Efficient Arithmetic on Hyperelliptic Koblitz Curves

Tanja Lange

Institute of Experimental Mathematics
University of Essen
Ellernstrasse 29
D-45326 Essen
Germany

lange@exp-math.uni-essen.de

June 2, 2001

CIP-DATA KONINKLIJKE BIBLIOTHEEK, DEN HAAG
Efficient Arithmetic on Hyperelliptic Koblitz Curves
Juni 2001, Essen, Germany
Autéur: Tanja Lange
With refs.
ISBN 90-74249-25-6
Publisher: Shannon Foundation

Due to the emerging market of electronic commerce public key cryptosystems gain more and more attention. Unlike for military purposes there is a need of flexible user groups. Besides RSA most cryptosystems and protocols like the Diffie-Hellman key exchange [1] and the ElGamal cryptosystem [3] are based on the discrete logarithm as the underlying one-way function. Given a cyclic subgroup of an abelian group generated by g and an integer m one can compute $g^m = b$. If $\langle g \rangle$ is a group suitable for cryptographic applications then it is computationally hard to retrieve m for given b and g. m is called the *discrete logarithm* of b to the base g. The problem of determining m given b and g is called the *discrete logarithm problem*. A group is suitable if

1. the group operation is fast,
2. the group order can be computed efficiently,
3. the discrete logarithm problem is hard,
4. the representation is easy and compact.

Two common kinds of groups used in practice are the multiplicative group of a finite field and the group of points on an elliptic curve over a finite field. To obtain a speed-up for the main operation on an elliptic curve – computing m-folds – Koblitz [11] proposed the use of a special kind of curves. These *Koblitz* or *subfield* curves are curves defined over a comparably small finite field $\mathbf{F}_q$. They are then considered as curves over a large extension field $\mathbf{F}_{q^n}$, where n is prime. The arithmetic makes use of the fact that if the curve C is defined over $\mathbf{F}_q$ and $P = (x, y) \in \mathbf{F}_{q^n} \times \mathbf{F}_{q^n}$ lies on C then the point $\sigma(P) = (x^q, y^q)$ lies on C, too. σ is an endomorphism of the curve called the Frobenius endomorphism. These curves have thoroughly been studied by Koblitz [11, 12], Meier and Staffelbach [16], Müller [17], Smart [19], and Solinas [20, 21], where the last reference contains a detailed analysis of the maximal speed-up achievable for curves over $\mathbf{F}_2$.

In [10] Koblitz proposed the Picard group $\mathrm{Pic}^0(C/\mathbf{F}_q)$ of a hyperelliptic curve as a further group suitable for cryptographic applications. For genus ≤ 4 these groups are secure provided that the group order is sufficiently large and that one avoids curves for which special attacks are known. The advantages over the elliptic curves are the smaller field size and the larger variety of curves to choose from. Due to Mumford there is a representation of the group elements given by two polynomials of degrees bounded by $g + 1$ and g respectively, the group satisfies the requirement of 4. But there are several disadvantages:
At the moment no-one is able to compute the group order of a randomly generated hyperelliptic curve over a prime field with group order $\sim 2^{160}$. The best result obtained for curves of genus two is a curve over the prime field $\mathbf{F}_p$ with $p = 10^{19} + 51$ by Gaudry and Harley [8] which leads to a group order $\sim 10^{38} \sim 2^{129}$ which is smaller than recommended for cryptographic applications. Hence, one is forced to take special curves. In this article we investigate *hyperelliptic Koblitz curves*. The idea of elliptic Koblitz curves was generalized by Günther, Lange, and Stein [9]. There we investigate two special examples of binary curves of genus 2. We show in that paper that also in the hyperelliptic case the Frobenius endomorphism can be used to achieve fast arithmetic, i. e. to speed up scalar multiplication. The Frobenius endomorphism operates on the divisor classes $D = [a(x), b(x)]$ in Mumford representation by raising the coefficients of the polynomials a and b to the qth power, i. e. $\sigma(D) = [\sigma(a), \sigma(b)]$ and $\sigma(\sum a_i x^i) = \sum a_i^q x^i$. Hence, if the finite field $\mathbf{F}_{q^n}$ is represented via a normal

