High Performance Arithmetic for Hyperelliptic Curve Cryptosystems of Genus Two

Jan Pelzl, Thomas Wollinger, and Christof Paar

Communication Security Group (COSY)
Department of Electrical Engineering and Information Sciences
Ruhr-Universitaet Bochum, Germany
{pelzl, wollinger, cpaar}@crypto.rub.de

Abstract. Nowadays, there exists a manifold variety of cryptographic applications: from low level embedded crypto implementations up to high end cryptographic engines for servers. The latter require a flexible implementation of a variety of cryptographic primitives in order to be capable of communicating with several clients. On the other hand, on the client it only requires an implementation of one specific algorithm with fixed parameters such as a fixed field size or fixed curve parameters if using ECC/ HECC. In particular for embedded environments like PDAs or mobile communication devices, fixing these parameters can be crucial regarding speed and power consumption. In this contribution, we propose a highly efficient algorithm for a hyperelliptic curve cryptosystem of genus two, well suited for these constraint devices.

In recent years, a lot of effort was made to speed up arithmetic on genus-2 HEC. This work is based on the work of Lange [Lan02a,Lan03] and presents a major improvement of HECC arithmetic for curves defined over fields of characteristic two. We optimized the group doubling operation for certain types of genus-2 curves and we were able to reduce the number of required multiplications to a total of 9 multiplications. The saving in multiplications is 47% for the cost of one additional squaring. Thus, the efficiency of the whole cryptosystem was drastically increased.

Keywords: hyperelliptic curves, explicit formulae, efficient implementation, genus two

1 Introduction

Modern cryptographic implementations vary from highend server applications to applications on conventional PCs, Workstations, and applications on embedded devices such as PDAs and mobile communications devices. Every implementation is adapted to its requirements given by the application. If we consider, e.g., a scenario consisting of a central server controlling access from and to a network of several PDAs, each PDA will use a different cryptographic primitive (algorithm, curve, field polynomial etc.). To be capable of communicating with all PDAs, the cryptographic engine running on the server has to support a whole suite of cryptographic algorithms whereas each algorithm has to cope with different input parameters. In contrast, the implementations on constrained platforms (like the PDA) usually require only one cryptographic algorithm with a fixed set of input parameters. Hence, implementations with fixed parameters are attractive for embedded applications and those with flexible parameters for systems with fewer constraints such as servers.

During the last decade, asymmetric cryptosystems based on elliptic curves have become very popular, especially for embedded applications. Elliptic curve cryptosystems (ECC) benefit from shorter operand sizes when compared to RSA or DL based systems. This fact makes ECC particularly well suited for small processors and memory constrained environments. Since their introduction, ECC have been extensively studied by the research community and in industry.

Elliptic curves are a special case of hyperelliptic curves (HEC). In 1988, Koblitz suggested HEC for the use in cryptosystems [Kob88]. In contrast to the EC case, it has only been until recently that the idea to use HEC for cryptographic applications has been analyzed and implemented both in software [Kri97,SS98,SSI98,Eng99b,SS00] and in more hardware-oriented platforms such as FPGAs [Wol01,WP02,BCLW02]. Since the HEC operand size is only a fractional amount of the EC operand size, HECC is a cryptosystem of choice when targeting embedded environments.

In 1999, [Sma99] concluded that there seems to be little practical benefit in using HEC, because of the difficulty of finding hyperelliptic curves and their relatively poor performance when compared to EC. However, quite recently the efficiency of HEC group operation has been improved in such way, that HECC has become an attractive alternative to other cryptosystems like RSA or ECC [Har00,MCT01,MDM+02,Tak02,Lan02a,Lan03,PWGP03,PWP03].

In this contribution, we improved the arithmetic of doubling a divisor on genus-2 HEC. When using windowing methods for a divisor scalar multiplication, doubling becomes the crucial step for the performance of the entire cryptosystem. Thus, improving the arithmetic of doubling has a direct impact on the efficiency of the whole system.

