quarter century of ECC — that the types of curves used in most cryptographic applications cannot be attacked by anything faster than the generic algorithms.

The last statement has to be qualified somewhat. One can group together a point and its negative so as to apply Pollard-rho to a set of $(n-1)/2$ pairs of points; this gives a speed-up of generic Pollard-rho by a factor of $\sqrt{2}$. Moreover, if the curve E is defined over a much smaller subfield — say, over $\mathbb{F}_{q_0}$, where $q = q_0^\ell$ — then by grouping together Frobenius conjugacy classes of points (obtained by applying the map $(x, y) \mapsto (x^{q_0}, y^{q_0})$) one can speed up the generic algorithm by an additional factor of $\sqrt{\ell}$ (see [31,108]). This is a relatively small effect, but it does have to be taken into account if one wants the efficiency advantage that comes from choosing an elliptic curve defined over a small field.

4. The Weil pairing attack

If E is an elliptic curve defined over $\mathbb{F}_q$, $q = p^f$, let $E[n] \subset E(\overline{\mathbb{F}}_q)$, where $\overline{\mathbb{F}}_q$ is the algebraic closure of $\mathbb{F}_q$, denote the set of all $\overline{\mathbb{F}}_q$-points of order n. If n is prime to p, then $E[n] \approx (\mathbb{Z}/n\mathbb{Z}) \times (\mathbb{Z}/n\mathbb{Z})$. Let $\mathbb{F}_{q^k}$ be the smallest field that contains the coordinates of the points in $E[n]$. In our applications n will be a prime dividing $\#E(\mathbb{F}_q)$ but not dividing $q-1$. In that case the integer k, which is called the *embedding degree*, is also equal to the smallest positive integer such that $n|(q^k-1)$. Let $\mu_n \subset \mathbb{F}_{q^k}^*$ denote the subgroup of nth roots of unity. Then the Weil pairing is a non-degenerate skew-symmetric bilinear map

$$E[n] \times E[n] \to \mu_n.$$

This pairing can be efficiently computed if k is not too big (see [81,82]).

The first use of the Weil pairing in cryptography was to solve the discrete log problem in subexponential time on an elliptic curve of low embedding degree. In [76] it was shown how the Weil pairing could be used to transport the elliptic curve discrete log problem to the discrete log problem in the group $\mathbb{F}_{q^k}^*$. In the latter group index calculus methods are effective provided that k is very small. (A similar attack using the Tate pairing was given by Frey and Rück [28], who introduced the pairing into cryptographic use.)

However, curves for which k is small are very rare. The most important class of such curves are the supersingular curves, i.e., those for which $\#E(\mathbb{F}_q) \equiv 1 \pmod{p}$. For those curves $k \leqslant 6$; in contrast, it was shown in [4] that very few *ordinary* (meaning non-supersingular) curves have small embedding degree. A randomly chosen curve over $\mathbb{F}_p$ has probability $O(p^{-1/2})$ of being supersingular, and a randomly chosen pair consisting of p and an ordinary curve over $\mathbb{F}_p$ has probability only $O(p^{-1})$ of having bounded embedding degree.

Nevertheless, the Weil pairing attack, which was the first subexponential algorithm for the discrete log problem on a prime order subgroup of points on any elliptic curve, had a major impact. As mentioned before, it had an especially chastening effect on those of us who for expository convenience had used supersingular curves with embedding degree $k = 2$ in articles and books.

For almost a decade it was widely assumed that supersingular curves should be completely avoided in cryptography, and that even an ordinary curve had to be checked to see that its embedding degree was fairly large. In practice $k \geqslant 20$ was usually considered to be sufficient to guarantee intractability of the discrete log problem in $\mathbb{F}^*_{q^k}$. However, some cryptographers have gone to the extreme of insisting that $k \geqslant (q-1)/100$ (see Section 11).

In reality, as long as $k \geqslant 6$, with the state-of-the-art techniques available at the time the discrete log problem in the field $\mathbb{F}_{q^k}$ into which $E(\mathbb{F}_q)$ embeds was at least as hard to solve using index calculus as the discrete log problem was to solve directly on $E(\mathbb{F}_q)$ using Pollard-rho. And on rare occasions curves with $k = 6$ were considered in cryptography papers (for example, [57]). But such curves were almost universally shunned. Certainly no curves with low embedding degree were permitted in the ECC standards.

5. Hyperelliptic curve cryptography

Just as the group of points on an elliptic curve can be used to construct cryptographic protocols, so can the jacobian group of a genus-g hyperelliptic curve

$$y^2 + h(x)y = x^{2g+1} + a_1x^{2g} + \cdots + a_{2g+1}$$

(with $\deg h \leqslant g$) defined over $\mathbb{F}_q$. This is a natural generalization of the elliptic curve group, which is the case $g = 1$, and it was first proposed for use in cryptography in 1989 [55]. The group order for a hyperelliptic jacobian is approximately q^g, that is, the same size as one gets in elliptic curve cryptography working over the extension field $\mathbb{F}_{q^g}$. In other words, if g is large, one can work over a small field. On the other hand, the group operation is much more cumbersome than in the elliptic curve case: it uses a process of reduction of divisors that is closely analogous to Gauss' method for composition of binary quadratic forms. But in any case it turns out that the discrete log problem is actually much easier on the jacobian of a high-genus curve than on a comparably sized group of points of an elliptic curve, as shown by Adleman, DeMarrais, and Huang [1] in 1994.

5.1. Two meanings of "complexity"

The subexponential-time algorithm in [1] for the discrete log problem on the jacobian of a high-genus curve came as a big surprise to people who were starting to think about implementing hyperelliptic curve cryptography. When N. Koblitz proposed such systems in [55], he thought that the difficulty of the discrete log problem for a genus g curve would probably be at least as great as that of the corresponding problem on an elliptic curve. Isn't it reasonable to assume that a problem would be at least as hard to solve on a more complicated object (a g-dimensional jacobian) as on a relatively simple object?

That way of thinking was a "rookie mistake" for a cryptographer to make, because he was confusing two meanings of "complexity": conceptual complexity and computational complexity. True, standard treatments of algebraic curves often describe the genus g as a measure of the complexity of the curve. And over a fixed field it is reasonable to regard high-genus curves as more complicated to compute with than elliptic curves.

However, in practical applications what is fixed is not the field $\mathbb{F}_q$, but rather the bitlength of the group size q^g. And the algorithm in [1], while not subexponential in $\log q$, *is* subexponential in $g \log q$ as g grows. Thus, from a computational standpoint the discrete log problem on a suitably chosen elliptic curve over $\mathbb{F}_{2^{163}}$ has much higher complexity than the discrete log problem on the jacobian of a genus-163 hyperelliptic curve over $\mathbb{F}_2$. Such an elliptic curve at present would provide adequate security for cryptographic applications, whereas the genus-163 curve definitely would not.

What made high-genus hyperelliptic curves computationally simpler than low-genus curves was that there was a natural choice of "small" divisors that could be used in index calculus. Namely, elements of the jacobian can be uniquely represented by certain pairs of polynomials of the form $(a(x), b(x))$, where $\deg a \leqslant g$ and $\deg b < \deg a$. The elements represented by $(a(x), b(x))$ with $a(x)$ of small degree can be used as the "factor base" (see Section 3.1) for index calculus, and this was what Adleman, DeMarrais, and Huang did.

5.2. Further developments in genus $\geqslant 3$

A few years after the subexponential index calculus algorithm was found for high-genus jacobians, Gaudry and others saw that for much smaller genus one could get similar algorithms that, while not subexponential, were significantly faster than Pollard-rho. The best and most recent of them [22] can find discrete logs in the jacobian group of a hyperelliptic curve over $\mathbb{F}_q$ of fixed genus g in $O(q^{2-2/g})$ operations. Since a squareroot attack takes $O(q^{g/2})$ operations, this is a big improvement for $g \geqslant 3$; for example, we get $O(q^{4/3})$ rather than $O(q^{3/2})$ for genus 3. Somewhat surprisingly, for *non*-hyperelliptic curves of fixed genus g, Diem [20] found an algorithm with significantly faster running time than in the hyperelliptic case, namely $O(q^{2-2/(g-1)})$. However, for genus 2 thus far we have nothing faster than Pollard-rho in the general case.

6. RSA vs. ECC

6.1. Early attitudes toward ECC

For several years after elliptic curve cryptography was proposed, the most common response from cryptographers was curiosity and approval. Although most researchers had never studied elliptic curves and at first had little understanding of the technical issues in ECC, they tended to react positively to the general idea of a type of cryptography based on algebraic curves. In the late 1980s a broad range of mathematicians were starting to work in the field, and the growing interest in cryptography by mathematicians and the increasing sophistication of the mathematics that was being introduced perhaps were taken as an indication that public key cryptography was coming into its own and would soon be "ready for prime time."

Moreover, ECC was not perceived as a commercial threat to anyone. Commercial rivalries were still in the future, and even RSA had not yet become a major force in commerce. In fact, to most people in the 1980s the term "information security" meant that you bought a lock for your file cabinet. In many ways the atmosphere during the first decade of academic work on cryptography was relaxed, open-minded and curious — a contrast with what came later.

6.2. ECC becomes a commercial threat

In the late 1980s three professors at the University of Waterloo formed a company, now called Certicom, that developed and promoted ECC. Researchers affiliated with Certicom started attending meetings of industrial standards bodies, where they lobbied for the inclusion of ECC protocols in the recommendations. For example, the Elliptic Curve Digital Signature Algorithm (see Section 3.3) was making headway as an efficient alternative both to RSA signatures and to NSA's original finite field Digital Signature Algorithm (see Section 7), although its final approval and inclusion in the standards did not occur until 1999 and 2000 [3,86].

Meanwhile, RSA Data Security was finally enjoying commercial success. RSA cryptography was becoming well known among the general public, and it had a virtual monopoly on the market for public key cryptography. On the other hand, there were clouds on the horizon. The recently developed number field sieve factoring method had lowered the running time for factoring a k-bit RSA modulus from $\exp(k^{1/2+\epsilon})$ to $\exp(k^{1/3+\epsilon})$. This was a dramatic improvement, and it meant that the size of the numbers recommended for safe use of RSA would soon grow to over a thousand bits. As small devices such as cell phones, pagers, and smart cards entered the mass market, promoters of ECC were cautioning that RSA would have significant disadvantages in "constrained environments" that had low storage capacity and bandwidth.

6.3. "ECC Central": RSA strikes back

In 1997 RSA Data Security put on its website a section called "ECC Central," running to nine printed pages, in order to respond to what they termed "significant coverage in the media" and "the current excitement around elliptic curve cryptosystems." The website announced that

> The recommendation of RSA, supported by the world's top cryptographers and cryptanalysts, is that the use of ECC puts customer data at far too great a risk and that further study and testing is required.

The company's main argument to justify its recommendation was that

> ...the integer factorization problem (on which the security of RSA depends) has been studied intensively by number theorists and mathematicians around the world for literally hundreds of years and there is no doubt that the RSA cryptosystem has stood the test of time very well. By contrast, research into the elliptic curve discrete logarithm problem (on which the security of elliptic curve cryptosystems depends) and on elliptic curve cryptosystems in general represents a fraction of that spent on both RSA and the integer factorization.

The last section of the RSA policy statement rhetorically asked "Elliptic curve cryptosystems ready for prime time?" and answered the question in the negative.

RSA buttressed its position by appending a section called "The Experts Comment on ECC" in which eight cryptographers offered their skeptical commentary. Most interesting were the statements by two of the three founders of RSA, Leonard Adleman and Ron Rivest. Adleman started by saying, "I am suspicious of elliptic curve cryptosystems," and then explained his suspicion by citing his work [1] with DeMarrais and Huang giving a subexponential algorithm for the analogous problem for high-genus hyperelliptic curves. He correctly pointed out that those curves had been thought to be at least as secure as elliptic curves (see Section 5.1).

Rivest's comments were the most erudite:

> But the security of cryptosystems based on elliptic curves is not well understood, due in large part to the abstruse nature of elliptic curves. Few cryptographers understand elliptic curves, so there is not the same widespread understanding and consensus concerning the security of elliptic curves that RSA enjoys. Over time, this may change, but for now trying to get an evaluation of the security of an elliptic-curve cryptosystem is a bit like trying to get an evaluation of some recently discovered Chaldean poetry.

Supporters of ECC countered these statements by pointing out that the claim that the security of RSA rested on sturdier ground was a little misleading. Although Gauss himself spoke of his interest in the integer factorization problem almost two hundred years ago, it is a gross exaggeration to say that mathematicians have been studying the problem intensively since that time. In fact, it was the invention of RSA cryptography in 1977 that stimulated stepped-up efforts to improve algorithms, and most of the research on integer factorization is relatively recent.

In addition, the elliptic curve discrete log problem is analogous to the corresponding problem in a finite field, and most of the approaches to the elliptic curve problem are closely related to approaches that were studied earlier in the finite field context. The discrete log problem in a finite field played an important role in cryptography long before the invention of RSA and public key: in the 1950s it arose in work on shift-register sequences. Index calculus was actually first developed in the 1920s by Kraitchik [66,67] for the discrete log problem in a prime field, and it was not applied to integer factorization until many years later.