basis over $\mathbf{F}_q$ then computing the qth power of a field element just means a cyclic shift of its representation. Thus, the computation of $\sigma(D)$ is performed by at most $2g$ cyclic shifts. We now give evidence that the Frobenius endomorphism gives rise to a speed-up of at least a factor of 4 (for $q = g = 2$) and much more if many precomputations can be stored. The speed-up increases with q and g. In the following we state the results without proofs. For details consult Lange [14].
Let the hyperelliptic curve of genus g be given by

$$C : y^2 + h(x)y = f(x),\ \ h, f \in \mathbf{F}_q[x]$$

and consider the curve over $\mathbf{F}_{q^n}$. Let the characteristic polynomial of the Frobenius endomorphism of the Picard group $\mathrm{Pic}^0(C/\bar{\mathbf{F}}_q)$ be

$$P(T) = T^{2g} + a_1 T^{2g-1} + \cdots + a_g T^g + \cdots + a_1 q^{g-1} T + q^g.$$

For the divisor classes D we have that

$$\begin{aligned} q^g D &= -\sigma^{2g}(D) - a_1\sigma^{2g-1}(D) - \cdots - a_g\sigma^g(D) - \cdots - a_1 q^{g-1}\sigma(D) \\ &= -\sigma(\cdots\sigma(\sigma(\sigma(D) + a_1 D) + a_2 D) + \cdots + a_1 q^{g-1} D). \end{aligned}$$

We now compute for each integer m a representation of mD as $mD = \sum_{i=0}^{l-1} u_i \sigma^i(D)$ where we restrict the coefficients u_i to a set R. Let τ be a complex root of $P(T)$. Then to any representation as given above corresponds an expansion of $m = \sum_{i=0}^{l-1} u_i \tau^i$, $u_i \in R$. Given such a representation and a precomputed table containing $u_i D$ for all $a_i \in R$ we can compute mD like in the double-and-add method where the doublings are replaced by cyclic shifts, hence are for free, and the multiples $u_i D$ are stored in a precomputed table.
The key to compute an expansion of m is the following lemma. Note that the elements of $\mathbf{Z}[\tau]$ are of the form $c = c_0 + c_1\tau + \cdots + c_{2g-1}\tau^{2g-1}$ with $c_i \in \mathbf{Z}$.

Lemma 1 *$c = c_0 + c_1\tau + \cdots + c_{2g-1}\tau^{2g-1}$ is divisible by τ if and only if $q^g | c_0$.*

Therefore the minimal set of remainders R consists of a complete set of representatives of $\mathbf{Z}/q^g\mathbf{Z}$. Since taking the negative of a divisor class is essentially for free (to $-D$ corresponds $[a, h - b]$) we will use $R = \{0, \pm 1, \pm 2, \ldots, \pm\lceil \frac{q^g - 1}{2} \rceil\}$ if just a representation is needed. Note that we would not need to include $-q^g/2$ in the case of even characteristic. But since we get it for free we will make use of it.
The representation of m is computed making repeatedly use of

$$\begin{aligned} c &= c_0 + c_1\tau + \cdots + c_{2g-1}\tau^{2g-1} = u_0 + (c_0 - u_0) + c_1\tau + \cdots + c_{2g-1}\tau^{2g-1} \\ &= u_0 + \tau(c_1 - d a_1 q^{g-1} + (c_2 - d a_2 q^{g-2})\tau + \cdots + (c_{2g-1} - d a_1)\tau^{2g-2} - d\tau^{2g-1}), \end{aligned}$$

where $d = (c_0 - u_0)/q^g$ and $u_0 \in R$ is chosen such that $q^g | (c_0 - u_0)$.
Using the norm $\mathcal{N}(c) = \sqrt{\sum_{i=1}^{g} \left| \sum_{j=0}^{2g-1} c_j \tau_i^j \right|^2}$ for $c \in \mathbf{Z}[\tau]$ we have

Lemma 2 *Let q be odd. For every $m \in \mathbf{Z}[\tau]$ we have an unique expansion*

$$m = \sum_{i=0}^{k-1} u_i \tau^i + m'\tau^k,$$

where $u_i \in \{0, \pm 1, \pm 2, \dots, \pm \frac{q^g-1}{2}\}$,

$$\mathcal{N}(m') < \frac{\sqrt{g}}{2} \frac{q^g}{\sqrt{q}-1} =: K,$$

and

$$k \leq \lceil 2 \log_q \frac{2(\sqrt{q}-1)\mathcal{N}(m)}{\sqrt{g}} \rceil + 1.$$