The remainder of the paper is organized as follows: Section 2 summarizes contributions dealing with previous improvements of group operations for HECC. Section 3 gives a brief overview of the mathematical background related to HECC. Section 4 describes the derivation of the new explicit formulae for genus-2 curves. Finally, we present our results in Section 5 and conclude with a discussion of our results in Section 6.

2 Previous Work

In this section, we briefly summarize previous attempts to refine group operation on genus-2 curves. Since the proposal of HECC for the use in cryptography in 1988 [Kob88], it took several years until the rise of first improvements of its arithmetic. Nagao [Nag00] proposed several improvements of the polynomial arithmetic of Cantor's algorithm [Can87]. The same year, Harley came up with the first explicit formulae for Cantor's algorithm for genus-2 HEC [Har00] resulting in a drastic speed up of the cryptosystem. After Harley's proposal, several contributions containing improvements of the explicit formulae followed. The authors in [MCT01,MDM+02,Tak02,Lan02a,Lan03,Lan02b,Lan02c] targeted genus-2 curves whereas [KGM+02,Pel02,PWGP03,PWP03] deal with the derivation and improvement of explicit formuale for HEC of genus 3 or 4. For more details on previous improvements made to the explicit formulae the interested reader is referred to [Pel02,PWP03,PWGP03].

The publication by Lange [Lan02a] is the first to address formulae for curves of arbitrary characteristic. In particular curves of characteristic two are important from an implementational point of view since operands in the underlying field can be represented as binary vectors. The formulae for a group addition and a group doubling can be found in Appendix A in Table 2 and Table 3, respectively.

3 Mathematical Background

In this section we present an elementary introduction to some of the theory of hyperelliptic curves over finite fields of arbitrary characteristic, restricting attention to material that is relevant for this work. For more details the reader is referred to [Kob89,Kob98].

3.1 Hyperelliptic Curves

Let $\mathbb{F}$ be a finite field, and let $\overline{\mathbb{F}}$ be the algebraic closure of $\mathbb{F}$. A hyperelliptic curve C of genus $g \geq 1$ over $\mathbb{F}$ is the set of solutions $(x, y) \in \mathbb{F} \times \mathbb{F}$ to the equation

$$C : y^2 + h(x)y = f(x)$$

The polynomial $h(x) \in \mathbb{F}[x]$ is of degree at most g and $f(x) \in \mathbb{F}[x]$ is a monic polynomial of degree $2g + 1$. For odd characteristic it suffices to let $h(x) = 0$ and to have $f(x)$ squarefree. Such a curve is said to be non-singular if there are no pairs $(x, y) \in \overline{\mathbb{F}} \times \overline{\mathbb{F}}$ which simultaneously satisfy the equation of the curve C and the partial differential equations $2v + h(x) = 0$ and $h'(x)v - f'(x) = 0$.

If we want to define the Jacobian over $\mathbb{F}$, denoted by $\mathbb{J}_C(\mathbb{F})$, we say that a divisor $D = \sum m_i P_i$ is defined over $\mathbb{F}$ if $D^\sigma = \sum m_i P_i^\sigma$ is equal to D for all automorphisms σ of $\overline{\mathbb{F}}$ over $\mathbb{F}$ [MWZ96].

Each element of the Jacobian can be represented uniquely by a reduced divisor [Ful69,Can87]. This divisor can be represented as a pair of polynomials $u(x)$ and $v(x)$ with $\deg v(x) < \deg u(x) \leq g$, with $u(x)$ dividing $y^2 + h(x)y - f(x)$ and where the coefficients of $u(x)$ and $v(x)$ are elements of $\mathbb{F}$ [Mum84, page 3.17]. In the remainder of this paper, a divisor D represented by polynomials will be denoted by $div(u, v)$. Cantor's algorithm describes the group addition of two divisors on $\mathbb{J}_C(\mathbb{F})$ [Can87]. In 2000, Harley proposed the first explicit formulae for a group addition and a group doubling of divisors on $\mathbb{J}_C(\mathbb{F})$ [Har00].