Moreover, it is not quite right that more sophisticated mathematical knowledge is needed to study possible attacks on elliptic curves than to study approaches to the integer factorization problem. Although it takes a little more mathematical background to understand the group law on an elliptic curve than to understand the modular exponentiation in RSA, it is illogical to conclude from this that research on breaking RSA through factoring is easier to understand than research on finding discrete logs on elliptic curves. At the time that "ECC Central" appeared, by far the most complicated mathematics that had ever been applied to solve either problem was the number field sieve, which had had such a dramatic impact on factoring. Contrary to what Rivest implied in the remark quoted above, in 1997 the number of cryptographers with sufficient mathematical background to analyze and improve upon the best attacks on integer factorization was less than the number who were capable of evaluating the best attacks on the elliptic curve discrete log problem. As number theorists know well, there is no correlation between ease of understanding the statement of a problem and the level of difficulty involved in making progress in solving it.

6.4. Xedni calculus and liftings

In September 1998 J. Silverman circulated an outline of an attack [97] on the elliptic curve discrete log problem. He called it "xedni" calculus (xedni = index spelled backwards) because in some sense it reversed the steps in index calculus. Suppose we have two points P, Q in a prime order subgroup $G \subset E(\mathbb{F}_p)$, and we want to find x such that $Q = xP$. Silverman's general idea was to start by randomly generating a few (no more than 9) integer linear combinations $P_i = a_i P + b_i Q$ and then lifting them to points $\widetilde{P}_i$ with $\mathbb{Z}$-coordinates. He then finds a lifting $\widetilde{E}$ over $\mathbb{Q}$ that reduces to E mod p, passes through the $\widetilde{P}_i$, and satisfies some other conditions that, if one believes the heuristics of the Birch and Swinnerton–Dyer conjecture (and uses an analytic formula of J.F. Mestre for the Mordell rank of $\widetilde{E}$), increase the likelihood that the points $\widetilde{P}_i \in \widetilde{E}$ will be dependent over $\mathbb{Z}$. If they are dependent, then it is easy to find x.

Although the outline of xedni calculus was fairly simple, its running time would depend on some subtle considerations that were hard to pin down in computational terms. At first it was completely unclear whether or not xedni calculus would be more efficient than other algorithms for finding elliptic curve discrete logs. This was still a time when RSA and ECC were in fierce competition, and the promoters of ECC feared that RSA people would seize upon the opportunity provided by xedni calculus and proclaim to the world that ECC had been broken.

Fortunately, however, it turned out that slight modifications of Silverman's algorithm could be used to solve not only the discrete log problem in the multiplicative group of $\mathbb{F}_p$ (by applying it to a degenerate elliptic curve, i.e., a rational curve over $\mathbb{F}_p$), but also the problem of factoring an RSA modulus N (by applying it to a degenerate elliptic curve over $\mathbb{Z}/N\mathbb{Z}$). In other words, if Silverman's algorithm destroyed ECC, then it would destroy RSA as well. This feature of the algorithm was very opportune, because it gave us time to analyze it without having to worry about RSA people making premature announcements about the threat from xedni calculus.

The xedni algorithm was found to be extremely inefficient; in fact, it seemed to take super-exponential time to find discrete logs [46]. The reason was basically the same one that Miller [80] had used back in 1985 to argue that index calculus would not work on elliptic curves. Namely, the Néron–Tate height function guaranteed that $\widetilde{E}(\mathbb{Q})$ could not have a large number of "small" points.

Once again the height function played a crucial role in explaining why lifting techniques could not be efficiently used to find discrete logs. In 2000 N. Koblitz gave a talk on this at the ECC conference in Essen titled "Miracles of the Height Function — A Golden Shield Protecting ECC." Subsequent developments would show that Koblitz's celebration of the "golden shield" was premature, as researchers found faster-than-squareroot and even subexponential index calculus attacks on some elliptic curves defined over certain classes of finite fields (see Section 10). However, in none of these partial successes of index calculus have liftings to global fields played any role.

At the ECC conference in 2007 Silverman [98] gave a much more systematic analysis of the failure of four possible plans of attack on the discrete log problem based on lifting to a global field. But despite the repeated failure of lifting-based approaches, one cannot be absolutely certain that no lifting will ever work. For this reason some people recommend staying away from elliptic curves over $\mathbb{F}_q$ for which it is easy to construct a lifting to a number field that has special properties that might some day prove useful to an attacker. In particular, in Section 11 we will discuss the recommendation of the Brainpool consortium that all curves used in ECC have complex multiplication by quadratic imaginary fields of very high class number so that they cannot efficiently be lifted to a CM-curve over a number field.

7. The role of NSA

In the 1970s and early 1980s the U.S. National Security Agency was an extremely secretive organization. The standing joke at the time was that NSA stood for "No Such Agency." People from NSA would attend the crypto conferences that were starting to be held, but they would never identify where they worked.

NSA was unhappy with the sudden growth of open research on cryptography that had been stimulated by the invention of public key systems and especially RSA. In 1980 they made a heavy-handed and ultimately unsuccessful attempt to impose a system of prior restraint on publication of mathematical articles that they judged to have cryptographic relevance (see [69]).

But by the time the debates between RSA and ECC heated up in the 1990s, NSA had changed in a fundamental way — it had "come in from the cold." There were two main reasons for the transformation of NSA into an organization that started to participate openly in the cryptographic research community.

The first reason was a broadening of NSA's mandate after the passage of the Computer Security Act of 1987. Originally NSA had been given responsibility only for communication security for the U.S. military and government agencies. But with the emergence of the Internet and other technologies, communications were increasingly mixed up with computers, and it was becoming clear that issues of computer security and communication security could not be separated. In addition, most government communications were integrated with the public network and faced the same threats as everyone else. So government security could not be kept separate from similar issues in the private sector. Thus, in 1987 the U.S. government agency NIST (National Institute for Standards and Technology, the new name of the National Bureau of Standards) was given a mandate to investigate and help establish standards for security of all sorts of computer and communication networks. According to some accounts [25,72] the intent of the Computer Security Act of 1987 in explicitly assigning this task to NIST was to have non-military cryptography under civilian control and prevent NSA from venturing into the private sector. However, in practice NSA has had the resources and expertise to dominate NIST, and NIST has rarely played a significant independent role. In any case, whatever the intent of the Act was, in the aftermath NSA took on an increasing role in the civilian world.

The second basic reason for the emergence of a "kinder, friendlier NSA" (in the words of a top NSA official [109]) was the end of the Cold War. During the decade between the collapse of the Soviet Union and the 9/11/2001 attacks, the U.S. did not have any obvious external enemy. As a result the companies, government agencies, and even academic disciplines (such as Russian area studies) that had come to prominence during the Cold War had to re-tool or else risk losing their relevance and their funding. Thus, it was strongly in NSA's interest to show that it had a role to play in developing technology that could protect the commercial world and the public at large from all sorts of threats to their communications.

Whatever the reasons for NSA's new focus, the timing could not have been better for elliptic curve cryptography, which by the mid-1990s was locked in an increasingly nasty competition with RSA. From a commercial standpoint RSA had a tremendous advantage over the Canadian company Certicom, which was the main promoter of ECC. RSA was well established, had name recognition and had the lion's share of the public key cryptography market. On the other hand, the advent of the number field sieve forced RSA to use longer and longer keys. People who understood the math behind the two systems could see that over time RSA would be inferior to ECC in constrained environments where memory and bandwidth are very limited.

In the early 1990s there was a controversy over a proposed Digital Signature Standard that to some extent presaged the role that NSA would later play in the debate over ECC. NIST proposed a protocol for digital signatures that had been developed by NSA and closely resembled an earlier method invented by C. Schnorr. In these systems the security of signatures was based on the discrete log problem in a finite field. This choice was a direct challenge to the predominance of factorization-based cryptography, and it was bitterly opposed by RSA. Although the Digital Signature Standard — which was approved for commercial use in 1994 — was not based on elliptic curves, it signaled a dissatisfaction with RSA technology within NSA.

The technical people in NSA had been attracted to elliptic curve cryptography since the 1980s. But the first time these views became known to the outside world occurred at a meeting of the American National Standards Institute (ANSI) in December 1995. Meetings of standards bodies typically include industry representatives who have little mathematical background and so are easily manipulated by scare tactics. At the meeting in question, the RSA people were casting doubt on the safety of ECC-based protocols. As the heated debate continued, one of the NSA representatives left to make a phone call. He then returned to the meeting and announced that NSA believed that ECC had sufficient security to be used for communications among all U.S. government agencies, including the Federal Reserve. People were stunned. Normally the NSA representatives at standards meetings would sit quietly and hardly say a word. No one had expected such a direct and unambiguous statement from NSA — a statement that tipped the scales at ANSI in favor of ECC.

At Crypto '97 J. Solinas gave the first paper ever presented publicly at a cryptography meeting by an NSA member. It contained a procedure he had developed (see [103]) for greatly improved efficiency of ECC using anomalous binary curves (see Section 11.1). NSA's support for ECC became more and more obvious over the years. In 2003 it licensed 26 ECC-related patents from Certicom for US$25 million, and in 2005 it posted the paper "The Case for Elliptic Curve Cryptography" on its website (see Section 1).

The influence of NSA, which is part of the U.S. Department of Defense, on the RSA versus ECC debate is an example of a general phenomenon that has been documented by sociologists and historians of technology. For example, Braun and MacDonald (see [13] and [74, p. 16]) have shown that military support played an essential role in the history of the microchip, especially in the early years of semiconductor electronics when the commercial world viewed solid-state devices as inferior to the earlier valve technology. According to MacKenzie and Wajcman [74, p. 15], "Military interest in new technology has often been crucial in overcoming what might otherwise have been insuperable economic barriers to its development and adoption." In a sense, NSA served as a counterweight to RSA's market advantage, and in this way helped level the playing field between RSA and ECC.

8. XTR vs. ECC

At Crypto 2000 A. Lenstra and Verheul [71] proposed a new type of cryptosystem called XTR. They choose a prime p such that $p^2 - p + 1$ has a large prime factor n, and they let G be the subgroup $\mu_n \subset \mathbb{F}_{p^6}^*$ of order n in the multiplicative group of the degree-6 extension of $\mathbb{F}_p$. They have a way (which they call the "trace representation") of writing elements of G as efficiently as if they lived in the subfield $\mathbb{F}_{p^2}$ (which, of course, they do not). But to find discrete logs in G by index calculus methods one would have to work in the field of p^6 elements, not p^2 elements. Lenstra and Verheul explained the advantages of their system:

> XTR achieves security similar to RSA for much smaller key sizes than RSA. Although ECC key sizes can be somewhat further reduced than XTR key sizes, in many circumstances… key sizes of ECC and XTR will be comparable… XTR may be regarded as the best of two worlds, RSA and ECC.

They also claimed a security advantage over ECC:

> However, XTR is not affected by the uncertainty still marring ECC security.… Also, compared to ECC, the mathematics underlying XTR is straightforward, thus avoiding two common ECC-pitfalls:

> ascertaining that unfortunate parameter choices are avoided that happen to render the system less secure, and keeping abreast of… newly obtained results.

At the Crypto 2000 Rump Session, Menezes and Vanstone responded to the claims for XTR by pointing out that the XTR group is precisely the group to which a certain supersingular curve E defined over $\mathbb{F}_{p^2}$ is isomorphic using the Weil pairing. Since the appearance of the Weil pairing attack [76], such curves were generally avoided in ECC. Isn't it risky to start using a group that is so intimately related to a weak case of ECC? Of course, the Weil embedding transports the discrete log problem on E to the XTR group, not vice-versa. This means that the problem on E reduces to the problem on the XTR group, not that the two problems are equivalent. However, Menezes and Vanstone asked whether there might be an efficiently computable map in the other direction that inverted the map coming from the Weil pairing. If so, then that would show that the discrete log problems on the two groups are exactly equivalent.

8.1. Verheul's theorem

Lenstra and Verheul were bothered by the suggestion that their system was equivalent in security to that of supersingular ECC. In 2000 supersingular elliptic curves were still viewed as too weak for cryptography. Verheul took up the challenge of Menezes–Vanstone, and was able to prove a striking theorem [105,106]: If an efficiently computable isomorphism existed from the XTR group to the curve, then the Diffie–Hellman problem would be easy in both groups. Since that was unlikely, he concluded that the map goes only one way.

Verheul's own interpretation of his theorem — stated boldly in the title to [105,106] — was that it provided evidence that XTR has strictly greater security (in the sense of hardness of the discrete log problem) than the corresponding supersingular ECC. However, in the first place, that conclusion does not follow logically from the theorem. There is in fact no evidence that there is any method of solving the discrete log problem on the supersingular curve that is faster than embedding it in the XTR group and then solving the discrete log problem in that group. Just because a possible avenue to proving equivalence of two problems — namely, constructing an efficient isomorphism in both directions — has been shown to be unlikely, that does not mean that in practice the problems are not equivalent. For example, on curves of high embedding degree the so-called decision Diffie–Hellman problem (the problem, given g, g^x, g^y, of determining whether or not a fourth group element is equal to g^{xy}) is believed to be solvable only if one can find the discrete log of g^x or g^y. However, it is highly unlikely that anyone will be able to *prove* by a reduction that the decision Diffie–Hellman problem is equivalent to the discrete log problem in such a group.

If someone really believes, along with Verheul, that the supersingular curve E over $\mathbb{F}_{p^2}$ might be even less secure than the subgroup of $\mathbb{F}_{p^6}^*$ into which it embeds, then presumably the same would apply to all supersingular curves. That would have dire implications for much of pairing-based cryptography. However, Verheul's theorem was presented just a few months before the first major pairing-based protocols were announced. So at the time no one was worried about this implication of Verheul's claim in the title of his papers [105,106].