And for even characteristic the same holds true with $\frac{q^g}{\sqrt{q}-1}$ replaced by $\frac{q^g+1}{\sqrt{q}-1}$. $\mathcal{N}^2$ is a positive definite quadratic form of $2g$ variables. Hence, we can use the algorithm of Finke and Pohst [4] to find all elements of norm smaller than K. To prove the finiteness of the representations for an individual curve it suffices to expand all these elements to the base of τ. The experiments show that either the expansion is cyclic of period length one (and perhaps a change of sign) or it is finite of length at most $2g+1$. In [14] we investigate in detail the case of genus two curves supporting these observations. In case the expansion runs into a cycle we can still use the curve if we allow one more precomputation – for the value of c_0 where the period starts. A period of length one can only occur if for the curve or its twist we have $\lceil (q^g-1)/2 \rceil \geq d|\mathrm{Pic}^0(C/\mathbf{F}_q)|$. In the experiments only $d=1$ occurred. Thus q has to be fairly small and the time needed to compute the additional coefficient $\pm d(q^g - |\mathrm{Pic}^0(C/\mathbf{F}_q)|)$ can be neglected.

Example 3 *Put* $g=2, q=3$. *Among all the isogeny classes of curves with irreducible* $P(T)$ *only* $P(T) = T^4 \pm 2T^3 + 2T^2 \pm 6T + 9$, $P(T) = T^4 \pm T^3 - 2T^2 \pm 3T + 9$, *and* $T^4 \pm 3T^3 + 5T^2 \pm 9T + 9$ *lead to periodicities. The coefficients to include are* 5 *in the first two cases and* 6 *in the last one.*
In the case of even characteristic the situation is even a bit more relaxed. If we choose coefficients from $\{0, \pm 1, \dots, \pm q^g/2 - 1, q^g/2\}$ *unless* $c_0 = -q^g/2$ *then for all classes of curves of genus two over* $\mathbf{F}_2$ *the expansions are finite. For* $\mathbf{F}_4$ *we run into a cycle only for* $P(T) = T^4 \pm 4T^3 + 9T^2 \pm 16T + 16$. *To deal with this we include* ± 10 *in the set of coefficients.*

If we restrict ourselves to the group $\mathrm{Pic}^0(C/\mathbf{F}_{q^n})$ we additionally have $\sigma^n(D) = D$, hence two τ-adic expansions represent the same endomorphism if the are equivalent modulo $\tau^n - 1$. On the subgroup of prime order $l \mid |\mathrm{Pic}^0(C/\mathbf{F}_{q^n})|$ we also have that the Frobenius endomorphism cannot equal the identity thus we can even reduce modulo $(\tau^n - 1)/(\tau - 1)$. On these points of order l we have that $\sigma(D) = sD$ for an integer $s \bmod l$.

Theorem 4 *Let* τ *be a root of the characteristic polynomial* $P(T)$ *of the Frobenius endomorphism of the hyperelliptic curve* C *defined over* $\mathbf{F}_q$. *Consider the curve over* $\mathbf{F}_{q^n}$ *and let* $m \in \mathbf{Z}$. *There is an element* $M \in \mathbf{Z}[\tau]$ *such that*

1. $m \equiv M \bmod (\tau^n - 1)/(\tau - 1)$, *and*
2. $$2 \log_q \frac{2(\sqrt{q}-1)\mathcal{N}(M)}{\sqrt{g}} < n + 2g.$$

To compute $(T^n - 1)/(T - 1)$ in $\mathbf{Z}[\tau]$ we use a recursion. The inversion can be performed using the extended greatest common divisor of $(T^n - 1)/(T - 1)$ and $P(T)$.

Algorithm 5 (Computation of m-folds using τ-adic expansions)
`INPUT:` $m \in \mathbf{Z}, D = [a, b], a, b \in \mathbf{F}_{q^n}[x]$, $P(T)$, R *the set of coefficients.*
`OUTPUT:`mD *represented by the reduced ideal* $H = [s, t], \ s, t \in \mathbf{F}_{q^n}[x]$.

