On the Performance of Hyperelliptic Cryptosystems

Nigel P. Smart

Hewlett-Packard Laboratories, Filton Road, Stoke Gifford, Bristol, BS12 6QZ, U.K.
nsma@hplb.hpl.hp.com

Abstract. In this paper we discuss various aspects of cryptosystems based on hyperelliptic curves. In particular we cover the implementation of the group law on such curves and how to generate suitable curves for use in cryptography. This paper presents a practical comparison between the performance of elliptic curve based digital signature schemes and schemes based on hyperelliptic curves. We conclude that, at present, hyperelliptic curves offer no performance advantage over elliptic curves.

Elliptic curve cryptosystems are now being deployed in the real world and there has been much work in recent years on their implementation. A natural generalization of such schemes was given by Koblitz [12], who described how the group law on a Jacobian of a hyperelliptic curve can be used to define a cryptographic system. Almost all of the standard discrete logarithm based protocols such as DSA and ElGamal have elliptic and hyperelliptic variants. This is because such protocols only require the presence of a finite abelian group, with a large prime order subgroup, within which the basic group operation is easy whilst the associated discrete logarithm problem is hard. We shall not discuss these protocols in this paper since everything that can be said for elliptic curve based protocols can usually be said for hyperelliptic curve based protocols. Instead we shall concentrate more on the underlying group: In particular how one performs the group operation and how one produces groups of the required type.

The Jacobian of a genus g hyperelliptic curve will have roughly q^g points on it, where q denotes the number of elements in the field of definition of the Jacobian. By choosing hyperelliptic curves of genus greater than one we can achieve the same order of magnitude of the group order with a smaller value for q when compared with elliptic curve based systems which have $g = 1$. This has led some people to suggest that hyperelliptic curves may offer some advantages over elliptic curves in some special situations. For example if we wanted to only perform arithmetic using single words on a 32-bit computer we could choose $g = 5$ or 6 to obtain group orders of around 160 to 192 bits.

One has to be a little careful as to how large one makes g, since for large genus there is a sub-exponential method to solve the discrete logarithm problem [1]. However this does not appear to affect the security of curves of genus less than 10 over field sizes of around 32 bits.

J. Stern (Ed.): EUROCRYPT'99, LNCS 1592, pp. 165–175, 1999.

In this paper we give an overview of the group law on a curve of genus g in arbitrary characteristic. We shall give a more efficient reduction method than the standard method of Cantor [3]. This is an immediate extension of the method of Tenner reduction from [19]. We shall then describe various techniques for generating hyperelliptic curves for use in cryptography.

Finally we report on an actual implementation of a hyperelliptic digital signature algorithm. We will conclude that hyperelliptic systems, with current algorithms, are more efficient in characteristic two but appear to offer no practical advantage over elliptic curve systems.

1 Arithmetic

In this section we summarize the details and leave the reader to consult [12] for a fuller explanation. A hyperelliptic curve, C, of genus g will be given in the form

$$C : Y^2 + H(X)Y = F(X)$$

where $F(X)$ is a monic polynomial of degree $2g + 1$ and $H(X)$ is a polynomial of degree at most g. Both $H(X)$ and $F(X)$ have coefficients in $\mathbb{F}_q$. Such a curve is non-singular if for no point on $C(\overline{\mathbb{F}}_q)$ does there exist a point for which the two partial derivatives,

$$2Y + H(X) \text{ and } H'(X)Y - F'(X),$$

simultaneously vanish. We shall always assume that the curve C is non-singular.

In odd characteristic fields we will always assume that $H(X) = 0$, whilst in even characteristic fields we will assume that $H(X) = 1$, for reasons which will become clear later. Notice that if $H(X) = 1$ then in characteristic two any choice for the polynomial $F(X)$ will give rise to a non-singular curve.

The above representation gives rise to a so called 'imaginary' quadratic function field. It is given this name since there are no units of infinite order and the arithmetic in the Jacobian closely mirrors the arithmetic one uses for the class group of an imaginary quadratic number field.

