ECC: Do We Need to Count?

Jean-Sébastien Coron[1,2], Helena Handschuh[2,3], and David Naccache[2]

[1] École Normale Supérieure
45 rue d'Ulm, F-75005, Paris, France
coron@clipper.ens.fr
[2] Gemplus Card International
34 rue Guynemer, Issy-les-Moulineaux, F-92447, France
{coron,handschuh,naccache}@gemplus.com
[3] École Nationale Supérieure des Télécommunications
46 rue Barrault, F-75013, Paris, France
handschu@enst.fr

Abstract. A prohibitive barrier faced by elliptic curve users is the difficulty of computing the curves' cardinalities. Despite recent theoretical breakthroughs, point counting still remains very cumbersome and intensively time consuming.
In this paper we show that point counting can be *avoided* at the cost of a protocol slow-down. This slow-down factor is quite important (typically $\cong 500$) but proves that the existence of secure elliptic-curve signatures is *not* necessarily conditioned by point counting.

Keywords: Elliptic curve, point counting, signature.

1 Introduction

Point counting is the most complex part of elliptic-curve cryptography which, despite constant improvements, still remains time-consuming and cumbersome (we refer the reader to [3,4,8,9,13,14,15,11,17,20,21] for a comprehensive bibliography about cardinality counting).

Elliptic-curve cryptosystems that would not require point counting are thus theoretically interesting, although, having taken the decision to design such a scheme, one must overcome three technical difficulties :

• If the number of points on the curve ($\#\mathcal{C}$) is unknown to the participants, the protocol must never involve q, the large prime factor of $\#\mathcal{C}$. This excludes the computation of modular inverses modulo q by the signer and the verifier (recall that DSA signatures involve $s = (m + xr)/k \bmod q$ and verifications require $1/s \bmod q$).

• Being unknown, $\#\mathcal{C}$ may be accidentally smooth enough to be vulnerable to Pohlig-Hellman attack [18]. An attacker could then undertake the point counting avoided by the designer, factor $\#\mathcal{C}$ and break-down the Discrete Logarithm Problem's complexity into the much easier tasks of solving DLPs in the various subgroups that correspond to the factors of $\#\mathcal{C}$.

K. Y. Lam, E. Okamoto and C. Xing (Eds.): ASIACRYPT'99, LNCS 1716, pp. 122–134, 1999.

• Finally, even if $\#\mathcal{C}$ has a large prime factor q, the choice of the group generator G (*e.g.* ECDSA's exponentiation base) may still yield a small subgroup vulnerable to discrete logarithm extraction.

Sections 2, 3 and 4 will develop separately each of these issues which will be assembled as a consistent, point-counting-free cryptosystem in section 5. By easing considerably key-generation, our protocol will extend the key-range of elliptic-curve cryptosystems and open new research perspectives.

2 Poupard-Stern's *q*-free DSA

In Eurocrypt'98, Poupard and Stern [19] presented a DSA-like scheme that combines DLP-based provable security, short identity-based keys, very low transmission overhead and minimal on-line computations. By opposition to other Schnorr-like schemes, Poupard-Stern's protocol uses the order of the multiplicative group q *only* for system setup (figure 1).

System parameters	
	primes p and q such that $q\|(p-1)$
	$g \in \mathbb{Z}/p\mathbb{Z}$ of order q
	a hash function $h : \{0,1\}^* \rightarrow \mathbb{Z}/q\mathbb{Z}$
Key generation	
	secret : $x \in_R \mathbb{Z}/q\mathbb{Z}$
	public : $y = g^{-x} \bmod p$
Signature	
	pick a large random k
	$r = g^k \bmod p$
	$s = k + x \times h(m, r)$
	signature : $\{r, s\}$
Verification	
	check that $r \stackrel{?}{=} g^s y^{h(m,r)} \bmod p$

Fig. 1. Poupard-Stern signatures.

We refer the reader to [19] for a precise definition of the system parameters (*e.g.* the size of k), a formal security proof and a description of the scheme's implementation trade-offs.