1. *Precomputation: for* $i \in R, i > 0$ *compute*
   $D(i) := iD$;
   $D(-i) := -D(i)$; /* *for free*/*
2. /**compute a length reduced* $M \in \mathbf{Z}[\tau]$ *with* $m \equiv M \bmod (\tau^n - 1)/(\tau - 1)$;*/
   (a) *Initialize:* $d_0 = 1$ *and* $d_i = 0$ *for* $1 \le i \le 2g - 1$;
   $e_0 = 1$ *and* $e_i = 0$ *for* $1 \le i \le 2g - 1$;
   (b) *for* $1 \le k \le n - 1$ *do*
   i. $d_{old} := d_{2g-1}$;
   ii. *for* $2g - 1 \ge i \ge g$ *do*
   $d_i := d_{i-1} - a_{2g-i} d_{old}$;
   $e_i := e_i + d_i$;
   iii. *for* $g - 1 \ge i \ge 1$ *do*
   $d_i := d_{i-1} - a_i q^{g-i} d_{old}$;
   $e_i := e_i + d_i$;
   iv. $d_0 := -q^g d_{old}$;
   $e_0 := e_0 + d_0$;
   (c) *let* $e := \sum e_i T^i$;
   (d) *compute* $e' := e^{-1} \bmod P$ *using extended GCD;*
   (e) *compute* $M' :=\texttt{round}(m \cdot e')$;
   (f) *let* $M = \sum_{i=0}^{2g-1} M_i T^i := m - e \cdot M' \bmod P$;
3. /**compute the* τ*-adic representation of* M;*/
   (a) *Put* $i := 0$;
   (b) *While for any* $0 \le j \le 2g - 1$ *there exists an* $M_j \ne 0$ *do*
   if $q^g | M_0$ *choose* $u_i := 0$;
   else choose $u_i \in R$ *with* $q^g | M_0 - u_i$;
   /**in even characteristic choose* $u_i = M_0$ *if* $|M_0| = q^g/2$/*
   $d := (M_0 - u_i)/q^g$;
   for $0 \le j \le g - 1$ *do*
   $M_j := M_{j+1} - a_{j+1} q^{g-j-1} d$;
   for $0 \le j \le g - 2$ *do*
   $M_{g+j} := M_{g+j+1} - a_{g-j-1} d$;
   $M_{2g-1} := -d$;
   $i := i + 1$;
4. /* *compute* m*-fold of* D;*/
   (a) *initialize* $H := [1, 0]$;
   (b) *for* $l - 1 \le 0$ *do*
   $H := \sigma(H)$; /* *this means cyclic shifting* /*
   if $u_i \ne 0$ *then*
   $H := H + D(u_i)$;

5. *output(H).*

The routine `round` computes for an element of $\mathbf{Q}[\tau]$ the nearest element of $\mathbf{Z}[\tau]$ in the sense that the coefficients are rounded to the nearest integer. If the algorithm is carried out several times with the same divisor class D (like in the first step of the Diffie-Hellman key exchange) then we need to do the precomputations of Step 1 and the determination of e' (i.e. most of Step 2) only once and for all at the set-up of the system.
For estimates on the complexity we need the following theorem.

Theorem 6 (Main result on the Length)
Let C be a hyperelliptic curve of genus g and with characteristic polynomial of the Frobenius endomorphism $P(T)$. Let P be such that the τ-adic expansion is not periodic and that for an element c of $\mathbf{Z}[\tau]$ of norm $< \frac{g}{4}\left(\frac{q^g}{\sqrt{q}-1}\right)^2$ (respectively $< \frac{g}{4}\left(\frac{q^g+1}{\sqrt{q}-1}\right)^2$ for even characteristic) the τ-adic expansion is no longer than $2g+1$. Then we have:
For every element $m \in \mathbf{Z}$ we can compute a τ-adic expansion of length k using coefficients in the set R only, where