We can also define a hyperelliptic curve of genus g to be given by an equation, like that above but, with $\deg F = 2g + 2$. This gives rise to a 'real' quadratic function field. It is easy to see that, unlike the number field situation, an imaginary quadratic function field can be viewed as a real quadratic function field after making a change of variables. However, just as in the case of the class group of real quadratic number fields, the arithmetic in the Jacobians of real quadratic hyperelliptic curves is more involved and requires the use of 'infrastructure'. The reader should consult [18] for an explanation of the algorithms required and [19] for a complexity analysis of the two situations. For the rest of this article we will concentrate on the imaginary quadratic representation, which is more suited to efficient implementations in practice.

Following Cantor and Koblitz, an element of the Jacobian of C will be given by two polynomials $a, b \in \mathbb{F}_q[x]$ which satisfy

i) $\deg b < \deg a \leq g$.
ii) b is a solution of the equation $b^2 + Hb - F \pmod{a}$.

Addition in the Jacobian is accomplished by two procedures: Composition and Reduction. Given (a_1, b_1) and (a_2, b_2) the composition of these two elements in the group of divisors is given by (a_3, b_3) using the following algorithm due to Cantor and Koblitz:

Composition

---

1. Perform two extended gcd computations to compute
$$d = \gcd(a_1, a_2, b_1 + b_2 + H) = s_1 a_1 + s_2 a_2 + s_3(b_1 + b_2 + H).$$
2. Set $a_3 = a_1 a_2 / d^2$ and
3. $b_3 = (s_1 a_1 b_2 + s_2 a_2 b_1 + s_3(b_1 b_2 + F))/d \pmod{a_3}$.

Note that a_3 will have degree at most $2g$ and hence (a_3, b_3) will most probably need to be reduced. We shall return to this later. Notice, however, that for cryptography the most important composition step is doubling, where $a_1 = a_2$ and $b_1 = b_2$. This is because in discrete logarithm based systems we wish to perform a multiplication operation on the Jacobian. Using window techniques this involves mainly the doubling of elements rather than a general composition. Hence it is important that doubling an element can be accomplished efficiently.

With our above choice of curves in odd and even characteristic we find:

Doubling in Odd Characteristic Fields

Since we have chosen $H(X) = 0$ the doubling operation simplifies to: Put $d = \gcd(a_1, 2b_1) = s_1 a_1 + s_3(2b_1)$ then $a_3 = (a_1/d)^2$ and $b_3 = (2s_1 a_1 b_1 + s_3(b_1^2 + F))/d$.

Doubling in Even Characteristic Fields

Now since we have $H(X) = 1$ the doubling operation simplifies to: Put $a_3 = a_1^2$ and $b_3 = b_1^2 + F \pmod{a_3}$. This is much simpler than the odd characteristic step and contributes to much faster times for the verifying of messages using curves over even characteristic fields, see below for details.

We shall now describe the reduction step, which given the result (a_3, b_3) of a composition will return an element, (a, b), of the Jacobian with $\deg a \leq g$. The element (a_3, b_3) represents an element in the group of divisors. Since we are in an imaginary quadratic situation every divisor class (and so every element in the Jacobian) can be represented by a unique, so called reduced, divisor. The reduction step takes the divisor represented by (a_3, b_3) and returns the unique reduced divisor (a, b) in the same divisor class as (a_3, b_3). As mentioned above we use a variant of Tenner reduction which is more efficient than the method given by Cantor and Koblitz.

Reduction

1. $a = (b_3^2 + b_3 H - F)/a_3$.
2. $(u, b) = \texttt{quo/rem}(-b_3 - H, a)$.
3. `While` $\deg a > g$
4. $\quad a^* = a_3 + u(b_3 - b)$.
5. $\quad a_3 = a, a = a^*, b_3 = b$.
6. $\quad (u, b) = \texttt{quo/rem}(-b_3 - H, a)$.