3.2 Group Operations on the Jacobian

This section will provide a brief description of the algorithms used for adding and doubling divisors on $\mathbb{J}_C(\mathbb{F})$. We will solely concentrate on Harley's algorithm which is the starting point for all further improvements done on genus-2 curve arithmetic.

In [GH00], Gaudry and Harley could reduce the number of operations by distinguishing between possible cases according to the properties of the input divisors. They described an efficient algorithm (using Karatsuba multiplication, CRT, and Newton Iteration) to reduce the overall complexity of the group operations. All further contributions dealing with explicit formulae concentrated solely on the (frequent) case where the two input divisors are coprime and of weight two. For HEC of genus 2 over $\mathbb{F}_{2^n}$, this case occurs with overwhelming probability of $P \approx 1 - 2^{-80}$ and, thus, is within main concern of all implementations.

Algorithm 1 combines all steps of the most frequent case of doubling a divisor for arbitrary characteristic. Algorithm 2 depicts all steps when adding two divisors.

Algorithm 1 Frequent Case for Group Doubling (g=2)

Require: $D_1 = \mathrm{div}(u_1, v_1)$
Ensure: $D_2 = \mathrm{div}(u_2, v_2) = 2D_1$
1: $k = \frac{v_1^2 - v_1 h - f}{u_1}$ (exact division)
2: $s \equiv \frac{k}{h + 2v_1} \bmod u_1$
3: $u' = s^2 + \frac{k - s(h + 2v_1)}{u_1}$ (exact division)
4: $u_2 = u'$ made monic
5: $v_2 \equiv -(h + su_1 + v_1) \bmod u'$

Algorithm 2 Frequent Case for Group Addition (g=2)

Require: $D_1 = \mathrm{div}(u_1, v_1)$, $D_2 = \mathrm{div}(u_2, v_2)$
Ensure: $D_3 = \mathrm{div}(u_3, v_3) = D_1 + D_2$
1: $k = \frac{f - v_1 h - v_1^2}{u_1}$ (exact division)
2: $s \equiv \frac{v_2 - v_1}{u_1} \bmod u_2$
3: $z = su_1$
4: $u' = \frac{k - s(z + h + 2v_1)}{u_2}$ (exact division)
5: $u_3 = u'$ made monic
6: $v_3 \equiv -(h + z + v_1) \bmod u_3$

3.3 Security of HECC

Most cryptographic applications based on EC or HEC require a group order of size of at least $\approx 2^{160}$. Thus, for HECC over $\mathbb{F}_q$ we will need at least $g \cdot \log_2 q \approx 2^{160}$, where g is the genus of the curve. In particular, for a curve of genus two, we will need a field $\mathbb{F}_q$ with $|\mathbb{F}_q| \approx 2^{80}$, i.e., 80-bit long operands. It is well known that the best algorithm to compute the discrete logarithm in generic groups such as the Jacobian of a HEC is Pollard's rho method or one of its parallel variants [Pol78,vOW99]. They solve the DLP with complexity $O(\sqrt{n})$ in generic groups of order n. In [FR94,Rüc99], attacks against special cases of HECC were discovered with complexity smaller than $O(\sqrt{n})$. An algorithm to compute the DL in subexponential time for sufficiently large genera and variants of this algorithm were published in [ADH94,FS97,Eng99a,Gau00,EG02]. The complexity of these algorithms is only better than the Pollard's rho method for $g \geq 3$. In [Gau00] it is shown that index-calculus algorithms in the Jacobian of HEC have a higher complexity than the Pollard rho method for curves of genus greater than 4. Recent results by Thériault [Thé03] show progress in attacks against HEC of genus 3 and higher. In the case of genus-3 curves, the group size should be increased by $\approx 5\%$ according to [Thé03].