$$k \leq n + 4g + 2.$$

Besides the length the second important quantity to consider is the *density* of the representation. By density we mean the number of nonzero coefficients occurring in the representation divided by the length of the representation.
Naturally the density will depend heavily on the choice of the set R and therefore on the number of precomputations. As stated before the minimal set R simply to make possible the expansion is $R = \{0, \pm 1, \pm 2, \ldots, \pm\lceil\frac{q-1}{2}\rceil\}$. Using this set, we get a zero coefficient only at random, hence with a probability of $1/q^g$. Therefore the asymptotic density in that case is $(q^g-1)/q^g$ resulting in a complexity of $(q^g-1)/q^g n < n$.
If

$$P(T) \equiv T^{2g} + a_g T^g + q^g \bmod q^g, \quad a_g \neq 0$$

then we can also use a much larger set of coefficients, namely for q even $\tilde{R} = \{0, \pm 1, \pm 2, \ldots, \pm\frac{q^{2g}}{2} - 1\} \backslash \{q^g, 2q^g, \ldots, (q^g-1)q^g\}$ and for q odd $\tilde{R} = \{0, \pm 1, \pm 2, \ldots, \pm\frac{q^{2g}-1}{2}\} \backslash \{q^g, 2q^g, \ldots, (q^g-2)q^g\}$ of size $|\tilde{R}| = (q^g-1)q^g$, that guarantees that for each non-zero coefficient we obtain at least one zero. This results in an asymptotic density of $\frac{q^g-1}{2q^g-1}$ and a complexity of $\frac{q^g-1}{2q^g-1} n < \frac{1}{2} n$.
All usual windowing techniques carry through to τ-adic windowing, i.e. using $b_0 + b_1\tau + \cdots + b_i\tau^i$ as coefficients, as well. Using $b_0 + b_1\tau, b_i \in R$, R as above also leads to the density $\frac{q^g-1}{2q^g-1}$.
Since for integers $m \leq |\mathrm{Pic}^0(C/\mathbf{F}_{q^n})| \sim q^{gn}$ the binary expansion has a complexity of $3/2gn\log_2(q)$, the speed-up obtained using the first set of coefficients is $\frac{3}{2}g\log_2 q$ and for the larger set $3g\log_2 q$.
Even if one compares the τ-adic method to the usual binary windowing method with at least the same number of precomputations the speed-up is of order at least g respectively $2g$.

For timings we used the binary curve $C: y^2+(x^2+x+1)y = x^5+x^4+1$ with characteristic polynomial $P(T) = T^4-2T^3+3T^2-4T+4$ over $\mathbf{F}_{2^{89}}$. Its class number

is $2 \cdot 191561942608242456073498418252108663615312031512914969$, thus this curve is appropriate for applications. For the computations we used Magma. Unfortunately Magma does not provide a representation of the finite fields using a normal basis. Thus instead of using the cyclic shifting as proposed we raise each coefficient to the q-th power. Thus we cannot get the whole speed-up.
We carried out 1000 random scalar multiplications using the τ-adic method in Magma. For the τ-adic method we needed only one precomputation for $2D$, thus the time and space needed for this is negligible. To compare we also used the built-in routine for computing m-folds in Magma.
The average length of the τ-adic expansion is 90.18 and the average time to compute the expansion is 0.005318. The complete multiplication takes 0.070261 on average. The corresponding time with the usual function is 0.146036 on average. Hence, we obtained a speed-up by a factor of 2.
The program used for this comparison `FrobExample` and a program to play around with a self-defined curve `FrobSelf` can be obtained from
`http://www.exp-math.uni-essen.de/ lange/KoblitzC.html`.
To save the time needed to compute the expansion it is also possible to use an alternative set-up where instead of computing the expansion of a given integer one uses a string of length $n-1$ of elements from R as a key which is then identified with an expansion. In the subgroup of $\mathrm{Pic}^0(C/\mathbf{F}_{q^n})$ of order l we need not worry whether this expansion belongs to an integer since the operation of the Frobenius endomorphism corresponds to a multiplication by s modulo l and, hence, there is always a corresponding integer modulo l. Note that although the key-space is reduced this does not weaken the system unless for a brute force attack. To make sure that the multipliers occurring are equally distributed we do some experiments before choosing the curve. Even if some vectors represent the same integer modulo l, the keys are still almost equally distributed.
We consider again the above curve over $\mathbf{F}_{2^{89}}$ and let l be the large prime factor of the class number. The operation of the Frobenius endomorphism on the cyclic group of this prime order corresponds to the multiplication by $s = -10909476359861941088449855420776379666052262767680104 1 \bmod l$. Choosing a sequence of 88 elements u_i from $R := \{-1, 0, 1, 2\}$ at random and computing $\sum_{i=0}^{87} u_i s^i \bmod l$ we get the multiplier of D corresponding to the key $(u_0, \ldots, u_{87})$. If two sums represent the same integer modulo l then their difference has coefficients in $0, \pm 1, \pm 2, \pm 3$. To get the correct probabilities of occurrence we used the following multi-set $U := \{-3, -2, -2, -1, -1, -1, 0, 0, 0, 0, 1, 1, 1, 2, 2, 3\}$ and computed 10,000,000 such sums modulo l. The zero sum never occurred.
Hence, there are no obvious weaknesses and this curve is probably suitable for using this modified set-up.