Elliptic-curve generalization is straightforward : let p be the size of the underlying field (or ring) on which the curve is defined (a prime, an RSA modulus or 2^n); when p is a prime or an RSA modulus the equation of the curve $\mathcal{C}$, characterized by a and b, is given by $y^2 = x^3 + ax + b$; the curve will be defined by

$y^2 + xy = x^3 + ax + b$ when the underlying field is $\mathrm{GF}(2^n)$. In the elliptic curve Pourpard-Stern signature scheme, $p - 1$ and q are respectively replaced by $\#\mathcal{C}$ and one of its large prime factors (figure 2).

System parameters	
	a prime q
	an elliptic curve $\mathcal{C}$ such that $q \mid \#\mathcal{C}$
	$G \in \mathcal{C}$ of order q
	a hash function $h : \{0,1\}^* \rightarrow \mathbb{Z}/q\mathbb{Z}$
Key generation	
	secret : $x \in_R \mathbb{Z}/q\mathbb{Z}$
	public : $Y = -xG$
Signature	
	pick a large random k
	$R = kG = (x_R, y_R)$
	$s = k + x \times h(m, x_R)$
	signature : $\{x_R, s\}$
Verification	
	compute $R' = sG + h(m, x_R)Y = (x_{R'}, y_{R'})$
	check that $x_R \stackrel{?}{=} x_{R'}$

Fig. 2. Elliptic-curve Poupard-Stern signatures.

Poupard and Stern's security proof can be extended, *mutatis mutandis*, to the elliptic-curve variant; the proof can be consulted in the appendix.

We will now suppress from the above protocol the last references to q; care should be taken to underline that we *do not claim yet* that the resulting protocol (figure 3) is secure.

3 The Expected Smoothness of $\#\mathcal{C}$

As an inescapable consequence of our modification, $\#\mathcal{C}$ may now be smooth enough to be at Pohlig-Hellman's reach. An attacker could then perform the point counting, factor $\#\mathcal{C}$ and reduce the DLP's complexity into the much easier tasks of solving DLPs in the various subgroups that correspond to the different factors of $\#\mathcal{C}$. Moreover, even if $\#\mathcal{C}$ has a large prime factor it may still be divisible by a product π of small primes, allowing the adversary to find a portion of the secret key ($x \bmod \pi$) using Pohlig-Hellman. Using Hesse's theorem, we set $L = \log_2 \lfloor p + 1 - 2\sqrt{p} \rfloor$ and *deliberately* accept that only ℓ bits of the L-bit secret key will actually remain unknown to the attacker.

System parameters	
	a random elliptic curve $\mathcal{C}$
	$G \in_R \mathcal{C}$
	a hash function $h : \{0,1\}^* \rightarrow \{0,1\}^L$
Key generation	
	secret : $x \in_R \{0,1\}^L$
	public : $Y = -xG$
Processing	
	pick a large random k
	$R = kG = (x_R, y_R)$
	$s = k + x \times h(m, x_R)$
	output : $\{x_R, s\}$
Verification	
	compute $R' = sG + h(m, x_R)Y = (x_{R'}, y_{R'})$
	check that $x_R \stackrel{?}{=} x_{R'}$

Fig. 3. q-free EC variant of Poupard-Stern's protocol.

We consider that a curve is *weak* if all the factors of $\#\mathcal{C}$ are smaller than 2^ℓ (*i.e.* $\#\mathcal{C}$ is 2^ℓ-smooth) where ℓ is a security parameter. The odds of such an event are analyzed in this section under the assumption that $\#\mathcal{C}$ is uniformly distributed over $[p+1-2\sqrt{p}, p+1+2\sqrt{p}]$.