This is exactly the same as the standard method except for Step 4. In this step we have replaced the division $a^* = (b^2 + Hb - F)/a$ with simpler operations, on noticing that u in general will have small degree whilst $\deg a$ in Step 4 could be at most $2g - 2$. To see that Step 4 is equivalent to the standard method we notice that $u = (-b_3 - H - b)/a$ and so

$$\begin{aligned} a^* &= a_3 + (b_3 - b)\left(\frac{-b_3 - H - b}{a}\right) \\ &= (b^2 + Hb - F)/a. \end{aligned}$$

In [6] the extended Euclidean algorithm is analyzed in the context of hyperelliptic cryptosystems. As we have already pointed out for even characteristic fields for the most important operation, point doubling, no extended Euclidean algorithm is required. Most of the effort in performing a sign or verify operation is in the reduction step. Hence analyzing the reduction step is far more important, luckily this has already been done in [19], where it is shown that the above reduction step takes $12g^2 + O(g)$ field operations, in [18] the standard method is stated to take $3g^3 + O(g^2)$ field operations. However, a complexity analysis can often be inapropriate since complexity only deals with the assymptotics of an algorithm. In real life the relative performance of algorithms in small ranges can depend on factors such as cache size and processor type.

2 Curve Generation

There are many ways, in theory, that one could proceed if one wanted to produce curves suitable for use in cryptography. Many of the methods are analogues of those used in the elliptic curve case. The order of $|J(\mathbb{F}_q)|$ can be computed in polynomial time using methods due to Adleman, Huang and Pila, see [2] and [20], which are themselves generalizations of the method of Schoof [25] used in the elliptic case. There is no implementation of this method for genus greater than one at the present time. This is probably because the algorithm, although easy to understand, appears very hard to implement. Another reason is that there is no known analogue of the improvements made by Atkins and Elkies to the

original Schoof algorithm. Hence only the 'naive' Schoof algorithm is available in genus greater than one. Such an algorithm appears hopeless as a method, since the 'naive' Schoof algorithm is far too inefficient even for elliptic curves.

The fact that it seems unlikely that anyone can compute the order of $J(\mathbb{F}_q)$ for a general curve of genus 5 or 6 could lead one to propose that one should not worry. For example, if I do not believe that someone can compute the order of $J(\mathbb{F}_q)$ then I do not need to worry about many of the attacks on such systems, since most attacks such as Pohlig-Hellman require knowledge of the group order. This of course also means that our protocols need to be changed so that they do not require knowledge of the group order. Although this is a possible approach, it is to be rejected as it is assumes that someone will not make a known polynomial time algorithm run efficiently. Our security is therefore not built on the difficulty of some underlying mathematical problem but on the difficulty of programming a known algorithm efficiently.

Just as for elliptic curves one can compute hyperelliptic curves using the theory of Complex Multiplication (CM). This has been worked out in detail for the case of $g = 2$ in [30] and uses the class groups of complex quadratic extensions of real quadratic number fields, which are the quartic CM fields. Clearly the class numbers of any such field used should be small, and hence the curves which are produced will in some sense be 'special'. In the CM method for hyperelliptic curves multi-variable analogues of the Hilbert polynomial are constructed, the roots of which modulo p gives the j-invariants of the curve. The curve is then recovered from its j-invariants.

This method is only currently effective in genus two since the j-invariants of a hyperelliptic curve have only been worked out for genus less than three. The invariants used are the Igusa Invariants [11] which are linked to the classical 19th Century invariants of quintic and sextic polynomials. After the demise of classical invariant theory at the end of the 19th Century the drive to compute invariants of the higher order quantics, as they were then called, died out. Even today with the advent of computer algebra systems this seems a daunting task. One way around this problem, which still uses CM, is to use reductions of hyperelliptic curves defined over $\mathbb{Q}$ which have global complex multiplication, see [4]. However, here one is restricting to an even more special type of hyperelliptic curve than the general CM method above.