And what if a map in the reverse direction *could* be constructed? It turns out that Verheul's theorem can be generalized (see [30,84]) to all supersingular curves and all finite fields. Thus, the construction of such a map would imply that the Diffie–Hellman problem is easy in all finite fields and all supersingular elliptic curves. We do not mean to suggest that this is likely — we only want to illustrate the point that Verheul's theorem lends itself to multiple interpretations.

8.2. Skepticism's last gasp

Despite the disparaging comments about ECC by the promoters of XTR, skepticism about elliptic curves was very much on the decline by the start of the new millennium. Industrial standards bodies had endorsed certain forms of ECC (see, e.g., [3,86]), and "ECC Central" had been removed from the RSA website.

This is not to say that no one in recent years has expressed doubts about ECC. Occasionally a writer on cryptography might object to the increasing acceptance of elliptic curve technology. For example, Bruce Schneier, the author of a best-selling guide to cryptography, has a popular blog that in 2005 took comments on the NSA paper "The Case for Elliptic Curve Cryptography" (see Section 1). In response to a blogger who wrote, "But ECC was less researched than the others [sic] algorithms!" Schneier posted the comment: "I agree with you, not the NSA."

9. The dramatic entry of pairing-based cryptography

Starting in 2001, pairing-based cryptosystems were proposed by Dan Boneh, Matt Franklin, and others. Although some of the ideas had been around for a couple of years (see, for example, [48,90]), their tremendous potential had not been realized before.

The basic idea is that the Weil or Tate pairing on elliptic curves allows certain cryptographic functions to be performed more efficiently than ever before, provided that one works with elliptic curves where the pairing can be efficiently computed, i.e., curves of low embedding degree. Such curves have the "Diffie–Hellman gap" property, which means that the Diffie–Hellman problem is thought to be difficult, whereas the decision Diffie–Hellman problem (see Section 8.1) can be easily solved using the pairing.

One of the first uses of pairing-based cryptography was the elegant solution by Boneh and Franklin [10] to an old question of Shamir [94], who had asked whether an efficient encryption scheme could be devised in which a user's public key would be just her identity (e.g., e-mail address). Such a system is called *identity-based encryption*. Another early application (see below) was to obtain short signatures.

9.1. Boneh–Lynn–Shacham signatures

We shall describe the pairing-based signature scheme of Boneh, Lynn and Shacham [11] in the setting of the supersingular elliptic curve

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

defined over $\mathbb{F}_p$, $p \equiv 3 \pmod 4$. This curve E has group order $p+1$ and embedding degree 2; suppose that p is chosen so that $n = (p+1)/4$ is prime. Let P be a fixed and publicly known generator of the subgroup $G \subset E(\mathbb{F}_p)$ of prime order n. We define what is called a *distortion* map on the $\mathbb{F}_{p^2}$-points of E as follows:

$$Q = (u, v) \mapsto \widetilde{Q} = (-u, iv), \quad \text{where } i^2 = -1,\ i \in \mathbb{F}_{p^2}.$$

This is the reduction mod p of the usual complex multiplication on the $\mathbb{Q}$-curve with Eq. (1). It gives an isomorphism from $G \subset E(\mathbb{F}_p)$ to a "distorted group" $\widetilde{G} \subset E(\mathbb{F}_{p^2})$ that, together with G, generates all of $E[n]$. Note that non-degeneracy of the Weil pairing implies that the pairing of a non-trivial element of G with a non-trivial element of $\widetilde{G}$ gives a non-trivial nth root of unity.

Each user Alice chooses a random integer x mod n, which is her secret key, and computes the point $Q = xP$, which is her public key. Suppose that Alice wants to sign a message to Bob that has hash value H, which we suppose is an $\mathbb{F}_p$-point of E. All she does is compute $S = xH$, which is her signature for the message. When Bob receives the message and the signature he computes the hash value H and then the two pairing values

$$(H, \widetilde{Q}) \quad \text{and} \quad (S, \widetilde{P}).$$

If Alice created the signature correctly, then the two values must be equal, because they are both equal to

$$(H, \widetilde{P})^x.$$

Bob has confidence that only Alice could have signed the message, because only she would have been able to generate the point S whose discrete log to the base H is the same as the discrete log of Q to the base P.

Not only is the Boneh–Lynn–Shacham signature shorter in bitlength (if implemented with a suitably-chosen elliptic curve) and easier to describe than the Elliptic Curve Digital Signature Algorithm (see Section 3.3), but, unlike ECDSA, it uses properties of elliptic curves in an essential way and does not have any analogue in the simpler group $\mathbb{F}_q^*$.

9.2. Selection of curves

There are two ways to select a curve of low embedding degree k. One can choose a supersingular curve, for which $k \leqslant 6$. Supersingular curves have the advantage that there is a computable distortion map that can be used to construct protocols (see Section 9.1).

However, one often wants $k \geqslant 6$ to be large enough so that the time required to find discrete logs using index calculus in $\mathbb{F}_{q^k}$ is comparable to the time required to find discrete logs directly in the group G using a squareroot attack. At present $k = 6$ is a reasonable choice, but with increased computing power the optimal choice of k will soon be larger. A supersingular curve with $k = 6$ exists only in characteristic 3, and there is no supersingular curve with $k > 6$. Thus, implementers might want to use ordinary curves of low embedding degree. Such curves are rare, and the only way known to construct them is to use the so-called *CM-method*.

Let E be an ordinary elliptic curve defined over $\mathbb{F}_p$ with *trace* t, which means that $\#E(\mathbb{F}_p) = p + 1 - t$. We want $p + 1 - t$ to be a prime (or a prime times a very small cofactor); we want $p^k \equiv 1 \pmod{p+1-t}$ (note that this is equivalent to $(t-1)^k \equiv 1 \pmod{p+1-t}$); and we want the discriminant $t^2 - 4p$ to have small squarefree part d, in which case a curve over a number field can be found with complex multiplication by $\mathbb{Q}(\sqrt{d})$ that reduces modulo a prime lying over p to a curve with the desired properties. The idea of the construction of ordinary curves with low k by the CM-method is to find a family of integers (p, t) parameterized by an integer z such that the second and third of these conditions hold (and there is a reasonable probability that p is prime and $p + 1 - t$ has a large prime factor). The first results of this type were found for $k = 3, 4, 6$ in [83]; in the case $k = 6$ they set $p = 4z^2 + 1$ and $t = 1 \pm 2z$. Subsequently other authors showed how to construct ordinary curves for certain embedding degrees $k > 6$.

9.3. Like a knife through butter

Pairing-based cryptography received near-universal acceptance and acclaim from the beginning. Unlike traditional ECC, it did not pass through a period of several years of skepticism and resistance. We find this especially striking because the curves used in this type of cryptography are precisely the ones that were shunned in ECC for many years after the discovery of the Weil pairing attack [76] and were still being disparaged as late as Eurocrypt 2001 by Verheul [105].

This paradoxical turn of events has several possible explanations. In the first place, it is hard not to be attracted by the sheer elegance of some of the constructions in pairing-based cryptography. Note, for example, how much simpler the Boneh–Lynn–Shacham signature is to describe (see Section 9.1) than the ECDSA was (see Section 3.3).

In the second place, the timing was propitious. The first major pairing-based protocols were being promoted in the years right after traditional ECC had won acceptance and the once-bitter rivalry between RSA and ECC had subsided. Basically, most of the earlier critics of ECC had thrown in the towel — starting in the late 1990s the RSA software toolkit even included a version of ECC.

In the history of technology it often happens that after a period of intense debate (what the sociologists Kline and Pinch [50] call *interpretative flexibility*) a consensus emerges to admit the "new kid on the block" into full membership in the club. At that point most people see no benefit in standing in the way of adopting the newer technology; rather, it seems to be in everyone's interest to incorporate it into their theories and products. This process is known as *closure*. As Kline and Pinch explain,

> Interpretative flexibility, however, does not continue forever. 'Closure' and stabilization occur, such that some artifacts [i.e., inventions] appear to have fewer problems and become increasingly the dominant form of the technology. This, it should be noted, may not result in all rivals vanishing, and often two very different technologies can exist side by side (for example, jet planes and propeller planes). Also this process of closure and stabilization need not be final. New problems can emerge and interpretative flexibility may reappear [50, pp. 113–114].

A third explanation for the immediate acceptance of pairing-based cryptography is that by 2001 the viewpoint that papers proposing new protocols must always include a "proof of security" had become pervasive, especially on cryptography conference program committees. Almost all papers proposing pairing-based protocols included such "proofs," and they served to reassure people about the security of the systems.

This is not the place to repeat the critique of "provable security" in the series of papers [60,59,61]. Suffice it to say that the guarantees given by such proofs, even when the proofs are mathematically correct, are very conditional and contingent. In recent years what has often happened is that, whether or not readers fully understand the proof, they are mesmerized by it and are willing to put aside any doubts they might have had. Most likely this effect was at work in causing virtually universal and unquestioning acceptance of pairing-based cryptography in the research community.

What a "security proof" — or, as we prefer to say, a *reductionist security argument* [60] — actually does show is that an adversary cannot succeed in mounting a certain category of attack unless a certain underlying mathematical problem is tractable. What is peculiar in the case of pairing-based cryptography is that the underlying mathematical problem is often a very contrived one, of the sort that hardly any mathematician would recognize as natural, let alone want to study. Nevertheless, it has become customary to regard a conditional result related to such a problem as a type of guarantee of security.

For example, the underlying problem in [8] is called the *m-strong Diffie–Hellman problem* in a group G of prime order n. Let g be a generator of G, and let x denote an unknown integer mod n. Given the $m+1$ group elements $g, g^x, g^{x^2}, \ldots, g^{x^m}$, the m-strong Diffie–Hellman problem asks one to find a pair (c, h) (where c is a non-zero integer mod n and h is a group element) such that $h^{x+c} = g$. At first it seemed that in practice this problem would prove to be as hard as finding discrete logs — in other words, in a generic group G no algorithm would be faster than $\sqrt{n}$. However, at Eurocrypt 2006 Cheon [17], using the same method that had been described earlier in a different context by Brown and Gallant [16], showed that if $n-1$ has a factor $m_0 \leqslant m$ of size a little less than $n^{1/3}$, then the m-strong problem can be solved in roughly $n^{1/3}$ operations. So the underlying problem used in the security proof turned out to be weaker than expected.

Some of the other underlying problems that occur in reductionist security arguments for pairing-based systems are even more ornate and contrived than the m-strong Diffie–Hellman problem (see [62] for some examples). Nevertheless, few people have expressed skepticism regarding the true security of the "provably secure" pairing-based protocols.

We wish to stress that we have no reason to believe that any pairing-based protocol is actually insecure. Our purpose in discussing this issue is not to urge people to avoid such cryptosystems, but rather to raise the intriguing question of why there have been hardly any skeptics in the research community.

A final reason for the rapid acceptance of pairing-based cryptography is that it was not perceived as a threat either to important commercial interests or to established traditions of cryptographic research. On the contrary, the idea had an immediate appeal both for practical reasons — it provided the opportunity to improve functionality — and for intellectual reasons as well — it used some clever ideas in both mathematics and protocol design.

Moreover, the timing could not have been better for the cryptography profession, which was having some difficulty coming up with a lot of nice problems for research projects. A large number of people with math or computer science backgrounds had entered the field and were faced with the challenge — especially, but not exclusively, in academia — to "publish or perish." In addition, there had been a proliferation of cryptography conferences, all of which hoped to attract cutting-edge research papers. An increasing concern of program committees was that it was unrealistic to expect the amount of high-quality research to have increased at the same rate as the number of conferences. Against this backdrop the entrance onto the stage of pairing-based cryptography was like a godsend.

As mentioned before, pairing-based cryptography started at the height of influence of the notion of "provable security," and almost all papers in the area included reductionist arguments for the security of the proposed protocol. Interestingly, this tradition of always including a security proof led to even more possibilities for research projects, thanks to the controversy surrounding the so-called "random oracle model." (The *random oracle model* basically allows one to make arguments for the security of a protocol under the plausible assumption that hash function values are indistinguishable from random bitstrings.) Leading theoreticians — apparently inspired in part by the Biblical story of the Bronze Serpent (see [35, pp. 10–11]) — had decided that the random oracle assumption that is used in many security proofs is suspect, and cryptographers should try to design protocols that have security proofs that avoid the use of this assumption. As a result it became common first to develop a protocol with nice properties that has a proof of security in the random oracle model, and then to publish a modified version, usually with slightly less desirable properties but with a security proof in a "standard" model. This was an important advance for the profession, since in one fell swoop it increased the number of papers that could be published on provably secure protocols from N to $2N$.

10. A chink in the golden shield: Index calculus again rears its head

10.1. Weil descent

In the late 1990s Gerhard Frey had the idea of attacking the discrete log problem on an elliptic curve defined over $\mathbb{F}_{q^m}$ by transporting it to the jacobian group of a curve over the smaller field $\mathbb{F}_q$, where it could be solved using index calculus in a way similar to [1]. This program was first carried out in certain cases by Gaudry, Hess, and Smart [33], and their method has been generalized by Hess [40].

In some very special situations it has been possible to transport the discrete log problem on the elliptic curve defined over $\mathbb{F}_{q^m}$ to the corresponding problem on the jacobian group of a genus-m curve defined over $\mathbb{F}_q$ (in that case both groups have the same order $\approx q^m$). In other special cases the genus of the curve is considerably larger than m, but the resulting algorithm is still faster-than-squareroot as a function of q^m.