Defining $\psi(x,y) = \#\{n < x, \text{ such that } n \text{ is } y\text{-smooth}\}$, it is known [5,6,7] that, for large x, the ratio :

$$\frac{\psi(x, \sqrt[t]{x})}{x}$$

is equivalent to Dickman's function defined by :

$$\rho(t) = \begin{cases} 1 & \text{if } \ 0 \leq t \leq 1 \\ \rho(n) - \displaystyle\int_n^t \frac{\rho(v-1)}{v} dv & \text{if } \ n \leq t \leq n+1 \end{cases}$$

$\rho(t)$ is thus an approximation of the probability that a $\ell \times t$-bit number is 2^ℓ-smooth; table 1 summarizes the value of ρ for $2 \leq t \leq 10$.

Since $\rho(t)$ is not easy to compute, we will use throughout this paper the exact formula for $t \leq 10$ and de Bruijn's asymptotic approximation [1,2] for $t > 10$:

$$\rho(t) \cong (2\pi t)^{-1/2} \exp\left(\gamma - t\zeta + \int_0^\zeta \frac{e^s - 1}{s} ds\right)$$

where ζ is the positive solution of $e^\zeta - 1 = t\zeta$ and γ is Euler's constant.

t	2	3	4	5	6	7	8	9	10
$\rho(t)$	3.07e-1	4.86e-2	4.91e-3	3.54e-4	1.96e-5	8.75e-7	3.23e-8	1.02e-9	2.79e-11

Table 1. $\rho(t)$ for $2 \leq t \leq 10$.

Table 1 shows that the proportion of weak curves is too high for immediate use : values of t, such as 2 and 3, which would respectively yield 320 and 480-bit field size for $\ell = 160$, correspond to a percentage of 0.3 and 0.05 weak curves. In section 5, we will propose a signature strategy that decreases exponentially these probabilities.

As pointed out earlier, the above assumes that $\#\mathcal{C}$ is distributed uniformly over $[p+1-2\sqrt{p}, p+1+2\sqrt{p}]$. A more accurate result, valid for prime p, was proved by Lenstra in [12] :

Theorem 1. *Denoting by #' the number of isomorphism classes of elliptic curves, and* $\Delta(S) = \#'\{$*elliptic curves* $\mathcal{E}$ *over* F_p *such that* $\#\mathcal{E} \in S \subset \mathbb{N}\}$ *there exist effectively computable positive constants* c_1, c_2 *such that for each prime* $p > 3$, *the following holds :*

- *if for all* $s \in S$, $|s-(p+1)| \leq 2\sqrt{p}$ *then* $\Delta(S) \leq c_1 \#S \sqrt{p}(\log p)(\log\log p)^2$
- *if for all* $s \in S$, $|s-(p+1)| \leq \sqrt{p}$ *then* $\Delta(S) \geq c_2\sqrt{p}(\#S-2)/\log p$

Since all classes have a number of representatives which is roughly p, Lenstra's theorem basically claims that by taking a curve at random, the probability τ_S that its cardinality lies in S satisfies the inequality :

$$\frac{c_3}{\log p} \leq \frac{\tau_S}{\pi_S} \leq c_4(\log p)(\log\log p)^2$$

where π_S denotes the probability of picking an element of S at random in the interval $\left[p-\sqrt{p}, p+\sqrt{p}\right]$. The theorem indicates that (at least if p is prime) when $\mathcal{C}$ is random, the proportion of weak-curves respects Dickman's estimate. We consider this as heuristically satisfactory for further build-up.

4 The Expected Order of the Generator G

Even when $\#\mathcal{C}$ has an prime factor larger than ℓ bits, G could still yield a small subgroup, which would again weaken the scheme.

We refer the reader to [16] for the following theorem :

Theorem 2. *The set of points of an elliptic curve is an abelian group which is either a cyclic group or the product of two cyclic groups.*

Let q be a large prime factor of $r = \#\mathcal{C}$ of multiplicity 1.

• Assume that $\mathcal{C}$ is a cyclic abelian group, isomorphic to $\mathbb{Z}/r\mathbb{Z}$, with generator $g \in \mathcal{C}$. The order d of a random $G = g^\alpha$ is given by $d = r/\gcd(r, \alpha)$. Therefore q does not divide d if and only if α is a multiple of q. The probability that the order of a random G is not divisible by q is thus $1/q$.