Another technique is to use the theory of the modular curves, $X_0(N)$, see [8] and [15]. Such curves are well studied and much is known about them. This enables us to compute the orders of the Jacobians of such curves in a much easier way than other general curves. However, paranoid readers should beware since they are well understood curves with special properties they may be susceptible to some new attack which makes use of the fact that they are modular.

Koblitz, in [13], suggests using curves of the form

$$v^2 + v = u^n,$$

over some finite prime field $\mathbb{F}_p$. Given such curves he then gives a procedure to determine the group order by evaluating a Jacobi sum of a certain character. We refer the reader to Koblitz's book for details. However once again we are restricting to a very special type of curve which may be susceptible to some, as yet unknown, attack.

In characteristic two one can use curves defined over subfields [12] just as one can do for elliptic curves. For example a simple search found the curves in Table 1, which all have subgroups of their Jacobians of 'large' prime order; We could also use such a technique to generate curves over $\mathbb{F}_p$, where p is a small odd prime and look at the Jacobian over $\mathbb{F}_{p^n}$.

Table 1. Curves of the form $Y^2 + Y = F(X)$

$\mathbb{F}_q$	$F(X)$	$\log_2 p$ where $p \mid \#J(\mathbb{F}_q)$
$\mathbb{F}_{2^{31}}$	$X^{11} + X^5 + 1$	150
$\mathbb{F}_{2^{29}}$	$X^{13} + X^{11} + X^3 + X$	157
$\mathbb{F}_{2^{29}}$	$X^{13} + X^{11} + X^7 + X + 1$	153
$\mathbb{F}_{2^{29}}$	$X^{13} + X^{11} + X^7 + X^3 + 1$	169
$\mathbb{F}_{2^{29}}$	$X^{13} + X^{11} + X^9 + X^5 + 1$	170
$\mathbb{F}_{2^{29}}$	$X^{13} + X^{11} + X^9 + X^7 + X^3 + X + 1$	152
$\mathbb{F}_{2^{31}}$	$X^{13} + X^{11} + X^7 + X^3 + X$	162
$\mathbb{F}_{2^{31}}$	$X^{13} + X^{11} + X^9 + X + 1$	154
$\mathbb{F}_{2^{31}}$	$X^{13} + X^{11} + X^9 + X^5$	158
$\mathbb{F}_{2^{31}}$	$X^{13} + X^{11} + X^9 + X^7$	178
$\mathbb{F}_{2^{31}}$	$X^{13} + X^{11} + X^9 + X^7 + X^3 + X + 1$	181
$\mathbb{F}_{2^{31}}$	$X^{15} + X$	207
$\mathbb{F}_{2^{31}}$	$X^{15} + X^5 + X^3 + X$	200

Apart from the, currently unimplemented, method of Schoof, Pila et al the above methods do not seem very pleasing. It is a good general principle never to choose a curve with 'special structure', and all of the above schemes use 'special' properties of the curves to make the group order computation easier.

To see why one should avoid special curves one only has to look at the history of elliptic curve cryptography. In the past various authors proposed using supersingular or anomalous curves as they offered some advantages over other more general curves. However, both types of curves are now known to be weak, see [14], [24], [26] and [27]. Hence it is probably worth adopting the principle of always avoiding special curves of any shape or form. In the current authors opinion this is the major open problem with using hyperelliptic curves for cryptographic purposes: How to choose a suitable curve efficiently ?

3 The Discrete Logarithm Problem in Hyperelliptic Jacobians

The security of hyperelliptic cryptosystems is based upon the difficulty of solving the discrete logarithm problem in the Jacobian of the curve. We summarize the main characteristics of the possible attacks on the hyperelliptic discrete logarithm problem below. The reader should note that in all but one case they closely mirror analogues for the elliptic curve discrete logarithm problem.