4 New Improvements of the Arithmetic

Quite recently, the research community put a lot of effort into increasing the efficiency of HEC group operations [MCT01,MDM$^+$02,Tak02,Lan02a]. The most efficient formulae known for group operations on genus two HEC over fields of even characteristic are summarized by Lange [Lan02a]. In this section we briefly outline our improvements on the formulae presented in [Lan02a]. All formulae are based on the ideas of Gaudry and Harley [GH00], who introduced the first explicit formulae with which the group operations were computed using the original algorithm presented by Cantor [Can87]. They noticed that one can reduce the number of operations required to add/double divisors by distinguishing between possible cases according to the properties of the input divisors. This technique is combined with the use of the Karatsuba multiplication algorithm [KO63] and the Chinese remainder theorem to further reduce the complexity of the overall group operations. All following contributions including this paper improved the algorithm proposed in [Har00].

With this work, we further optimized the formulae for doubling a divisor for fields $\mathbb{F}_{2^n}$. Table 2 and Table 3 present the explicit formulae for a group addition and a group doubling [Lan03] and Table 1 presents the optimized explicit formulae for doubling a divisor as derived in this paper.

The starting point for the improvements are the formulae displayed in Table 3. According to [Lan03], for even characteristic and $\deg[h(x)] = 1$, we can achieve $f_2 = f_3 = f_4 = 0$. With the substitution $x \rightarrow x - f_4/5$ we obtain a curve where $f_4 = 0$. $y \rightarrow y + h_1 f_3 x^2$ provides $f_3 = 0$. In addition, if we can find a b such that $f_3 h_0 + b^2 h_1 + b h_1^3 = f_2 h_1$ has a solution, we can achieve $f_2 = 0$ by substituting $y \rightarrow y + h_1 f_3 x^2 + bx$.

For our improvements we assume curves of the form $y^2 + xy = x^5 + f_1x + f_0$ where $f_0, f_1 \in \mathbb{F}_{2^n}$. The following modifications are applied to the steps of Table 3 to reduce the required number of multiplications from 17 to 9 as diplayed in Table 1:

- Steps 1-3 stay the same: the resultant turns out to be $r = u_0$ and the inverse is $inv = x + u_1$; solely the computation of $k = k_1x + k_0$ costs $1M$.
- In the calculation of Step 4, $s' = s'_1x+s'_0$ costs only two multiplications, since the terms $k_1inv_1 = k_1 = w_1$ and $w_1u_1 = k_1$ are already computed in Step 3. Furthermore, $s'_1 = inv_0k_1+inv_1k_0-u_1w_1$ reduces to $s'_1 = u_1k_1+k_0-u_1k_1 = k_0$. Instead of computing $s'_0 = k_0inv_0 - u_0k_1inv_1 = k_0u_1 - u_0k_1$ we compute $s'_0 = (u_0 + u_1)(k_0 + k_1) - u_1k_1 - u_0k_0 = (u_0 + u_1)(k_0 + k_1) - t_1 - t_2$. Since t_1 is precomputed in Step 3, the calculation of $(u_0 + u_1)(k_0 + k_1)$ and t_2 has to be carried out $(2M)$. Note that t_2 is required in Step 5, where $1/(rs'_1) = 1/(k_0u_0) = 1/t_2$.
- In Step 5, we do not normalize the s-polynomial as suggested by Takahashi [Tak02] and compute $s_1 = s'_1w_2$. Instead of computing s_0, we set $t_6 = s_0u_1 = u_1^3+u_1^2s_1 = k_1(u_1+s_1)$, which is required in Step 6. Thus, we do not need the former value w_2 from Table 3 and can reduce the cost by two multiplications at the cost of one additional squaring.
- With the previously calculated value t_6, the computation of $z = su$ comes for free, reducing the cost by $2M$ compared to Table 3. These savings are possible due to s being *not* monic.
- Contrary to Step 7 in Table 3, we have to normalize the u' polynomial by multiplicating with $1/s_1^2 = w_4$ in Step 7. This results in $u'_1 = w_4$ and $u'_0 = w_4s_0^2 + k_1 + w_3$. $w_4s_0^2$ can be computed as $w_4s_0^2 = w_4(u_1^2 + s_1^2)u_1^2 = w_4u_1^4+w_4s_1^2u_1^2 = w_4u_1^4+u_1^2 = w_4k_1^2+k_1$ (due to the fact that $s_0 = k_1+u_1s_1 \Leftrightarrow s_0/u_1 = k_1/u_1 + s_1 = u_1 + s_1$). Thus, Step 7 consumes only $1M + 1S$.
- With s not monic, v' simply is obtained by $v' = -(h + z + v) \bmod u'$. Using Karatsuba style reduction and some values computed in previous steps, Step 8 can be carried out with two multiplications only.