• Assume that $\mathcal{C}$ is the product of two cyclic abelian groups, then $\mathcal{C}$ is isomorphic to some product $\mathbb{Z}/r_1\mathbb{Z} \times \mathbb{Z}/r_2\mathbb{Z}$ where r_2 divides r_1 and $r_1 r_2 = r$. For a large prime factor q of r (with multiplicity 1), q divides r_1 but not r_2. Therefore q divides the order of an element of the curve if and only if q divides the order of this element with respect to $\mathbb{Z}/r_1\mathbb{Z}$. This leads back to the first case, and the probability that the order of a random G is not divisible by q is $1/q$ again.

In both cases, the probability that a random choice for G yields a small subgroup is negligible.

5 The New Scheme

The new protocol iterates the signature on a few curves in order to reduce (below an $\epsilon = 2^{-\ell/2}$) the probability that *all* curves will be smooth (figure 4) :

System parameters

σ random elliptic curves $\mathcal{C}_1, \ldots, \mathcal{C}_\sigma$
σ random points $G_1, \ldots, G_\sigma$ such that $G_i \in \mathcal{C}_i$
a hash function $h : \{0,1\}^* \to \{0,1\}^L$

Key generation

secret : σ random integers $x_i \in_R \{0,1\}^L$
public : σ points $Y_i = -x_i G_i$ such that $Y_i \in \mathcal{C}_i$

Signature

for $i = 1$ to σ
  pick a large random k_i
  compute $R_i = k_i G_i \in \mathcal{C}_i = (x_{R_i}, y_{R_i})$
  compute $s_i = k_i + x_i \times h(m, x_{R_i})$
signature : $\{\{x_{R_1}, s_1\}, \ldots, \{x_{R_\sigma}, s_\sigma\}\}$

Verification

for $i = 1$ to σ
  Compute $R'_i = s_i G_i + h(m, x_{R_i}) Y_i = (x_{R'_i}, y_{R'_i})$
  Check that $x_{R'_i} \stackrel{?}{=} x_{R_i}$

Fig. 4. q-free elliptic-curve Poupard-Stern signatures.

The number of necessary iterations σ is given by :

$$\rho(|p|/\ell)^{\sigma} \leq \epsilon \qquad \Rightarrow \qquad \sigma = \left\lceil \frac{\ell}{2\log \rho(|p|/\ell)} \right\rceil$$

and is summarized in table 2 for $\ell = 160$.

The slow-down factor γ between the elliptic curve Poupard-Stern signature scheme and the new scheme (signature generation times) is due to the iteration of the signature on σ curves and the increased complexity of point operations over bigger underlying fields. Since the time complexity of elliptic curve scalar multiplication is in $\mathcal{O}(|p|^3)$, γ is basically given by :

$$\gamma = \sigma \times \left(\frac{|p|}{\ell}\right)^3$$

The slow-down factor is summarized in table 2 for $\ell = 160$.

# of iterations σ	size of p	slow-down	# of iterations σ	size of p	slow-down
20	460 bits	474	10	654 bits	683
19	471 bits	486	9	693 bits	732
18	484 bits	499	8	740 bits	791
17	497 bits	509	7	798 bits	868
16	513 bits	526	6	873 bits	977
15	529 bits	542	5	973 bits	1125
14	548 bits	562	4	1115 bits	1352
13	570 bits	588	3	1337 bits	1746
12	594 bits	613	2	1757 bits	2646
11	622 bits	645	1	2800 bits	5355

Table 2. Protocol trade-offs for $\ell = 160$.

Letting alone the factor γ, the verification times of the new scheme are also slower than usual ECC ones (*e.g.* ECDSA) because of the additional increase in the size of s due to the Poupard-Stern construction.