Weil descent does not apply over prime fields, and in the range of interest in cryptography it seems not to apply to curves considered over prime degree extensions of $\mathbb{F}_2$ (see [77]). Its main successes so far have been for curves defined over the fields $\mathbb{F}_{2^f}$ when f is divisible by 3, 5, 6, 7, or 8. For example, in [79] one of the classes of curves to which the Weil descent methods in [33] were shown to be applicable is the set of all elliptic curves E defined over $\mathbb{F}_{2^{5\ell}}$ (with ℓ prime) and not over a proper subfield. In theory it might be possible to transport the discrete log problem on E to the jacobian of a genus-5 curve over $\mathbb{F}_{2^\ell}$, for which there would be an index calculus algorithm requiring $O(2^{1.6\ell})$ operations. In practice, though, the curves that came out of the Weil descent had genus 15 or 16, resulting in an algorithm with running time roughly $2^{2\ell}$. This was still significantly better than Pollard-rho, which takes time $2^{2.5\ell+\epsilon}$.

10.2. Other potentially weak fields for ECC

In [32] Gaudry used index calculus methods directly on elliptic curves defined over $\mathbb{F}_{q^m}$ with $m > 1$. For a factor base he used the set of points whose x-coordinate lies in $\mathbb{F}_q$. He performed the crucial step of expressing a randomly generated point in terms of the factor base by means of summation polynomials, a concept introduced by Semaev [93]. For fixed m the running time of Gaudry's algorithm was $O(q^{2-2/m})$, so for $m \geqslant 3$ this gave a faster-than-squareroot attack.

In addition, Diem [21] proved that Gaudry's algorithm yields a subexponential algorithm when the size of the field $\mathbb{F}_{q^m}$ increases in such a way that m^2 is of order $\log q$.

Fortunately for ECC, by the late 1990s implementers had largely restricted themselves to either prime fields or prime degree extensions of $\mathbb{F}_2$. Prime fields and binary fields have traditionally been the easiest finite fields to use in most applications. The choice of prime degree m of $\mathbb{F}_{2^m}$ was partly dictated by the desire to allow the use of anomalous binary curves (see Section 11.1), which must be taken over such an extension if one wants the group order to be divisible by a prime of roughly m bits. Thus, when NIST decided to recommend one random curve and one anomalous binary curve for each recommended binary field, it was natural to choose m to be prime values for which the order of one of the curves (2) or (3) (see below) is equal to twice a prime or four times a prime [86]. In any case, because of this preference for prime fields and prime degree binary fields, the faster-than-squareroot attacks described in this section, none of which applied to curves over such fields, had no impact on real-world implementations.

11. A tale of two standards: Brainpool vs. Voltage

In this section we compare two recent recommendations concerning which elliptic curves to use in ECC. One comes from Brainpool, a European consortium of companies and government agencies led by Bundesamt für Sicherheit in der Informationstechnik (BSI, the German equivalent of NSA). The other one [12] comes from an American company called Voltage, which presented it at a NIST workshop on pairing-based protocols (see [75]). What is interesting to us is the extent to which these recommendations contradict one another.

The Brainpool draft [73] explicitly excludes all elliptic curves of low embedding degree (hence all supersingular curves) and all ordinary elliptic curves whose CM-field has low class number (hence all curves constructed by the CM-method). In particular, Brainpool rules out all curves (supersingular or ordinary) used in pairing-based cryptography.

Both of these Brainpool requirements are given in a rather extreme form. The embedding degree must be greater than $(q-1)/100$. If, for example, q has 160 bits (the smallest size they allow), then they are saying that it is a bad idea to use an elliptic curve group that embeds in a finite field of the form $\mathbb{F}_{q^k}$ with k a 150-bit integer (i.e., $k \approx (q-1)/1000$). Note that in such a humongous field the fastest algorithms known for the finite field discrete log would take time greater than $\exp(10^{15})$ — roughly 1 followed by four hundred trillion zeros. Brainpool certainly seems to want to err on the side of caution! They also require that when the elliptic curve lifts to an elliptic curve over a number field that has complex multiplication, that number field must have degree greater than ten million.

In contrast, the Voltage recommendation [75] states under "security considerations" that

> The conservative choice for implementing a pairing-based algorithm is to use a supersingular curve.

The elliptic curve they recommend using is the curve

$$y^2 = x^3 + b$$

over a prime field $\mathbb{F}_p$ with $p \equiv -1 \pmod{12}$. This curve has $p+1$ points, embedding degree 2, and complex multiplication by the ring $\mathbb{Z}[\zeta]$, $\zeta = \exp(2\pi i/3)$.

Moreover, the Voltage curve has far more structure than most curves because it is supersingular. Namely, supersingular curves have a gigantic endomorphism ring — a quaternion algebra that includes imaginary quadratic rings as a small part. For the Voltage curve E the endomorphism ring of $E(\overline{\mathbb{F}}_p)$ is the following quaternion algebra:

$$\mathbb{Z} + \mathbb{Z}\zeta + \mathbb{Z}\phi + \mathbb{Z}\phi\zeta,$$

where ϕ is the Frobenius endomorphism $(x, y) \mapsto (x^p, y^p)$, which satisfies the relation $\phi^2 = -p$. Since $\zeta^p = \zeta^{-1}$, the commutation relation here is: $\zeta\phi = \phi\zeta^{-1} = -\phi - \phi\zeta$.

No one has ever been able to use this vast stable of endomorphisms to mount an attack on the discrete log problem on a supersingular curve. So there is no real evidence that the Voltage curve is weak. But its extensive special properties would certainly give heartburn to the Brainpool cryptographers.

What is Voltage's rationale for referring to its curve as the "conservative choice"? In the context of pairing-based cryptography, if you want to use an ordinary curve rather than a supersingular curve, you must use a very special version of the CM-method to construct your curve (see Section 9.2). As explained in [75]: "With ordinary curves, additional structure is needed to get a low embedding degree." As mentioned in [58], the type of special values of p that are used might cause the discrete log problem in $\mathbb{F}_{p^k}$ (where k is the embedding degree) to be vulnerable to a version of the *special* number field sieve rather than the *general* number field sieve. Indeed, Schirokauer [91] has shown that this is true in a few cases. However, in most cases it is far from clear that the "additional structure" in the choice of the prime p or the group order n could ever be utilized by an attacker. So whether the "conservative choice" is to use supersingular curves or to generate ordinary curves of low embedding degree — or perhaps (if one is a follower of Brainpool) to avoid pairing-based cryptography altogether — is anyone's guess.

11.1. Special or random selection of parameters?

A general philosophy one often encounters in cryptography is that whenever possible parameters should be chosen by some random process. If a special choice is made to increase efficiency, there is always the risk that the same property that made the choice so attractive will also lead to vulnerability to an unanticipated attack.

In the case of elliptic curve cryptography one of the arguments for its superiority over RSA was the tremendous variety of curves to choose from. This means that there are several opportunities to introduce randomness into parameter selection. One can make a random choice of prime field $\mathbb{F}_p$, for instance, followed by random choices of the coefficients in the Weierstrass equation of the curve. This is, in fact, essentially what Brainpool recommends.

In 1985 R. Schoof [92] devised the first polynomial-time algorithm to determine the group order for an arbitrary elliptic curve. His method was greatly improved upon by Atkin, Morain, Elkies, and others. Much of this work was based on isogenies of elliptic curves; this was the first — but not the last (see Section 11.2) — cryptographic application of isogenies.

The availability of efficient point-counting algorithms for random elliptic curves means that there is no practical reason not to use them. On the other hand, certain special curves have significant efficiency advantages. For example, over characteristic-2 fields one can save a lot of time computing point multiples by using the so-called *anomalous binary curves* defined over $\mathbb{F}_2$

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

and

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

(see [56,103]). In elliptic curve cryptography one would use a prime order subgroup of the group of points defined over an extension of $\mathbb{F}_2$.

The conventional wisdom is that there is a trade-off. If you want long-term security, you must be willing to sacrifice a little bit of efficiency and generate your parameters in a random way. On the other hand, if a special choice of parameters allows for greater efficiency and if there are no known attacks that utilize their special properties, and if you are willing to risk the possibility that such attacks will be found some day, then by all means use, for example, anomalous binary curves.

This point of view seems logical, and it is uncontroversial among cryptographers. However, under certain circumstances it may be wrong. In particular, it is conceivable that Brainpool's super-cautious recommendations might cause one to choose curves that are *less secure* than some CM-type curves might be. In other words, random curves might be riskier than special curves.

Before explaining how this is possible, we would like to make a remark about the notion of special versus generic curves. As we saw in Section 5.1, a word such as "complexity" might have a different meaning in cryptography than in traditional mathematics; conceptual complexity and computational complexity are two quite different things. The same goes for the term “special curve.”

Example 1. In the study of algebraic curves one normally regards hyperelliptic curves as a very special subclass. For $g \geqslant 3$ there are far fewer hyperelliptic than non-hyperelliptic curves in the sense that the hyperelliptic curves correspond to a submanifold of codimension $g-2$ in the moduli space of all genus-g curves. That is, for $g \geqslant 3$ there are roughly $1/q^{g-2}$ times as many hyperelliptic curves as non-hyperelliptic curves over $\mathbb{F}_q$.

Yet Diem and Thomé [23] found an index calculus attack on the discrete log problem in the jacobian group of a genus-3 non-hyperelliptic curve over $\mathbb{F}_q$ that has running time of order only $q^{1+\epsilon}$. In [20] Diem generalized this algorithm to all “sufficiently general” non-hyperelliptic curves of arbitrary genus $g \geqslant 3$ with running time $q^{2-2/(g-1)+\epsilon}$. This algorithm is substantially faster than the fastest known algorithm for discrete logs in the jacobian group of a hyperelliptic curve (see [22]), which takes time $q^{2-(2/g)+\epsilon}$. To put it another way, over a fixed field $\mathbb{F}_q$ the discrete log problem on the jacobian of a genus-g hyperelliptic curve has the same computational complexity (in the sense of the best available algorithms) as the discrete log problem on the (much larger) jacobian of a genus-$(g+1)$ non-hyperelliptic curve. It turned out that a certain way in which a generic non-hyperelliptic curve can be represented as a plane curve allows for a particularly efficient generation of relations among divisor classes. Thus, to the best of our current knowledge, it is the non-hyperelliptic curves and not the hyperelliptic curves whose discrete log problems have a special vulnerability to index calculus.[4]

There are various scenarios in which someone (say, Alice) who chose to use ECC with a special curve might end up better off than someone else (say, Bob) who chose a random curve. Our first example is a little removed from practice because we use extension fields of composite degree, whereas real-world implementations of ECC generally are over either a prime field $\mathbb{F}_p$ or an extension of $\mathbb{F}_2$ of prime degree.

Example 2. Suppose that Alice wants to use an anomalous $\mathbb{F}_2$-curve (2) or (3) over a field extension of degree 5ℓ with ℓ prime. She knows, of course, that, because of the large subgroup of order $\approx 2^\ell$ consisting of the $\mathbb{F}_{2^\ell}$-points of the curve, the largest prime order subgroup she can hope to get has order roughly $2^{4\ell}$. (It will actually be a little smaller because there is also a subgroup of $\mathbb{F}_{2^5}$-points.) This means that the Pollard-rho algorithm will take time approximately $2^{2\ell}$. (There will also be a slight speed-up from grouping together conjugate points as explained in [31,108], but this is only by $\sqrt{\ell}$, and we are using rough asymptotic running times here.) So if Alice wants k bits of security she will have to use ℓ of order $k/2$. Nevertheless, she still wants to use an anomalous curve because she feels that its efficiency advantage is great enough to compensate for the need to choose ℓ a little larger.

Meanwhile, Bob thinks that Alice is being unwise, because if he uses a random curve over the same type of field $\mathbb{F}_{2^{5\ell}}$ with ℓ prime, he can find a curve whose group order is twice a prime, in which case Pollard-rho will take time roughly $2^{2.5\ell}$. That means that he can get the same k bits of security as Alice with ℓ equal only to $0.4k$.

Bob’s reasoning made perfect sense throughout the 1990s. However, the study of Weil descent in [79] showed that the discrete log problem on a random curve over $\mathbb{F}_{2^{5\ell}}$ can be reduced to the corresponding problem in the jacobian of a genus-15 or genus-16 curve over $\mathbb{F}_{2^\ell}$, which, in turn, can be solved in time roughly $2^{2\ell}$. This is asymptotically the same as the time for Pollard-rho on Alice’s special curve. Thus, Bob took a bad gamble when he decided to use a random curve with a lower value of ℓ than Alice’s. He gets only $0.8k$ bits of security, not the k bits he thought he would get.

[4] An algorithm of Smith [100] allows one sometimes to transport the discrete logarithm problem on the jacobian of a genus-3 hyperelliptic curve to a non-hyperelliptic jacobian. For large $\mathbb{F}_q$ the procedure works for approximately 18% of all hyperelliptic curves. Smith’s algorithm applies only to curves of genus 3 and fields of characteristic > 3.

In order to give two more examples of possible security disadvantages of randomness in ECC, we have to talk about isogenies.

11.2. Isogenies and endomorphism rings

We shall give a brief overview of isogenies between elliptic curves. For proofs and details see [96, 107]. Let E_1 and E_2 be defined over $\mathbb{F}_q$. An isogeny $\psi : E_1 \to E_2$ defined over $\mathbb{F}_q$ is a non-constant rational map defined over $\mathbb{F}_q$ that maps ∞ to ∞; its *degree* is its degree as a rational map. If ψ is a *separable* isogeny, then the kernel of ψ is a subgroup of E_1 of order $\deg\psi$.