Apart from the generic discrete logarithm algorithms such as the baby-step / giant-step and the rho/kangaroo method there are three known methods which are specific to hyperelliptic curves. Two of these give rise to two weak classes of hyperelliptic curve cryptosystems:

1. Curves of order n over $\mathbb{F}_q$ such that $q^l \equiv 1 \pmod{n}$ for some small value of n. This is due to a generalization of the method of Menezes et al [14] for supersingular elliptic curves due to Frey and Rück [9].
2. Anomalous curves over $\mathbb{F}_p$ and in general curves which have a large subgroup of order p in a field of characteristic p. This attack uses a generalization due to Rück [21] of the anomalous curve attack for elliptic curves due to Semaev, Satoh, Araki and Smart, see [24], [26] and [27].

However, such cases are easy to check for and only eliminate a small fraction of all possible curves.

For hyperelliptic curves the most interesting case, from a theoretical standpoint, is when the genus is large in comparison to the size of the field of definition of the Jacobian. In this case there are conjectured subexponential methods. The first of these was due to Adleman, De Marrais and Huang which is based on the number field sieve factoring method.

Paulus [17] and Flassenberg and Paulus [7] have implemented such a method for solving discrete logarithms in Jacobians of hyperelliptic curves. Flassenberg and Paulus did not, however, use the method of Adleman, De Marrais and Huang directly. Instead they made use of the fact that our hyperelliptic curves correspond to real quadratic function field extensions. Using the analogy between quadratic function fields and quadratic number fields, Flassenberg and Paulus adapt the class group method of Hafner and McCurley [10] (see also [5]). Then combining this with a sieving method they obtain a working method which can be applied to hyperelliptic curves of relatively small genus. It should be pointed out that although Flassenberg and Paulus do not actually solve discrete logarithm problems their methods are such that they can be easily extended so that they do.

Flassenberg and Paulus compared their algorithm to the baby-step / giant-step approach. Over finite prime fields, $\mathbb{F}_p$, their implementation of the Hafner-McCurley method beat the baby-step / giant-step method, as soon as $3g > \log p$. However, this is only given a very small sample size. But it would appear, for theoretical reasons as well, to be a good rule of thumb to avoid curves for which $2g > \log q$. Hence if $q \approx \mathbb{F}_{2^{31}}$ then we should avoid curves whose genus is larger than eleven.

4 Implementation

In [22] the number of bit operations for implementing a hyperelliptic cryptosystem is studied and compared with both ECC and RSA systems which offer roughly the same level of security. It is concluded that hyperelliptic cryptosystem could be efficient enough in practice to use in real life situations. Following on from this work in [23] an implementation of such a system is described. However this implementation makes no use of Tenner reduction and generally uses field sizes which require more than a single word to represent each field element.

We decided to implement the group law in the Jacobian for curves of arbitrary genus over $\mathbb{F}_{2^n}$ and $\mathbb{F}_p$, where p is a prime. We decided to choose values of p and n such that p and 2^n are less than 2^{32}. This choice was to make sure that our basic arithmetic could all be fitted into single words on our computer. Such curves and fields have attracted some interest in the community in recent years since they may offer some implementation advantages. In even characteristic we used a trinomial basis while in odd characteristic we used a small in-lined machine code subroutine to perform the modular multiplication. Field inversion in both cases was carried out using a modification of the binary method.

The general multiplication algorithm on the Jacobian for curves defined over odd characteristic fields ended up being around twice as slow as that for even characteristic fields, of an equivalent size, in genus two. In genus five the odd characteristic fields were nearly three times slower. This fact led us to only implement a full digital signature scheme in characteristic two.

For the signing operation the multiplication performed is on the fixed group generator. Hence this can be efficiently accomplished using a precomputed table of powers of the generator. The verification step requires two multiplications, one of the generator and one of a general point. Hence for verification we cannot use precomputed tables and the difficulty of doubling an element will dominate the computation. For the general multiplication, used in the verification step, we used a signed window method, since negation in the Jacobian of a hyperelliptic curve comes virtually for free.