Note that Poupard-Stern's security proof will still apply to (at least one of) our curves with probability greater than $1-\epsilon \cong 1$. Surprisingly, instances will be either *provably secure* against existential forgery under adaptive chosen message attacks (probability greater than $1-\epsilon$) or *insecure* (probability lower than $\epsilon = 2^{-\ell/2}$) without transiting through intermediate gray areas where security is only conjectured (our ϵ is, of course, not related to [19]'s one).

Although the security proof has not been extended to the case where all curves have the same system parameters (identical p, intersection in G), we conjecture that the resulting scheme (figures 5 and 6) is still secure.

Secret parameters (x_i and k_i) must however remain distinct for every curve, given the (deliberately accepted) risk that the DLP might be easy on *some* of our curves.

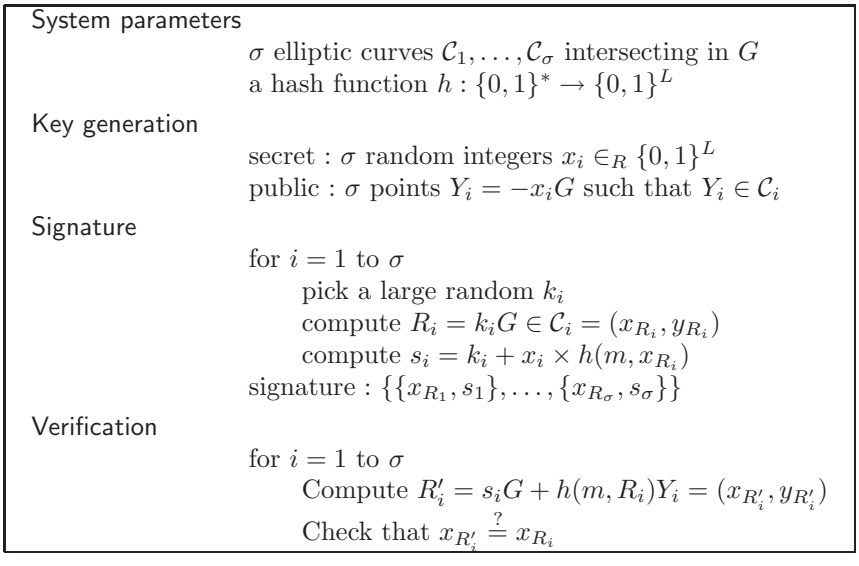

Fig. 5. q-free elliptic-curve Poupard-Stern signatures (common G).

It is important to point-out that, due to our probabilistic design, the *signer* must generate $\mathcal{C}_1, \ldots, \mathcal{C}_\sigma$ or (at least) make sure that the authority can exhibit a random seed (similar to the DSA's *certificate of proper key generation*) that yields all the curves' parameters by hashing.

6 Extensions and Variants

The scheme can be improved in many ways : by hashing $x_i = h'(x, i)$ and $k_i = h''(k, i)$ one can make the economy of $\sigma - 1$ secret keys and session randoms; a particularly efficient variant consists in grouping $\{R_1, \ldots, R_\sigma\}$ in a single digest (figure 7); the scheme can, of course, be implemented on any group.

Note that when p is an RSA modulus (hereafter n), life becomes much harder for the attacker who must (in our present state of knowledge) factor n (equivalent to point counting [10]), compute the orders d_1 and d_2 of the curve modulo the prime factors of n, factor d_1 and d_2 and compute the exact order of G as a multiplicative combination of the prime factors of d_1 and d_2.