Any isogeny $\psi : E_1 \to E_2$ has a *dual isogeny* $\widehat{\psi} : E_2 \to E_1$ such that the composition $\psi \circ \widehat{\psi}$ is the endomorphism of multiplication by $\deg\psi$. We say that E_1 and E_2 are *isogenous* over $\mathbb{F}_q$ if there exists an isogeny from one to the other that is defined over $\mathbb{F}_q$. A theorem of Tate states that E_1 and E_2 are isogenous over $\mathbb{F}_q$ if and only if $\#E_1(\mathbb{F}_q) = \#E_2(\mathbb{F}_q)$.

For example, suppose that ℓ is a prime not equal to the characteristic of the field, and C is a subgroup of E_1 of order ℓ that is defined over $\mathbb{F}_q$. (Recall that this means that the subgroup C is fixed by the Frobenius map $\phi : (x, y) \mapsto (x^q, y^q)$, which can be true even if the individual points of C are not in $\mathbb{F}_q$.) For every such group C there is a degree-ℓ isogeny from E_1 to a curve E_2 defined over $\mathbb{F}_q$ whose kernel is C.

The *modular polynomial* $\Phi_\ell[X, Y] \in \mathbb{Z}[X, Y]$ has the property that if we let $\overline{\Phi}_\ell$ denote the reduction mod p and set Y equal to the j-invariant of E_1, then the $\ell + 1$ roots of $\overline{\Phi}_\ell[X, j] \in \mathbb{F}_q[X]$ are the j-invariants of all of the curves E_2 that are ℓ-isogenous to E_1 over the algebraic closure $\overline{\mathbb{F}}_q$. Each such isogeny corresponds to one of the $\ell + 1$ subgroups of order ℓ in the group of ℓ^2 points of order ℓ on $E_1(\overline{\mathbb{F}}_q)$. An ℓ-isogeny from a given curve can be quickly constructed if ℓ is small; however, in general the best available algorithm [65] has running time roughly ℓ^3, so for large ℓ the construction is not feasible.[5]

Let E be an elliptic curve defined over $\mathbb{F}_q$, $q = p^f$, and let $t = q + 1 - \#E(\mathbb{F}_q)$ denote its trace. An *endomorphism* of E is an isogeny from E to itself that is defined over $\overline{\mathbb{F}}_q$. The endomorphisms form a ring denoted $\mathrm{End}(E)$. We consider the *ordinary* case when p does not divide t; in that case the elements of $\mathrm{End}(E)$ are all defined over $\mathbb{F}_q$. Let $\Delta = t^2 - 4q$ denote the *discriminant* of E. Then the *complex multiplication* (CM) field of E is $K = \mathbb{Q}(\sqrt{\Delta})$. We have $\Delta = c_0^2 d$, where $d < 0$ is the discriminant of the imaginary quadratic number field K. Let $\mathbb{Z}_K$ denote the ring of integers; then $\mathrm{End}(E) \subset \mathbb{Z}_K$ is an order in $\mathbb{Z}_K$. We let c denote the *conductor* of $\mathrm{End}(E)$, i.e., its index in $\mathbb{Z}_K$.

As before, let ϕ denote the Frobenius endomorphism, given by $(x, y) \mapsto (x^q, y^q)$. We regard ϕ as an element of $\mathbb{Z}_K$ of norm q, since its characteristic polynomial is $T^2 - tT + q = 0$. The subring $\mathbb{Z}[\phi]$ has index $c_0 = \sqrt{\Delta/d}$ in $\mathbb{Z}_K$, and $\mathrm{End}(E)$ is an order of $\mathbb{Z}_K$ that contains $\mathbb{Z}[\phi]$, and so its index in $\mathbb{Z}_K$ is a divisor c of c_0, $1 \leqslant c \leqslant c_0$.

The curves in the $\mathbb{F}_q$-isogeny class of a given E can be partitioned according to their endomorphism ring, i.e., into "endomorphism classes" within the isogeny class. The endomorphism rings are the orders in $\mathbb{Z}_K$ that contain $\mathbb{Z}[\phi]$, and they are in one-to-one correspondence with the divisors c of c_0. The number of isomorphism classes of curves in a given endomorphism class is equal to the class number h_c of the order, and this is approximately equal to ch_K, where h_K is the class number of the imaginary quadratic CM field K (for a more precise formula, see [29, p. 123]). The class number h_K satisfies $h_K \leqslant \frac{1}{\pi}\sqrt{|d|}\log|d|$, where d is the discriminant of K.

As mentioned before, the first use of isogenies was to develop improved methods of determining the group order of a random curve. More recently, isogenies have been used to investigate possible attacks on the elliptic curve discrete log problem. The idea is that if an isogeny can be computed between E_1 and E_2, then the discrete log problem on E_1 can be transported to the same problem on E_2. If it is feasible to construct isogenies between any two curves in a certain subset of the isogeny class, then the discrete log problem is *random self-reducible* in that subset of curves; this implies that the problem is equally difficult for all of those elliptic curves.

[5] In certain special cases there are faster methods [15], but they work only within a single L-conductor-gap class (this term is defined below) and so do not affect our argument.

Suppose that starting with a curve E_1 in an endomorphism class with conductor c one randomly chooses an ℓ-isogeny $E_1 \to E_2$, where ℓ is a small prime. Then either E_2 is in the same endomorphism class as E_1, or else it has conductor $c\ell$ or else c/ℓ. The last two possibilities can occur only if $\ell | c_0$. The main result of [47] (assuming the Generalized Riemann Hypothesis) is that a sequence of isogenies of prime degree $\ell < L = (\log q)^{2+\epsilon}$ (for fixed ϵ), $(\ell, c_0) = 1$, can be used to fan out randomly throughout the endomorphism class. (The precise statement is that the graph whose vertices are the isomorphism classes in a given endomorphism class with adjacency determined by these ℓ-isogenies is an *expander graph*.)

It is also possible to use isogenies to go efficiently between two endomorphism classes, but only if they have a small *conductor gap*, by which we mean the largest prime that divides one conductor and not the other. Thus, if L is a bound on the size of primes ℓ for which it is feasible to construct an ℓ-isogeny, it is natural to divide a given isogeny class into subsets that each consist of endomorphism classes with conductor gap $< L$. Thus, in each isogeny class we define the *L-conductor-gap class* of a curve E to be the set of all endomorphism classes having conductor gap $< L$ with $\mathrm{End}(E)$. The result of [47] extends to these larger classes; that is, the discrete log problem is random self-reducible in each L-conductor-gap class. That means that if an efficient algorithm were found to solve the discrete log problem in time T_1 in a constant proportion ϵ of all elliptic curves defined over $\mathbb{F}_q$ (we shall call them “weak” curves), then the discrete log could be found on any curve in the L-conductor-gap class in time roughly $T_1 + T_2/\epsilon$, where T_2 is the time required to construct an ℓ-isogeny, $\ell < L$. Here, of course, we are assuming that the property of being a weak curve for the discrete log algorithm is independent of isogeny class and endomorphism ring; and we are also assuming that the L-conductor-gap class contains $\gg 1/\epsilon$ curves.

Note that if $c_0 = 1$ (which is often the case) or if c_0 is L-smooth, then all $O(\sqrt{q})$ curves in the isogeny class are in the same L-conductor-gap class. If c_0 is divisible by just a single large prime r, then there are two such classes: a small set of isomorphism classes of curves whose endomorphism ring has conductor not divisible by r, and the “generic” isomorphism classes where the endomorphism ring has conductor a multiple of r.

It is the possibility of random isogeny walks through a conductor-gap class that under certain circumstances might make a generic curve less secure than a special curve. We discuss this in the next subsection.

11.3. More examples of potential weakness of random curves

Example 3. In V. Müller’s Table 6.2 of [85] the following is a choice of parameters suggested for ECC. Let $q = 2^{3\cdot 59}$, and let E be the curve defined over $\mathbb{F}_8$ by the equation

$$y^2 + xy = x^3 + x^2 + \gamma,$$

where $\gamma \in \mathbb{F}_8$ satisfies $\gamma^3 = \gamma^2 + 1$. This curve has 6 $\mathbb{F}_8$-points, and its group of $\mathbb{F}_{2^{177}}$-points has order 6 times the 175-bit prime

$$P_{175} = 31926990434706017882465563211521159723534715689440269.$$

Suppose that Alice, following the suggestion of Müller, chooses this curve E and extension field $\mathbb{F}_{2^{177}}$. She calculates that Pollard-rho (with the speed-up of $\sqrt{59}$ in [31,108]) would take roughly 2^{84} operations, i.e., the curve will give her 84 bits of security.

Bob, as usual, thinks that Alice is foolish for having chosen a curve with very special properties that allow the $\sqrt{59}$ speed-up and may leave her vulnerable to other attacks. He figures that if he chooses a random curve over the same field with group order twice a 176-bit prime, then he will get 88 bits of security rather than 84, and he will also be less vulnerable to unanticipated specialized attacks.

At least through the 1990s Bob’s reasoning would have appeared to be correct. But a closer examination using more recent research (see [79,78]) shows that Bob might not have nearly the security level that he thinks he has.

The complex multiplication field for Alice's curve is $\mathbb{Q}(\sqrt{-23})$, and the discriminant is $-23c_0^2$ with c_0 factoring as the product of two primes:

$$c_0 = 11681 \cdot 98766024850235972863.$$

In [78] Menezes and Teske found that a certain proportion — roughly $\epsilon = 2^{-58}$ — of all elliptic curves over $\mathbb{F}_{2^{3\cdot 59}}$ with group order $\equiv 2 \pmod 8$ are "weak" in the sense that Weil descent can be used to transport the discrete log problem to the corresponding problem on the jacobian of a genus-3 hyperelliptic curve over $\mathbb{F}_{2^{59}}$. At that point the discrete log problem can be solved in time roughly $2^{59\cdot 4/3} \approx 2^{79}$ [22].

In what follows we shall make the (plausible) assumption that the property of being a "weak" curve is independent of isogeny and endomorphism classes — in other words, that the expected number of weak curves in such a class is roughly equal to $\epsilon = 2^{-58}$ times its cardinality.

In Bob's case almost certainly the discriminant of his curve E is not divisible by the square of a large prime, and so it is possible to use isogenies to transport the discrete log problem on E along a "random walk" throughout its isogeny class. If his group order is $\equiv 2 \pmod 8$ (of which there is a 50% chance), then after approximately 2^{58} isogenies we will have transported the discrete log problem to a curve where it can be solved in time roughly 2^{79}. Each step in the "walk" takes time approximately 2^{17}, so that the reduction to the weak curve will take time $\approx 2^{75}$. In other words, Bob's random curve will have just 79 bits of security, not 88 bits as Bob thought and not even the 84 bits that Alice has.

In contrast, even if the result in [78] applied to curves in Alice's isogeny class (which it does not, since the number of points on her curve is $\equiv 6 \pmod 8$), she would still be safe because her curve's endomorphism ring has conductor 1 and is in a 2^{66}-conductor-gap class containing fewer than 2^{16} curves (the ones with conductor 1 or 11681). Under our assumption that the "weak" property is independent of isogeny or endomorphism class, it is highly unlikely — a probability of about 2^{-42} — that the discrete log problem on Alice's curve can be moved to a weak curve by a sequence of isogenies. In other words, what saves Alice from Bob's fate is precisely the very special nature of her curve.[6]

Most practical implementations of ECC in characteristic two use prime extension degrees, in which case Weil descent appears not to be useful (see [77]). However, it is not inconceivable that either a new version of Weil descent or some entirely different approach will some day lead to a faster-than-squareroot attack on a certain small (but non-negligible) proportion of elliptic curves defined over $\mathbb{F}_q$, where q is a prime power of 2. If we are using a curve over this field when this happens, we had better hope that our curve cannot be linked to a weak curve by means of isogenies.

Example 4. In 2000 in its Digital Signature Standard [86], NIST recommended ten elliptic curves over binary fields $\mathbb{F}_{2^\ell}$ with $\ell = 163$, 233, 283, 409, 571. For each ℓ they gave one random curve and one anomalous binary curve. The security level in the face of squareroot attacks is roughly $2^{\ell/2}$.[7] For instance, both curves over $\mathbb{F}_{2^{571}}$ should provide more than the 256 bits of security necessary to protect a high-security Advanced Encryption Standard (AES) private key.

This raises the question: Which of the two NIST curves over $\mathbb{F}_{2^{571}}$ is safer? The conventional wisdom would be that the random curve is the more conservative choice.

However, let us suppose that a certain very small — but not negligible — fraction ϵ of curves over this field could be attacked by some new faster-than-squareroot algorithm. We further suppose that the property of being a "weak" curve is independent of isogeny class or endomorphism class. In such

[6] In this discussion we are ignoring Gaudry's recent algorithm [32] for the discrete log on an elliptic curve over $\mathbb{F}_{q^m}$, $m \geqslant 3$ (see Section 10.2). However, in Section 4.4 of [32] Gaudry says that, based on experimental results, he expects that his algorithm for $m = 3$ will be faster than Pollard-rho only for $q > 2^{65}$. In our example $q = 2^{59}$, so at present we need not concern ourselves with Gaudry's algorithm. But of course improved implementations could result in that algorithm beating Pollard-rho in our case as well.