A further important advantage of Koblitz curves is that due to the construction the group order can be determined very efficiently. A complete list of all isogeny classes of imaginary quadratic hyperelliptic curves of genus 2,3, and 4 for $\mathbf{F}_2$ and $\mathbf{F}_3$ and of genus 2 and 3 for $\mathbf{F}_4$ and $\mathbf{F}_5$ together with their class numbers can be obtained from
`http://www.exp-math.uni-essen.de/~lange/Koblitz.html.`
The computations of the class numbers were carried out using only integer

arithmetic by some recurrence sequences developed in [14]. One finds that among the Koblitz curves there are many providing a group of cryptographic relevance.

Hence, firstly the computation of m-folds is sped up considerably and can thus be regarded as fast. Secondly the group order can be computed very easily. The group elements can be represented by two polynomials of degree at most g over $\mathbf{F}_{q^n}$, thus the representation is compact and easy.
To the third point: The Picard group of Koblitz curves over $\mathbf{F}_{q^n}$ comes along with an automorphism group of order at least $2n$ – due to the Frobenius automorphism of order n and inversion. This can be used for cryptanalysis. The attack of Gallant, Lambert, and Vanstone [5] designed for elliptic curves was extended to hyperelliptic curves. Duursma, Gaudry, and Morain [2] make use of equivalence classes in Pollard's rho method and obtain a speed-up of $\sqrt{n}$ compared to a Picard group without automorphisms except for the inversion. This can be dealt with by choosing n some bits larger (at most 4 bits in the range considered here). Gaudry [7] used this automorphism group to speed-up his variant of the index-calculus method by n^2. But as he remarks in his thesis [6] the space consuming linear algebra step works only for genus ≥ 4.

Remarks:

1. Although our approach is described for curves over arbitrary fields and of arbitrary genus, in applications they are most likely used over small fields with $q \leq 7$ and genus 2, 3 or 4, since for larger genus the groups are insecure and for larger field size the number of precomputations to be stored increases, and we loose too much due to inevitable factors of the group order. Furthermore one has to be aware of Weil descent attacks if the degree of extension gets too small.

2. We only consider the case of hyperelliptic curves, but all this generalizes to arbitrary abelian varieties, thus especially to those attached to C_{ab}-curves, as soon as the action of the Frobenius endomorphism can be used efficiently. This holds since we only work with the characteristic polynomial not with the curves themselves.

3. When choosing a curve for "real-life" application one should not only look for the right order and the other security issues pointed out here but also make sure that the finite field is such that the arithmetic can be performed efficiently. Thus the choice of curves – or more correctly field extensions – is reduced. First of all we need to ensure that we are working in a field for which a normal basis exists such that the arithmetic of the field is not significantly slower than for a polynomial basis with a sparse polynomial. Using Gauss periods and – if necessary – working with a polynomial basis of a small extension field one obtains a field arithmetic much faster than using a matrix based multiplication. Furthermore it is also possible to use the Frobenius automorphism of the finite field for the arithmetic in the ground field. This is extremely interesting if one works in characteristic 2 since then squarings in the usual square and multiply method are for free. A generalization to composite Gauss periods was recently investigated by

Nöcker [18]. It is a topic of current research to find optimal choices for a pair curve and finite field. For hardware implementations it is also useful to work over fields of characteristic 2.