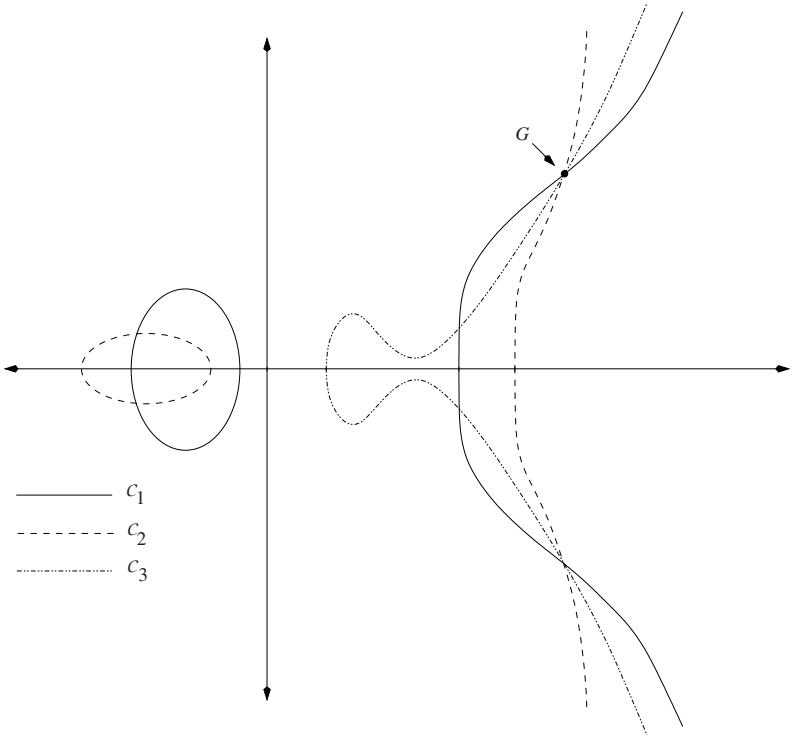

Fig. 6. System configuration (intersecting curves) for $\sigma = 3$.

The overwhelming security contribution comes from the factorisation of n although when this calculation comes to an end, the attacker may face a (non-smooth) curve where the DLP is hard. The attacker's success chances are consequently reduced to :

$$\epsilon' = \rho \left(\frac{|n|}{2\ell} \right)^2$$

for one curve and

$$\epsilon'' = \epsilon'^{\sigma} = \rho \left(\frac{|n|}{2\ell} \right)^{2\sigma}$$

for the σ curves. This indicates an interesting way of squeezing more complexity out of RSA moduli : since (in our present state of knowledge) smooth curves can not be spotted without factoring n, the inverse of ϵ'' represents a *strengthening factor* that multiplies[1] the attacker's effort by a factor depending on $|n|$ and σ (table 3 for $\ell = 160$).

[1] under the discrete logarithm assumption.

System parameters

σ elliptic curves $\mathcal{C}_1, \ldots, \mathcal{C}_\sigma$ intersecting in G
a hash function $h : \{0,1\}^* \to \{0,1\}^L$

Key generation

secret : σ random integers $x_i \in_R \{0,1\}^L$
public : σ points $Y_i = -x_i G$ such that $Y_i \in \mathcal{C}_i$

Signature

for $i = 1$ to σ
    pick a large random k_i
    compute $R_i = k_i G \in \mathcal{C}_i = (x_{R_i}, y_{R_i})$
$r = h(m, x_{R_1}, \ldots, x_{R_\sigma})$
for $i = 1$ to σ
    compute $s_i = k_i + x_i \times r$
signature : $\{r, s_1, \ldots, s_\sigma\}$

Verification

for $i = 1$ to σ
    compute $R'_i = s_i G + r Y_i = (x_{R'_i}, y_{R'_i})$
check that $r \stackrel{?}{=} h(m, x_{R'_1}, \ldots, x_{R'_\sigma})$

Fig. 7. q-free elliptic-curve Poupard-Stern signatures (common G and r).

$-\log_2$ factor $\searrow$	$\lvert n\rvert = 512$	$\lvert n\rvert = 768$	$\lvert n\rvert = 1024$
$\sigma = 1$	1.8	5.3	9.9
$\sigma = 2$	3.6	10.7	19.9
$\sigma = 3$	5.5	16.0	29.8
$\sigma = 4$	7.3	21.4	39.8
$\sigma = 5$	9.1	26.8	49.8
$\sigma = 6$	11.0	32.1	59.7
$\sigma = 7$	12.8	37.5	69.7

Table 3. Strengthening factors for $\ell = 160$ and $1 \leq \sigma \leq 7$.