[7] The actual security achieved with anomalous binary curves is a little less because of the speed-up of the Pollard-rho attack by $\sqrt{\ell}$.

a situation if our curve is in a large L-conductor-gap class, then after a "random walk" consisting of $O(1/\epsilon)$ isogenies, an attacker can move the discrete log problem to a weak curve. A random curve is virtually certain to be in a large endomorphism class (since the discriminant of a random curve is very unlikely to be divisible by a large square). In particular, this is true of the random NIST curve over $\mathbb{F}_{2^{571}}$, whose discriminant is squarefree, i.e., all isogenous curves are in the same endomorphism class [47]. In contrast, the anomalous binary curve K-571 in [86] has discriminant $\Delta = -7c_0^2$, where the conductor c_0 is the product of a 22-bit prime and a 263-bit prime, and its endomorphism ring has conductor 1. For $L = 2^{262}$ the L-conductor-gap class of the curve K-571 has approximately 2^{22} curves, so if $\epsilon \ll 2^{-22}$, this curve is likely to be safer than a random curve under our assumptions.

Weil descent methods are not applicable to curves defined over a prime field. But suppose that we are worried about the possibility that some new approach to the discrete log problem will turn out to give faster-than-squareroot algorithms for a certain proportion of curves defined over $\mathbb{F}_p$. Suppose also that the condition for a curve to be "weak" is likely to be independent of its isogeny or endomorphism class. In such a case we might want to choose our curve E over $\mathbb{F}_p$ to be in a very small endomorphism class — more precisely, in a small L-conductor-gap class for fairly large L — so that an attacker could not use isogenies to transport the discrete log problem from our curve to a weak curve.

Example 5. Choose B to be a random k-bit prime, and choose A to be a random even number (perhaps also of k bits, but A may be chosen to have fewer bits) such that (i) $p = A^2 + B^2$ is prime, and (ii) either $n = (p+1)/2 - A$ or $n = (p+1)/2 + A$ is a prime. Heuristically one expects to have to test $O(k^2)$ values of A in order to obtain conditions (i) and (ii). Then the curve E over $\mathbb{F}_p$ with equation

$$y^2 = x^3 - \alpha x$$

has $2n$ points, where $\alpha \in \mathbb{F}_p$ is a quadratic non-residue whose quartic residue class depends on the sign in $n = (p+1)/2 \mp A$ (see Sections 9.8 and 18.4 of [45]). The trace of E is $\pm 2A$, and its discriminant is $4A^2 - 4p = -4B^2$. Because B is prime, for $k \geqslant 80$ it is completely infeasible to transport the discrete log problem on E to that on a generic isogenous curve. Note that E has complex multiplication by the full ring of integers $\mathbb{Z}[i]$ (since i acts on the curve by $(x, y) \mapsto (-x, iy)$, where the latter occurrence of i denotes a square root of -1 in the finite field); that is, $\mathrm{End}(E)$ has conductor 1. Up to isomorphism E is the only curve in its conductor-gap class, and the endomorphism ring of any of the other isogenous curves has conductor B.

The method of parameter selection in this example of course directly flouts the advice of Brainpool [73]. But whether it is reckless or wise to do this is at present far from clear.

Our purpose in giving these examples is not to lobby for the use of special curves in preference to random ones. Rather, our point is that conventional wisdom may turn out to be wrong and that, as far as anyone knows, either choice has risks. The decision about what kind of curve to use in ECC is a subjective one based on the user's best guess about future vulnerabilities.

As frequently happens in cryptography, a close examination of a commonly accepted viewpoint on security issues reveals that opposing opinions or interpretations cannot be ruled out. Much as we might wish to convey to the outside world an impression of self-confidence and mathematical certainty about our recommendations (see Section 2), there is ample reason to wonder whether this self-confidence is justified.

12. Path dependence

In [74] MacKenzie and Wajcman discuss what they call the *path-dependence* of technical change:

> Technologies often manifest increasing returns to adoption. The processes of learning by doing and by using... and the frequent focus of inventive effort on removing weak points... from existing

> technologies, mean that the very process of adoption tends to improve the performance of those technologies that are adopted. This gives the history, especially the early history, of a technology considerable significance. Early adoptions, achieved for whatever reason, can be built into what may become irreversible superiority over rivals, because success tends to breed success and rejection can turn into neglect and therefore permanent inferiority. The history of technology is a path-dependent history, one in which past events exercise continuing influences. Which of two or more technologies eventually succeeds is not determined by their intrinsic characteristics alone, but also by their histories of adoption. The technology that triumphs is not necessarily abstractly best.... Path-dependence means that local, short-term contingencies can exercise lasting effects.

12.1. Historical what–ifs

One of the best ways to refute the technological deterministic view of the history of cryptography that is implicit in the Ideal Model (see the last paragraph of Section 2) is to indulge in what is sometimes called "counterfactual history" (see [19,27]) and ask some hypothetical questions of the form "What if...?"

• What if in 1977 someone who had just written a PhD thesis on elliptic curves had happened to read the classic article [24] that had appeared the year before? Fortunately for RSA, that appears not to have happened, since it probably would have occurred to such a person to suggest replacing the multiplicative group of a finite field by the group of points of an elliptic curve, and ECC would have been born eight years before 1985. In 1977 subexponential algorithms were already known for the integer factorization problem. The elliptic curve discrete log problem thus would have struck everyone as a much harder problem, and hence the one-way function in ECC would have appeared to be much safer for the construction of public key protocols. There would have been little reason for anyone to adopt RSA.

• What if the ideas described in Section 10 for finding discrete logs on an elliptic curve E over $\mathbb{F}_{q^m}$ — Weil descent followed by index calculus on a jacobian group, or index calculus directly on E using points with x-coordinate in $\mathbb{F}_q$ as the factor base — had been proposed in the late 1980s or early 1990s? At that time it was generally assumed that any finite field could be used in ECC, and the choice should depend only on convenience. In fact, some people proposed using fields of the form $\mathbb{F}_{q^m}$ with q a prime or a power of 2 of intermediate size (say, 8 or 32 bits). The faster-than-squareroot and even subexponential algorithms for some elliptic curves over such "weak" fields would have come as a shock, and opponents of ECC could have easily used the discovery of such algorithms as a reason not to have confidence in elliptic curves. As it happened, however, by the late 1990s implementers were almost exclusively using either prime fields or prime degree extensions of $\mathbb{F}_2$, to which those algorithms do not apply.

• What if pairing-based cryptography had been proposed just three or four years earlier, say in 1997 when "ECC Central" on the RSA website was warning of the dangers of ECC? The elliptic curve skeptics would have had a field day! "The ECC promoters are now using the very same low-embedding-degree elliptic curves that five years ago they acknowledged to be insecure and recommended avoiding!" they would have said. Some people would have hoped that by undermining confidence in pairing-based cryptography they might be able to bring down all of ECC with it. However, by 2001 the big rivalry between RSA and ECC was largely over, and hardly anyone wanted to reopen that debate.

These hypothetical questions show that the particular chronology of RSA, ECC, pairing-based cryptography, and new algorithms for factoring and discrete logs has a lot to do with the history of the paradigm shift to ECC.

12.2. Narrative inversion

In historical studies one often finds a wide gap between the image that a nation or group has of its past — the historical *narrative* — and what the record shows. In extreme cases it sometimes seems that the farther this narrative is from reality, the more adamantly people repeat it and insist on its validity. This is *narrative inversion*.

Take the hypothetical example of a country whose official version of its history is that the guiding principle of its foreign policy has been to “defend freedom.” Even though people in other countries might see an ever-widening chasm between this national myth and the reality, the narrative continues to be a centerpiece of the belief system of millions of people, who proclaim it with increasing fervor.

In the world of scholarship as well, one often encounters narrative inversion. Take, for example, the word *science*. Often the humanistic and social areas whose practitioners are the most insistent on using this word are those fields that have the worst track record in attempting to use scientific methodology. In the last century the term “social studies” was replaced by “social sciences,” and departments of government became departments of “political science.” Interestingly, the one profession in social studies that arguably uses a fair amount of scientific methodology and does it competently — history — has never insisted on changing its name to “historical science.”

12.3. Narrative inversion in cryptography

Modern cryptography can be viewed as an applied science in the overlap between mathematics and computer science. Nevertheless, the development of various technologies for implementing secure communications continues to be as much a story of chance occurrences, mistaken interpretations, zigzags, blunders, and strokes of good luck as was the cryptography of old. It seems impossible to remove the element of contingency — of intuition and of craft.

Part of the reason why cryptography has such a strong subjective element is that speculation is central to the field. When deciding on the basic type of cryptography to use (RSA or ECC, for example), when choosing the type of protocol for a given application (e.g., whether or not to use identity-based encryption), and when selecting parameters (for instance, random generation versus enhanced efficiency), one has to make a guess about future developments in order to evaluate the fundamental issue of safety of the system. One has to ask: What type of adversaries are we likely to encounter, and what will be their most likely avenues of attack? Will there be any breakthroughs in bringing down the asymptotic running time to solve any of the supposedly intractable mathematical problems? Will quantum computing (see [95]) ever become practical? What new “side-channel” attacks (see [2,9,63,64]) might be devised?

Perhaps it is because of this highly contingent element in the field that researchers increasingly feel the need to go out of their way to assure the public that it is rapidly becoming a science, that ironclad guarantees of security can be given (“provable security”), and that cryptographers faithfully follow the Ideal Model described in Section 2.

Among the leading researchers in cryptography, Mihir Bellare (coinventor of the subdiscipline of *practice-oriented provable security*) has a relatively moderate view of the scientific nature of the field. On the one hand, he acknowledges that the search for suitable mathematical one-way functions — what he calls *atomic primitives* — has a large element of artistry [6]. But on the other hand, he thinks that the part of cryptography concerned with constructing usable cryptosystems based on these primitives is becoming a science:

> ...I would like to claim that the design of protocols can be made a science [6].

Other theoreticians, writing more recently, are categorical in their rejection of any notion that cryptography is not fully a science. In response to comments in [60] questioning the claims of “provable security” and suggesting that cryptography is “more an art than a science,” Oded Goldreich [35] stated that

> ...*cryptographic research is indeed part of science.* [emphasis in original]

And in the preface to their recent book [49] Jonathan Katz and Yehuda Lindell insisted that

> ...*cryptographic constructions can be proven secure* with respect to a clearly-stated definition of security and relative to a well-defined cryptographic assumption. This is the essence of modern

> cryptography, and what has transformed cryptography from an art to a science. The importance of this idea cannot be over-emphasized. [emphasis in original]

As in other cases of narrative inversion, these belabored claims inevitably bring to mind the famous line from *Hamlet*

> The lady doth protest too much, methinks.

13. Social construction of science and technology

Until the work of Thomas Kuhn a half century ago, the term Scientific Revolution, used in the singular, referred to the birth of modern science in the 16th and 17th centuries. Kuhn, whose most famous book [68] appeared in 1962, used the term “scientific revolutions” in the plural to refer to the radical shifts of point of view that have punctuated the history of science. It was Kuhn who coined the term “paradigm shift” for this process — a term that was later used in many areas outside the sciences.

Before Kuhn’s work, the most common view among historians was that science and technology progressed steadily toward a more accurate and complete understanding of the natural world. Mistakes were often made — and one could frequently find social and political explanations for the backward steps — but the overall pattern was to build upon the edifice constructed by earlier generations, “standing on the shoulders of giants” in Newton’s famous formulation.

Kuhn, however, believed that the most important developments in the history of science occur by means of a radical challenge to earlier conceptions. In his view social and professional influences have a great effect on the direction of science as a whole, and should not be invoked simply to explain the “mistakes.” In fact, he believed in what later came to be called the “symmetry principle,” which states that historians should use the same methods to study the emergence of a scientific theory or school of thought independently of whether modern science regards the theory as correct or incorrect.

In the years after Kuhn’s book appeared, this methodological principle was carried much further by other historians and philosophers, who started advocating a type of scientific relativism, according to which science has no more objective validity than anything else. Science, according to this view, is “socially constructed” just as literature, politics, and religion are. The relativist tendency in thinking about science was most pronounced among postmodernists such as Jacques Lacan, Paul Feyerabend, Vandana Shiva, and Bruno Latour.

In the 1980s and 1990s as this tendency grew in strength among academics in the humanities and social studies, some scientists started to take notice. Most reacted with horror and anger at what the postmodern writers were saying. Some, such as Holton [41] and Gross and Levitt [37], published refutations. Journalists often referred to the debate as the “science wars.”

The culmination of the scientists’ counterattack on postmodern writings on science came in the form of a hoax — perhaps the most successful hoax in the history of academic writing. After two months of studying the relevant literature, physicist Alan Sokal wrote a parody of the postmodern science studies jargon in the form of an article on the “hermeneutics” of quantum gravity. He submitted it to the journal *Social Text* for publication. Astoundingly, the caricature was accepted, and his article [101] appeared in the Spring/Summer 1996 issue of the journal.

Sokal’s hoax and subsequent critiques [14,102] were directed against an extreme form of “science studies.” But even the more moderate sociologists of science sometimes write in a style that reveals an unmistakable undercurrent of resentment and pique toward the sciences and technology. For example, in his introduction to the book “Technology and Social Process,” B. Elliott [26] says:

> Running through many of the chapters in this book... there is a concern to demystify technology. The social studies of science have shown us that the closer we get to the laboratories, to the day-to-day practice of science, and the more intimately we explore the social processes through which scientific knowledge is constituted, the less in awe of it we stand. We appreciate the looseness of its boundaries, the contested nature of its claims. Scientific research turns out to be much messier than we had perhaps supposed, its development less the product of logical and rational

> progression than the product of hunches, improbable connections and various struggles for power. In consequence we grow more skeptical of claims that may be made for its special, privileged status and more critical of various forms of scientific determinism. And so it is, and should be, with technology.