Our timings, in milliseconds, for a hyperelliptic variant of the DSA method (HCDSA) are given in Table 2. These timings were obtained on a Pentium Pro 334MHz, running Windows NT, using the Microsoft Visual C++ compiler. We also give an estimate of the timings for an elliptic curve (ECDSA) system with approximately the same group order.

The elliptic curve implementation made no use of special field representations, such as using the subfield structure. The even characteristic field representation for the elliptic curve system was a standard polynomial basis. The odd characteristic field (of size approximately 2^{161}) used for the elliptic curve system used a Montgomery representation.

So we see that even though the finite field elements fit into a single word the extra cost of the polynomial arithmetic needed for operations in the Jacobian makes the time needed to perform the complete set of hyperelliptic curve operations over four times slower than in the elliptic curve case. If more efficient

Table 2. HCDSA and ECDSA Timings in Milliseconds

Curve	Field	Sign	Verify
HCDSA $g = 5$	$\mathbb{F}_{2^{31}}$	18	71
HCDSA $g = 6$	$\mathbb{F}_{2^{31}}$	26	98
HCDSA $g = 7$	$\mathbb{F}_{2^{31}}$	40	156
ECDSA	$\mathbb{F}_{2^{161}}$	4	19
ECDSA	$\mathbb{F}_p$	3	17

elliptic curve techniques were used then the relative performance of the HCDSA algorithm would degrade even more.

Given the relative difficulty of finding hyperelliptic curves for use in cryptography which do not possess some addition structure and the relatively poor performance of the HCDSA algorithm when compared to ECDSA there seems no benefit in using hyperelliptic curves.

Of course further work could result in significant speed improvements for hyperelliptic systems. For example at present there appears to be no notion akin to the projective representation in elliptic curves. Another possible avenue for improvement is to use Frobenius expansions. Not as much work has been carried out in the hyperelliptic case to the study of Frobenius expansions compared to the elliptic curve case. These are useful for curves defined over small subfields, such as those used above. The only cases having been considered in the hyperelliptic case are in [12]. However, for elliptic curves Frobenius expansions techniques can be made very fast in all characteristics, see [16], [28] and [29].

References

1. L. Adleman, J. De Marrais, and M.-D. Huang. A subexponential algorithm for discrete logarithms over the rational subgroup of the Jacobians of large genus hyperelliptic curves over finite fields. In *ANTS-1 : Algorithmic Number Theory*, Editors L.M. Adleman and M-D. Huang, Springer-Verlag, LNCS 877, pp 28–40, 1994.
2. L. Adleman and M.-D. Huang. Counting rational points on curves and abelian varieties over finite fields. In *ANTS-2 : Algorithmic Number Theory*, Editor H. Cohen, Springer-Verlag, LNCS 1122, pp 1–16, 1996.
3. D.G. Cantor. Computing in the Jacobian of a hyper-elliptic curve. *Math. Comp.*, **48**, 95–101, 1987.
4. J. Chao, N. Matsuda and S. Tsujii. Efficient construction of secure hyperelliptic discrete logarithms. In *Information and Communications Security*, Editors Y. Han, T. Okamoto and S. Quing, Springer-Verlag, LNCS 1334, pp 292–301, 1997.
5. H. Cohen. A Course In Computational Algebraic Number Theory. Springer-Verlag, GTM 138, 1993.
6. A. Enge. The extended Euclidean algorithm on polynomials, and the efficiency of hyperelliptic cryptosystems. Preprint, 1998.
7. R. Flassenberg and S. Paulus. Sieving in function fields. *Preprint*, 1997.