7 Acknowledgements

The authors are grateful to Jacques Stern for motivating and following the evolution this work; we also thank him for his insights into several mathematical details and for kindly providing the extended proof given in the appendix.

References

1. N. de Bruijn, *On the number of positive integers $\leq x$ and free of prime factos $\geq y$*, *Indagationes Mathematicae*, vol. 13, pp. 50–60, 1951. 125
2. N. de Bruijn, *On the number of positive integers $\leq x$ and free of prime factos $\geq y$, II*, *Indagationes Mathematicae*, vol. 28, pp. 236–247, 1966. 125
3. J.-M. Couveignes, L. Dewaghe & F. Morain, *Isogeny cycles and the Schoof-Elkies-Atkin algorithm*, Rapport de recherche LIX/RR/96/03, Laboratoire d'informatique de l'École Polytechnique, 1996. 122
4. J.-M. Couveignes & F. Morain, *Schoof's algorithm and isogeny cycles*, LNCS 877, ANTS-I, Proceedings of first algorithmic number theory symposium, Springer-Velrag, pp. 43–58, 1994. 122
5. K. Dickman, *On the frequency of numbers containing prime factors of a certain relative magnitude*, Arkiv för matematik, astronomi och fysik, vol. 22A(10), pp. 1–14, 1930. 125
6. J. Dixon, *Asymptotically fast factorization of integers*, Mathematics of computation, vol. 36(153), pp. 255–260, 1981. 125
7. H. Halberstam, *On integers whose prime factors are small*, Proceedings of the London mathematical society, vol. 3(21), pp. 102–107, 1970. 125
8. E. Howe, *On the group orders of elliptic curves over finite fields*, *Compositio mathematica*, vol. 85, pp. 229–247, 1993. 122
9. N. Koblitz, *Primality of the number of points on an elliptic curve over a finite field*, Pacific journal of mathematics, vol. 131, pp. 157–165, 1988. 122
10. N. Kunihiro & K. Koyama, *Equivalence of counting the number of points on elliptic curve over the ring $\mathbb{Z}_n$ and factoring n*, LNCS 1403, Advances in cryptology - proceedings of EUROCRYPT'98, Springer-Verlag, pp. 47–58, 1998. 129
11. G. Lay & H. Zimmer, *Constructing elliptic curves with given group order over large finite fields*, LNCS 877, ANTS-I, Proceedings of first algorithmic number theory symposium, Springer-Velrag, pp. 250–263, 1994. 122
12. H. Lenstra Jr., *Factoring integers with elliptic curves*, Ann. math., vol. 126, pp. 649–673, 1987. 126
13. R. Lercier, *Computing isogenies in GF(2^n)*, LNCS 1122, ANTS-II, Proceedings of 2-nd algorithmic number theory symposium, Springer-Velrag, pp. 197–212, 1996. 122
14. R. Lercier & F. Morain, *Counting the number of points on elliptic curves over finite fields : strategies and performances*, LNCS 921, Advances in cryptology - proceedings of EUROCRYPT'95, Springer-Verlag, pp. 79–94, 1995. 122
15. R. Lercier & F. Morain, *Counting the number of points on elliptic curves over F_{p^n} using Couveigne's algorithm*, Rapport de recherche LIX/RR/95/09, Laboratoire d'informatique de l'École Polytechnique, 1995. 122
16. A. Menezes, *Elliptic curve public key cryptosystems*, Kluwer academic publishers, pp. 25, 1983. 126
17. A. Menezes, S. Vanstone & R. Zuccharato, *Counting points on elliptic curves over F_{2^m}*, Mathematics of computation, vol. 60(201), pp. 407–420, 1993. 122

18. S. Pohlig & M. Hellman, *An improved algorithm for computing logarithms over GF(p) and its cryptographic significance*, IEEE Transactions on Information Theory, Vol. 24, pp. 106-110, 1978. 122