It is tempting for scientists to revel in Sokal's spoof of the nonsensical jargon of the postmodernists, react with annoyance to the tone of sociologists of science such as Elliott, and dismiss as ridiculous any attempt to describe a school of scientific thought as a "social construct." But that would be a mistake.

There have been many studies of the effects of such social factors as race, class, and gender on the content of scientific theories. In some areas of the sciences — especially those connected with human behavior — and in many areas of technology these studies have exposed serious methodological failings.

This work in the history and sociology of science and technology on occasion can be fascinating — and it can be especially entertaining when gender bias is involved. In the sciences the influence of gender has been most pronounced in such fields as primatology, endocrinology, embryology, archaeology, and sociobiology. We will give brief (and admittedly superficial) descriptions of three examples.

13.1. Gorillas

The field of primatology emerged toward the end of the 19th century in the wake of Darwin's pathbreaking work on natural selection and evolution, which was popularly known as the theory that "man descended from apes." For close to a century most primatologists visualized ape family life as conforming to Victorian views of gender roles. A now-classic example of this tendency could be seen in the primate hall of the American Museum of Natural History, which featured a majestic male silverback gorilla towering over and guarding his much smaller mate and their offspring. This memorable tableau directly replicated the stereotypical Victorian nuclear family in the primate world. Unstated anywhere in the exhibit was the fact that several large female gorillas had been shot in the mistaken belief that they were males worthy of being stuffed for the Museum. Unstated also was the fact that such a scene of a nuclear ape family would have been highly unlikely in nature [38,39]. It was not until the 1960s and 1970s, with the rise of second-wave feminism and the entrance of a new generation of women into the profession, that primatologists began to systematically question the patriarchal assumptions of their older colleagues and point out that primates exhibit a surprising variety of social organizations, parenting strategies, dominance hierarchies, and male/female relations [42,104].

13.2. The Hohokam

A more recent and equally amusing example of gender bias can be found in a currently fashionable trend in the study of the prehistoric Southwest of the U.S. A group of archaeologists led by Steven LeBlanc of Harvard and David Wilcox of the Museum of Northern Arizona have developed an elaborate theory of endemic warfare among the ancient Hohokam peoples of central Arizona (see [52,53]). Wilcox and his coauthors describe how, while talking late at night around a campfire, they arrived at their "exciting" conclusions, which seem to be the result of lively imagination stimulated by male-to-male comradery rather than any scholarly deductions. They recount with awe and describe as "scientific" the "seminal ideas" supplied by a much-decorated veteran of the Vietnam War who visited the sites with Wilcox. Although there is scant evidence for any warfare among the Hohokam, let alone battles of epic proportions, this group of archaeologists has insisted on a version of prehistory that appears to have more to do with a modern American culture of aggressive masculinity than with the actual interactions among peoples in 14th-century Arizona.

13.3. Smart houses

Examples of gender bias abound in technology. For example, the much-hyped "smart house" was analyzed in a delightful article by Anne-Jorunn Berg [7]. On the face of it, one would expect that efforts to automate household processes would naturally incorporate input from women, who are the primary users of technology within the home. Berg found, however, that “smart house” designers had been remarkably uninterested in grappling with such labor-intensive activities as cooking, cleaning and childcare. Instead, designers had concentrated on getting peripheral technologies to “talk” to one another through central control stations: getting lights to turn on automatically, arranging voice-activated controls for entertainment systems, etc. The closest any of the projects had come to a basic household task was the “robobutler,” which could supposedly serve drinks. But the drinks must first be made by a human and set precisely on the robobutler’s tray, at which point the robobutler could be remotely guided into the livingroom by controls much like those used for toy boats and planes. In other words, this technology was a “toy for the boys” rather than a true labor-saving advance. The core tasks that make housework so time-consuming remained untouched.

13.4. The social study of technology

The social study of technology has become a subfield in its own right relatively recently — a decade or two after the emergence of the social study of science. Sociologists of technology have usually avoided the pitfalls of jargon and over-generalization that have plagued writers on the social construction of science. In part this is a conscious strategy adopted in order to avoid a repetition of the “science wars.” In the preface to the 1999 edition of their book [74] MacKenzie and Wajcman warn their colleagues not to repeat the mistakes of sociologists of science:

> We fear a rerun in the social studies of technology of what has happened in the social studies of science. There, in the 1970s and 1980s, a variety of empirical studies… offered evidence that the content of scientific knowledge was influenced by the social circumstances of its production.… Those who produced this work knew well that the evidence was partial, tentative and patchy, and that the conceptual issues involved were poorly understood, but a wider audience of scholars in the humanities and social sciences grasped eagerly at the conclusion that scientific knowledge was ‘a social construct.’ The notion became something of a premature orthodoxy, and too little was done to clarify what the ambiguous phrase meant.… In consequence, when some natural scientists reacted with hostility to the notion of social construction (in the ‘science wars’ debate…), the field was not as well placed as it might have been.

In addition, it is easier for sociologists of technology to get a sympathetic reception from practitioners, because most leaders of the field are fully conscious of the role that economic, social and political factors play in the adoption of certain technologies and the rejection of others. A famous example was Thomas Edison (see [43,44]), who was as much an entrepreneur as an inventor. Few technologists would see their work as being entirely removed from the cultural environment. In contrast, researchers in the basic sciences — especially the physical sciences and mathematics — tend to see their work as a process of discovering a set of truths that transcend the exigencies of the moment.

However theoreticians in math and physics might view their work, practitioners in applied fields such as cryptography would have to be very naive in order to believe that their ideas and protocols have some sort of intrinsic value apart from human culture. It is quite a stretch to visualize RSA or ECC inhabiting a realm of Platonic Ideals side by side with the perfect circle, the Pythagorean theorem, and the Heisenberg uncertainty principle.

13.5. Conclusion

To what extent can ideas from “social construction of technology” help us to understand the paradigm shift from RSA to ECC? Certain broad social categories do not appear to be relevant to this history; to the best of our knowledge questions of gender, race, or class have nothing to do with this story. It is perhaps particularly surprising that gender has played no discernible role, since the popular history of cryptography is intimately tied up with the military and is full of male-dominated stories of intrigue. Nevertheless, we are aware of no examples of gender, race, or class bias influencing the direction of public key cryptography.

Nor are we aware of evidence that women are especially put off by the disciplinary culture of the field — any more than in other areas of computer science, engineering, and mathematics. In fact, the proportion of prominent women researchers is probably greater than that in most of the allied fields of science. This flies in the face of some currently popular notions about women in science, according to which women by nature avoid fields where conflict and confrontation are common (see, for example, [5,34]). In the case of cryptography, adversarial behavior is central to the very definition of the subject, and the "spy vs. spy" mentality of intense rivalry often seems to permeate the research community as well. Heated disputes might involve the relative merits of different types of cryptography or of different approaches to evaluating them, or such mundane matters as which of several contending research groups deserves credit for advancing a particular subfield.

Just as we reject technological determinism, we should also avoid a type of equally simplistic sociological determinism that is sometimes called *essentialism*. In thinking about disciplines such as cryptography, some feminist theoreticians would reason more or less as follows: "Women by their nature are less confrontational and militaristic than men, and hence are less likely to be attracted to a field whose entire purpose is to combat adversarial behavior." And postmodern feminist philosophers of science would also see a gender subtext in the debate over whether cryptography is more a science or an art. They would most likely claim that the very term "science" is so tied to masculinity that some cryptographers' fixation on the word "science" to describe their field (see Section 12.3) is *prima facie* evidence of a male bias. (See [51] for a discussion of some of the fallacies in postmodern feminist views of science.)

Similarly, in trying to explain the dramatic contrast between the recommendations of Brainpool and of Voltage (see Section 11) some might be tempted to resort to popular stereotypes of national character: "Germans are risk-averse by nature, whereas Americans have a penchant for high-stakes gambling, so that is why German-led Brainpool was extra-cautious, whereas the American security company Voltage happily endorsed a supersingular curve." In our opinion such attempts at explanation based on gender or national character are far-fetched and untenable.

The social influences on the course of public key cryptography appear to have come not from such broad categories as gender, race, class, or nationality, but rather from certain aspects of the professional culture. This is not unusual in the history of technology. As MacKenzie and Wajcman put it,

> …'social shaping' does not necessarily involve reference to wider societal relations such as those of class, gender and ethnicity. These *are* sometimes directly crucial… but often what is more immediately relevant are 'local' considerations, such as engineers' membership of professional communities, the reward structures of those communities, and so on [74, pp. 18–19].

Our examination of the history of ECC offers no support to those who would argue that technology follows an inevitable path that is independent of societal constraints. Rather, the evidence points toward ways in which technology is socially constructed:

- *path-dependence* — the importance of timing, the role of happenstance;
- *the role of the military* in intervening at crucial stages to send the technology in a different direction from that favored by market forces;
- *closure* — the need eventually to reach a consensus and stop most debate, even if some basic questions remain unanswered;
- *narrative inversion* — a desire to use high-status terms such as "science" and "mathematical proof" that becomes more fervent even as the field is showing itself again and again to be as much an art as a science.

Acknowledgments

We would like to thank Ian Blake, Claus Diem, Andreas Enge, Steven Galbraith, David Goss, David Jao, Susan Landau, and Arjen Lenstra for their comments on earlier drafts of this paper.

Supplementary material

The online version of this article contains additional supplementary material.
Please visit doi:10.1016/j.jnt.2009.01.006.

References

[1] L. Adleman, J. DeMarrais, M. Huang, A subexponential algorithm for discrete logarithms over the rational subgroup of the jacobians of large genus hyperelliptic curves over finite fields, in: Algorithmic Number Theory: First International Symposium, in: Lecture Notes in Comput. Sci., vol. 877, Springer-Verlag, 1994, pp. 28–40.

[2] D. Agrawal, B. Archambeault, J. Rao, P. Rohatgi, The EM side-channel(s), in: Cryptographic Hardware and Embedded Systems — CHES 2002, in: Lecture Notes in Comput. Sci., vol. 2523, 2002, pp. 29–45.

[3] American National Standards Institute (ANSI) X9.62, Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA), 1999.

[4] R. Balasubramanian, 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.

[5] M. Belenky, et al., Women's Ways of Knowing: The Development of Self, Voice, and Mind, Basic Books, 1986.

[6] M. Bellare, Practice-oriented provable-security, in: Proc. First International Workshop on Information Security (ISW '97), in: Lecture Notes in Comput. Sci., vol. 1396, 1998, pp. 221–231.

[7] A.-J. Berg, A gendered socio-technical construction: The smart house, in: The Social Shaping of Technology, second ed., Open Univ. Press, 1999, pp. 301–313.

[8] D. Boneh, X. Boyen, Short signatures without random oracles and the SDH assumption on bilinear groups, J. Cryptology 21 (2008) 149–177.

[9] D. Boneh, R. DeMillo, R. Lipton, On the importance of checking cryptographic protocols for faults, in: Advances in Cryptology — EUROCRYPT '97, in: Lecture Notes in Comput. Sci., vol. 1233, 1997, pp. 37–51.

[10] D. Boneh, M. Franklin, Identity-based encryption from the Weil pairing, SIAM J. Comput. 32 (2003) 586–615.

[11] D. Boneh, B. Lynn, H. Shacham, Short signatures from the Weil pairing, in: Progress in Cryptology — ASIACRYPT 2001, in: Lecture Notes in Comput. Sci., vol. 2248, 2001, pp. 514–532.

[12] X. Boyen, L. Martin, Identity-Based Cryptography Standard (IBCS) #1: Supersingular curve implementations of the BF and BB1 cryptosystems, IETF RFC, available from http://www.ietf.org/rfc/rfc5091.txt, 2007.

[13] E. Braun, S. Macdonald, Revolution in Miniature: The History and Impact of Semiconductor Electronics, Cambridge Univ. Press, 1978.

[14] J. Bricmont, A. Sokal, Fashionable Nonsense: Postmodern Intellectuals' Abuse of Science, St. Martin's Press, 1998.

[15] R. Bröker, D. Charles, K. Lauter, Evaluating large degree isogenies and applications to pairing based cryptography, in: Pairing-Based Cryptography — Pairing 2008, in: Lecture Notes in Comput. Sci., vol. 5209, 2008, pp. 100–112.

[16] D. Brown, R. Gallant, The static Diffie–Hellman problem, http://eprint.iacr.org/2004/306/.

[17] J. Cheon, Security analysis of the strong Diffie–Hellman problem, in: Advances in Cryptology — EUROCRYPT 2006, in: Lecture Notes in Comput. Sci., vol. 4004, 2006, pp. 1–11.

[18] D. Coppersmith, Fast evaluation of logarithms in fields of characteristic two, IEEE Trans. Inform. Theory 30 (1984) 587–594.

[19] R. Cowley (Ed.), What If? The World's Foremost Military Historians Imagine What Might Have Been, Berkley Trade, 2000.

[20] C. Diem, An index calculus algorithm for plane curves of small degree, in: Algorithmic Number Theory: Seventh International Conference, in: Lecture Notes in Comput. Sci., vol. 4076, 2006, pp. 543–557.

[21] C. Diem, On arithmetic and the discrete logarithm problem in class groups of curves, Habilitationsschrift, Universität Leipzig, available from http://www.math.uni-leipzig.de/~diem/preprints/habil.pdf.