8. G. Frey and M. Müller. Arithmetic of modular curves and its applications. Preprint, 1998.
9. G. Frey and H.-G. Rück. A remark concerning m-divisibility and the discrete logarithm problem in the divisor class group of curves. *Math. Comp.*, **62**, 865–874, 1994.
10. J.L. Hafner and K.S. McCurley. A rigorous subexponential algorithm for computation of class groups. *J. AMS*, **2**, 837–850, 1989.
11. J.I. Igusa. Arithmetic variety of moduli for genus two. *Ann. Math.*, **72**, 612–649, 1960.
12. N. Koblitz. Hyperelliptic cryptosystems. *J. of Crypto.*, **1**, 139–150, 1989.
13. N. Koblitz, Algebraic aspects of cryptography. *Vol. 3, Algorithms and Computation in Mathematics*, Springer-Verlag, Berlin, 1998.
14. A. Menezes, T. Okamoto and S. Vanstone. Reducing elliptic curve logarithms to a finite field. *IEEE Trans. on Inform. Theory*, **39**, 1639–1646, 1993.
15. M. Müller. Algorithmische Konstruktion hyperelliptischer Kurven mit kryptographischer Relevanz und einem Endomorphismenring echt grösser als $\mathbb{ZZ}$. *Phd Thesis*, Universität Essen, 1996.
16. V. Müller. Fast multiplication on elliptic curves over small fields of characteristic two. *J. Crypto.*, **11**, 219–234, 1998.
17. S. Paulus. An algorithm of sub-exponential type computing the class group of quadratic orders over principal ideal domains. In *ANTS-2 : Algorithmic Number Theory*. Editor H. Cohen, Springer-Verlag, LNCS 1122, pp 243–257, 1996.
18. S. Paulus and H.-G. Rück. Real and imaginary quadratic representation of hyperelliptic function fields. To appear *Math. Comp.*.
19. S. Paulus and A. Stein. Comparing real and imaginary arithmetics for divisor class groups of hyperelliptic curves. In *ANTS-3 : Algorithmic Number Theory*, Editor J. Buhler, Springer-Verlag, LNCS 1423, pp 576–591, 1998.
20. J. Pila. Frobenius maps of abelian varieties and finding roots of unity in finite fields. *Math. Comp.*, **55** , 745–763, 1996.
21. H.-G. Rück. On the discrete logarithm problem in the divisor class group of curves. Preprint 1997.
22. Y. Sakai, K. Sakurai and H. Ishizuka. Secure hyperelliptic cryptosystems and their performance. In *Public Key Cryptography*, Editors H. Imai and Y. Zheng, Springer-Verlag, LNCS 1431, pp 164–181, 1998.
23. Y. Sakai and K. Sakurai. Design of hyperelliptic cryptosystems in small characteristic and a software implementation over $\mathbb{F}_{2^n}$. In *Advances in Cryptology, ASIACRYPT 98*, Editors K. Ohta and D. Pei, Springer-Verlag, LNCS 1514, pp 80–94, 1998.
24. T. Satoh and K. Araki. Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves. *Comm. Math. Univ. Sancti Pauli*, **47**, 81–92, 1998.
25. R. Schoof. Elliptic curves over finite fields and the computation of square roots mod p. *Math. Comp.*, **44**, 483–494, 1985.
26. I.A. Semaev. Evaluation of discrete logarithms on some elliptic curves. *Math. Comp.*, **67**, 353–356, 1998.
27. N.P. Smart. The discrete logarithm problem on elliptic curves of trace one. To appear *J. Crypto.*, 1999.
28. N.P. Smart. Elliptic curves over small fields of odd characteristic. To appear *J. Crypto.*, 1999.

29. J.A. Solinas. An improved algorithm for arithmetic on a family of elliptic curves. In *Advances in Cryptology, CRYPTO 97*, Editor B. Kaliski, Springer Verlag, LNCS 1294, pp 357-371, 1997.
30. A-M. Spallek. Kurven vom Geschlecht 2 und ihre Anwendung in Public-Key-Kryptosytemen. *Phd Thesis*, Universität Essen, 1994.