19. G. Poupard & J. Stern, *A practical and provably secure design for on the fly authentication and signature generation*, LNCS 1403, Advances in cryptology - proceedings of EUROCRYPT'98, Springer-Verlag, pp. 422–436, 1998. 123, 128, 133

20. R. Schoof, *Elliptic curves over finite fields and the computation of square roots mod p*, Mathematics of computation, vol. 44, pp. 483–494, 1985. 122

21. R. Schoof, *Counting points on elliptic curves over finite fields*, CACM, vol. 21(2), pp. 120-126, 1978. 122

APPENDIX

Using Poupard and Stern's notations, the following is a generalization of [19]'s security proof :

Theorem 3. *Assume that* $kS\tau/X$ *and* $1/k$ *are negligible. If an existential forgery of the signature scheme under adaptive chosen message attack has a non-negligible success probability then the discrete logarithm on elliptic curves can be computed in a time polynomial in* $|q|$.

The proof is based on the same 3-fork variant of Pointcheval-Stern's forking lemma. However, there is a technical difficulty; in the modular case studied by Poupard and Stern, the authors deal with an RSA modulus $n = pq$, assuming that g is of order $\lambda(n) = \mathrm{GCD}(p-1, q-1)$. Their proof includes three steps :

1. compute a multiple L of $\lambda(n)$.

2. factor n, using L and a number-theoretic algorithm due to Miller.

3. finally, use the 3-fork variant of the forking lemma to yield a couple of relations involving the unknown key s :

$$\alpha s + \beta = 0 \bmod \lambda(n) \quad \text{and} \quad \alpha' s + \beta' = 0 \bmod \lambda(n)$$

such that for some polynomial B, which only depends on the machine which presumably performs the existential forgery, $\mathrm{GCD}(\alpha, \alpha') \leq B$; from these equations, s can be computed in polynomial time.

In the elliptic curve case, there is no analog to step 2; which requires a further twist :

1. compute a multiple ρ of the (unknown) order r of G, which is approximately $|X| + |k|$-bit long.

2. use the forking lemma's 3-fork variant to yield a couple of relations involving s :

$$\alpha s + \beta = 0 \bmod \lambda(n) \quad \text{and} \quad \alpha' s + \beta' = 0 \bmod \lambda(n)$$

such that for some polynomial B, which only depends on the machine which presumably performs the existential forgery, $\mathrm{GCD}(\alpha, \alpha') \leq B$. Furthermore, we cancel all primes smaller than B from ρ. From these relations and ρ, one can compute a substitute to s which satisfies $V = -sG$ without being in the proper range.

3. Finally, we show that an algorithm which computes a substitute of s and a multiple of r with significant probability can be turned into an algorithm which computes the proper value of s :

Lemma 1. *An algorithm $\mathcal{A}$ which computes with significant probability a fixed-size multiple ρ of the unknown order r of G and a substitute to the secret key $s < \rho$ can be turned into an algorithm $\mathcal{B}$ which computes the proper value of s.*

PROOF Let ϵ be the success probability of $\mathcal{A}$ and fix $\delta = \epsilon/|\rho|$. By induction on $|\rho|$, we show how to design an algorithm $\mathcal{B}$ which discloses the actual key with probability at least δ : Apply $\mathcal{A}$ to $V = -sG$, where s is in the proper range for keys. $\mathcal{A}$ could either output s with probability δ (in which case the proof is complete) or it outputs a substitute $s' \neq s$ with probability bigger than $(\rho - 1)\delta$. In this case, we can consider $s' - s$ and $\rho - s' + s$; both are multiples of r and one of them (hereafter ρ') is smaller than $\rho/2$. Note that $\mathcal{A}$ produces, with probability $\delta|\rho'|$ a multiple ρ' of r. Furthermore, it also produces substitute keys smaller than ρ', since one can always replace a substitute s by $s \bmod \rho'$; we can thus apply the inductive hypothesis, which completes the proof. □