[22] C. Diem, P. Gaudry, N. Thériault, E. Thomé, A double large prime variation for small genus hyperelliptic index calculus, Math. Comp. 76 (2007) 475–492.

[23] C. Diem, E. Thomé, Index calculus in class groups of non-hyperelliptic curves of genus three, J. Cryptology 21 (2008) 593–611.

[24] W. Diffie, M. Hellman, New directions in cryptography, IEEE Trans. Inform. Theory 22 (1976) 644–654.

[25] W. Diffie, S. Landau, Privacy on the Line: The Politics of Wiretapping and Encryption, second ed., MIT Press, 2007.

[26] B. Elliott, Introduction, Technology and Social Process, Edinburgh Univ. Press, 1988, pp. 1–7.

[27] N. Ferguson (Ed.), Virtual History: Alternatives and Counterfactuals, Picador, 1997.

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

[29] S. Galbraith, Constructing isogenies between elliptic curves over finite fields, LMS J. Comput. Math. 2 (1999) 118–138.

[30] S. Galbraith, V. Rotger, Easy decision Diffie–Hellman groups, LMS J. Comput. Math. 7 (2004) 201–218.

[31] R. Gallant, R. Lambert, S. Vanstone, Improving the parallelized Pollard lambda search on an anomalous binary curve, Math. Comp. 69 (2000) 1699–1705.

[32] P. Gaudry, Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem, J. Symbolic Comput., in press, available from http://www.loria.fr/~gaudry/papers.en.html.

[33] P. Gaudry, F. Hess, N. Smart, Constructive and destructive facets of Weil descent on elliptic curves, J. Cryptology 15 (2002) 19–34.

[34] C. Gilligan, In a Different Voice: Psychological Theory and Women's Development, Harvard Univ. Press, 1982.

[35] O. Goldreich, On post-modern cryptography, available from http://eprint.iacr.org/2006/461/.

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

[37] P. Gross, N. Levitt, Higher Superstition: The Academic Left and Its Quarrels with Science, Johns Hopkins Univ. Press, 1994.

[38] D. Haraway, Primate Visions: Gender, Race, and Nature in the World of Modern Science, Routledge, 1989.

[39] D. Haraway, Race: Universal Donors in a Vampire Culture, The Haraway Reader, Routledge, 2003, pp. 251–293.

[40] F. Hess, Generalizing the GHS attack on the elliptic curve discrete logarithm problem, LMS J. Comput. Math. 7 (2004) 167–192.

[41] G. Holton, Science and Anti-Science, Harvard Univ. Press, 1993.

[42] S. Hrdy, The Woman That Never Evolved, Harvard Univ. Press, 1981.

[43] T. Hughes, The seamless web: Technology, science, et cetera, et cetera, in: Technology and Social Processes, Edinburgh Univ. Press, 1988, pp. 9–19.

[44] T. Hughes, Edison and electric light, in: The Social Shaping of Technology, second ed., Open Univ. Press, 1999, pp. 50–63.

[45] K. Ireland, M. Rosen, A Classical Introduction to Modern Number Theory, second ed., Springer-Verlag, 1990.

[46] M. Jacobson, N. Koblitz, J. Silverman, A. Stein, E. Teske, Analysis of the xedni calculus attack, Des. Codes Cryptogr. 20 (2000) 41–64.

[47] D. Jao, S. Miller, R. Venkatesan, Do all elliptic curves of the same order have the same difficulty of discrete log?, in: Advances in Cryptology — ASIACRYPT 2005, in: Lecture Notes in Comput. Sci., vol. 3788, 2005, pp. 21–40.

[48] A. Joux, A one round protocol for tripartite Diffie–Hellman, in: Algorithmic Number Theory: Fourth International Symposium, in: Lecture Notes in Comput. Sci., vol. 1838, 2000, pp. 385–393.

[49] J. Katz, Y. Lindell, Introduction to Modern Cryptography, CRC Press, 2007.

[50] R. Kline, T. Pinch, The social construction of technology, in: The Social Shaping of Technology, second ed., Open Univ. Press, 1999, pp. 113–115.

[51] A.H. Koblitz, A historian looks at gender and science, Internat. J. Sci. Ed. 9 (1987) 399–407.

[52] A.H. Koblitz, Male bonding around the campfire: Constructing myths of Hohokam militarism, Men Masculinities 9 (2006) 95–107.

[53] A.H. Koblitz, Warriors, campfires, and a big stick: Modern male fantasies of Hohokam militarism, Bull. Old Pueblo Archaeology Center 53 (2008) 2–5.

[54] N. Koblitz, Elliptic curve cryptosystems, Math. Comp. 48 (1987) 203–209.

[55] N. Koblitz, Hyperelliptic cryptosystems, J. Cryptology 1 (1989) 139–150.

[56] N. Koblitz, CM-curves with good cryptographic properties, in: Advances in Cryptology — CRYPTO '91, in: Lecture Notes in Comput. Sci., vol. 576, 1992, pp. 279–287.

[57] N. Koblitz, An elliptic curve implementation of the finite field Digital Signature Algorithm, in: Advances in Cryptology — CRYPTO '98, in: Lecture Notes in Comput. Sci., vol. 1462, 1998, pp. 327–337.

[58] N. Koblitz, A. Menezes, Pairing-based cryptography at high security levels, in: Proceedings of the Tenth IMA International Conference on Cryptography and Coding, in: Lecture Notes in Comput. Sci., vol. 3796, 2005, pp. 13–36.

[59] N. Koblitz, A. Menezes, Another look at 'provable security'. II, in: Progress in Cryptology — INDOCRYPT 2006, in: Lecture Notes in Comput. Sci., vol. 4329, 2006, pp. 148–175.

[60] N. Koblitz, A. Menezes, Another look at 'provable security', J. Cryptology (2007) 3–37.

[61] N. Koblitz, A. Menezes, Another look at generic groups, Adv. Math. Commun. 1 (2007) 13–28.

[62] N. Koblitz, A. Menezes, Another look at non-standard discrete log and Diffie–Hellman problems, J. Math. Cryptol. 2 (2008) 311–326.

[63] P. Kocher, Timing attacks on implementations of Diffie–Hellman, RSA, DSS, and other systems, in: Advances in Cryptology — CRYPTO '96, in: Lecture Notes in Comput. Sci., vol. 1109, 1996, pp. 104–113.

[64] P. Kocher, J. Jaffe, B. Jun, Differential power analysis, in: Advances in Cryptology — CRYPTO '99, in: Lecture Notes in Comput. Sci., vol. 1666, 1999, pp. 388–397.

[65] D. Kohel, Endomorphism rings of elliptic curves over finite fields, PhD thesis, Univ. of California at Berkeley, 1996.

[66] M. Kraitchik, Théorie des nombres, vol. 1, Gauthier–Villars, 1922.

[67] M. Kraitchik, Recherches sur la théorie des nombres, Gauthier–Villars, 1924.

[68] T. Kuhn, The Structure of Scientific Revolutions, Univ. of Chicago Press, 1962.

[69] S. Landau, Communications security for the twenty-first century: The Advanced Encryption Standard, Notices Amer. Math. Soc. 47 (2000) 450–459.

[70] H.W. Lenstra Jr., Factoring integers with elliptic curves, Ann. of Math. 126 (1987) 649–673.

[71] A.K. Lenstra, E.R. Verheul, The XTR public key system, in: Advances in Cryptology — CRYPTO 2000, in: Lecture Notes in Comput. Sci., vol. 1880, 2000, pp. 1–19.

[72] S. Levy, Crypto: How the Code Rebels Beat the Government — Saving Privacy in the Digital Age, Viking Penguin, 2001.

[73] M. Lochter, J. Merkle, ECC Brainpool standard curves and curve generation, IETF Internet-Draft, February 18, 2008.

[74] D. MacKenzie, J. Wajcman, Preface to the second edition, and Introductory essay: The social shaping of technology, in: The Social Shaping of Technology, second ed., Open Univ. Press, 1999, pp. xiv–xvi and 3–27.

[75] L. Martin, A closer look at pairings, presentation at the NIST workshop "Applications of Pairing Based Cryptography: Identity Based Encryption and Beyond", June 3–4, 2008, available from http://csrc.nist.gov/groups/ST/IBE/documents/June08.

[76] A. Menezes, T. Okamoto, S. Vanstone, Reducing elliptic curve logarithms to logarithms in a finite field, IEEE Trans. Inform. Theory 39 (1993) 1639–1646.

[77] A. Menezes, M. Qu, Analysis of the Weil descent attack of Gaudry, Hess and Smart, in: Topics in Cryptology — CT-RSA 2001, in: Lecture Notes in Comput. Sci., vol. 2020, 2001, pp. 308–318.

[78] A. Menezes, E. Teske, Cryptographic implications of Hess' generalized GHS attack, Appl. Algebra Engrg. Comm. Comput. 16 (2006) 439–460.

[79] A. Menezes, E. Teske, A. Weng, Weak fields for ECC, in: Topics in Cryptology — CT-RSA 2004, in: Lecture Notes in Comput. Sci., vol. 2964, 2004, pp. 366–386.

[80] V. Miller, Uses of elliptic curves in cryptography, in: Advances in Cryptology — CRYPTO '85, in: Lecture Notes in Comput. Sci., vol. 218, 1986, pp. 417–426.

[81] V. Miller, Short programs for functions on curves, unpublished manuscript, 1986.

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

[83] A. Miyaji, M. Nakabayashi, S. Takano, New explicit conditions of elliptic curve traces for FR-reduction, IEICE Trans. Fundamentals E84-A (2001) 1234–1243.

[84] D. Moody, The Diffie–Hellman problem and generalization of Verheul's theorem, PhD thesis, Univ. of Washington, 2009.

[85] V. Müller, Fast multiplication on elliptic curves over small fields of characteristic two, J. Cryptology 11 (1998) 219–234.

[86] National Institute of Standards and Technology, Digital Signature Standard, Federal Information Processing Standards Publication 186-2, 2000.

[87] National Security Agency, The case for elliptic curve cryptography, http://www.nsa.gov/ia/industry/crypto_elliptic_curve.cfm.

[88] S. Pohlig, M. Hellman, An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance, IEEE Trans. Inform. Theory 24 (1978) 106–110.

[89] J. Pollard, Monte Carlo methods for index computation mod p, Math. Comp. 32 (1978) 918–924.

[90] R. Sakai, K. Ohgishi, M. Kasahara, Cryptosystems based on pairings, in: Proceedings of the 2000 Symposium on Cryptography and Information Security, Okinawa, 2000.

[91] O. Schirokauer, The number field sieve for primes of low hamming weight, available from http://eprint.iacr.org/2006/107.

[92] R. Schoof, Elliptic curves over finite fields and the computation of square roots mod p, Math. Comp. 44 (1985) 483–494.

[93] I. Semaev, Summation polynomial and the discrete logarithm on elliptic curves, http://eprint.iacr.org/2004/031/.

[94] A. Shamir, Identity-based cryptosystems and signature schemes, in: Advances in Cryptology — CRYPTO '84, in: Lecture Notes in Comput. Sci., vol. 196, 1985, pp. 277–296.

[95] P.W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Comput. 26 (1997) 1484–1509.

[96] J. Silverman, The Arithmetic of Elliptic Curves, Springer-Verlag, 1986.

[97] J. Silverman, The xedni calculus and the elliptic curve discrete logarithm problem, Des. Codes Cryptogr. 20 (2000) 5–40.

[98] J. Silverman, The four faces of lifting for the elliptic curve discrete logarithm problem, in: 11th Workshop on Elliptic Curve Cryptography, Univ. College Dublin, September 5, 2007, available from http://mathsci.ucd.ie/~gmg/ECC2007Talks.

[99] J. Silverman, J. Suzuki, Elliptic curve discrete logarithms and the index calculus, in: Advances in Cryptology — ASIACRYPT '98, in: Lecture Notes in Comput. Sci., vol. 1514, 1998, pp. 110–125.

[100] B. Smith, Isogenies and the discrete logarithm problem in jacobians of genus 3 hyperelliptic curves, in: Advances in Cryptology — EUROCRYPT 2008, in: Lecture Notes in Comput. Sci., vol. 4965, 2008, pp. 163–180.

[101] A. Sokal, Transgressing the boundaries: Toward a transformative hermeneutics of quantum gravity, Social Text 46/47 (1996) 217–252.

[102] A. Sokal, Beyond the Hoax: Science, Philosophy and Culture, Oxford Univ. Press, 2008.

[103] J. Solinas, Efficient arithmetic on Koblitz curves, Des. Codes Cryptogr. 19 (2000) 195–249.

[104] S. Strum, L. Fedigan (Eds.), Primate Encounters: Models of Science, Gender, and Society, Univ. of Chicago Press, 2000.

[105] E. Verheul, Evidence that XTR is more secure than supersingular elliptic curve cryptosystems, in: Advances in Cryptology — EUROCRYPT 2001, in: Lecture Notes in Comput. Sci., vol. 2045, 2001, pp. 195–210.

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

[107] L. Washington, Elliptic Curves: Number Theory and Cryptography, second ed., CRC Press, 2008.

[108] M. Wiener, R. Zuccherato, Faster attacks on elliptic curve cryptosystems, in: Selected Areas in Cryptography — SAC '98, in: Lecture Notes in Comput. Sci., vol. 1556, 1999, pp. 190–200.

[109] D. Wolf, Assurance provider: Designing a roadmap for information security, http://www.military-information-technology.com/article.cfm?DocID=1294.