After finishing this paper it was brought to our attention that Lee [15] has also generalized the results of Günther, Lange, and Stein [9] to arbitrary characteristic. His paper does not contain a proof of the finiteness and length of the representations obtained. Furthermore he uses larger ground fields than we recommend and does not use the full power of the Frobenius endomorphism since he uses only a polynomial basis and precomputes the needed powers $\sigma^i(D)$.

Acknowledgments: Work supported by DFG Graduiertenkolleg "Cryptography"

References

[1] W. Diffie, M. E. Hellman, New Directions in Cryptography, *Mathematics of Computation* **48** (1976), 95-101.

[2] I. Duursma, P. Gaudry, F. Morain, Speeding up the discrete log computation on curves with automorphisms, in: *Advances in Cryptology, Asiacrypt'99*, Lecture Notes in Computer Science **1716**, (Springer 1999), 103-121.

[3] T. ElGamal, A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, *IEEE Transactions on Information Theory* **IT-31** (1985), 469-472.

[4] U. Finke, M. Pohst, Methods for Calculating Vectors of Short Length in a Lattice, *Mathematics of Computations* **44** (1985), 463-482.

[5] R. Gallant, R. Lambert, S. Vanstone, Improving the Parallelized Pollard Lambda Search on Anomalous Binary Curves, *Mathematics of Computation* **69** (2000), 1699-1705.

[6] P. Gaudry, Algorithmique des courbes hyperelliptiques et applications à la cryptologie, *Ph.D. Thesis*, (2000).

[7] P. Gaudry, An algorithm for solving the discrete log problem on hyperelliptic curves, in: *Advances in Cryptology, Eurocrypt'2000*, Lecture Notes in Computer Science **1807**, (Springer 2000), 19-34.

[8] P. Gaudry, R. Harley, Counting points on hyperelliptic curves over finite fields, in: *Algorithmic Number Theory Seminar ANTS-IV*, Lecture Notes in Computer Science **1838**, (Springer 2000), 313-332.

[9] C. Günter, T. Lange, A. Stein, Speeding up the Arithmetic on Koblitz Curves of Genus Two, in: *Selected Areas in Cryptography SAC 2001*, Lecture Notes in Computer Science **2012**, (Springer 2001), 106-117; see also *University of Waterloo Technical Report CORR 00-04* (2000).

[10] N. Koblitz, Hyperelliptic Cryptosystems, *Journal of Cryptology* **1** (1989), 139 - 150.

[11] N. Koblitz, CM-curves with good cryptographic properties, in: *Advances in Cryptology - Crypto '91*, Lecture Notes in Computer Science **576**, (Springer 1992), 279-287.

[12] N. Koblitz, An Elliptic Curve Implementation of the Finite Field Digital Signature Algorithm, in: *Advances in Cryptology - Crypto '98*, Lecture Notes in Computer Science **1462**, (Springer 1998), 327-337.

[13] N. Koblitz, *Algebraic Aspects of Cryptography*, (Springer 1998).

[14] T. Lange, Efficient Arithmetic on Hyperelliptic Koblitz Curves, *Preprint*, Universität Gesamthochschule Essen (2001).

[15] J. W. Lee, Speeding Up the Arithmetic on the Jacobians of Hyperelliptic Curves, Preprint.

[16] W. Meier, O. Staffelbach, Efficient Multiplication on Certain Nonsupersingular Elliptic Curves, in: *Advances in Cryptology - Crypto '92*, Lecture Notes in Computer Science **740**, (Springer 1993), 333-344.

[17] V. Müller, Fast Multiplication on Elliptic Curves over Small Fields of Characteristic Two, *Journal of Cryptology* **11** (1998), 219-234.

[18] M. Nöcker, Data structures for parallel exponentiation, Ph.D. Thesis, Universität Paderborn (2001).

[19] N. P. Smart, Elliptic Curve Cryptosystems over Small Fields of Odd Characteristic, *Journal of Cryptology* **12** (1999), 141-151.

[20] J. Solinas, An Improved Algorithm for Arithmetic on a Family of Elliptic Curves, in: *Advances in Cryptology - Crypto '97*, Lecture Notes in Computer Science **1294**, (Springer 1997), 375-371.

[21] J. Solinas, Efficient arithmetic on Koblitz curves, *Journal of Designs, Codes and Cryptography* **19** (2000), 195-249.