5 Results

Table 1 presents the new improved doubling algorithm according to the modifications described in Section 4. The total cost of doubling a divisor on a HEC of genus 2 over fields $\mathbb{F}_{2^n}$ is one field inversion, 9 field multiplications, and 6 field squarings. The saving in multiplications is 47% for the cost of one additional squaring. From a computational point of view, squarings can be neglected in fields $\mathbb{F}_{2^n}$ since their calculation is almost for free.

The main operation of a hyperelliptic cryptosystem is a scalar multiplication with a divisor. If we assume a scalar of bitsize n and a windowing method to perform the multiplication, we have to compute n group doublings and approximately $0.2n$ additions. Thus, with the new proposed formulae, the cryptosystem obtains a speed up of ≈27%[1] compared to a system based on the doubling formulae in Table 3.

Table 1. Optimized explicit formulae for doubling a divisor on a HEC of genus two over $\mathbb{F}_{2^n}$

Input	Weight two reduced divisors $D = (u, v)$ with $u = x^2 + u_1x + u_0$ $v = v_1x + v_0$ furthermore: $h = x$ and $f = x^5 + f_1x + f_0$	
Output	A weight two reduced divisor $D' = (u', v') = [2]D$ with $u' = x^2 + u'_1x + u'_0$; $v' = v'_1x + v'_0$;	
Step	Procedure	Cost
1	Compute resultant r of u and $h + 2v$: $r = u_0$;	–
2	Compute almost inverse $inv \equiv r/\tilde{v} \bmod u_1$: $inv_1 = 1$; $inv_0 = u_1$;	–
3	Compute $k \equiv [(f - hv - v^2)/u] \bmod u$: $w_0 = v_1^2$; $w_1 = u_1^2$; $k_1 = w_1$; $t_1 = u_1k_1$; $k_0 = t_1 + w_0 + v_1$;	$1M + 2S$
4	Compute $s' = kinv \bmod u$: $t_2 = u_0k_0$; $s'_1 = k_0$; $s'_0 = (u_0 + u_1)(k_0 + k_1) + t_1 + t_2$; If $s'_1 = 0$ perform Cantor's Algorithm	$2M$
5	Compute s_1 and s_0u_1: $t_3 = t_2^{-1}(= 1/(rs'_1))$; $w_3 = r^2t_3(= 1/s_1)$; $w_4 = w_3^2$; $s_1 = s_1'^2t_3$; $t_6 = t_1 + k_1s_1(= s_0u_1)$;	$I + 3M + 3S$
6	Compute $z = su$ (Karatsuba): $z_0 = s'_0$; $z_1 = t_6 + s'_1$; $z_2 = w_1$; $z_3 = s_1$;	–
7	Compute $u' = 1/s_1^2((su + h + v)^2 + f)/u^2$: $u'_2 = 1$; $u'_1 = w_4$; $u'_0 = w_4k_1^2 + k_1 + w_3$;	$M + S$
8	Compute $v' \equiv h + z + v \bmod u'$ (Karatsuba): $t_4 = w_3$; $t_7 = t_4 + z_2$; $t_5 = t_7u'_0$; $v'_1 = (z_3 + t_7)(u'_0 + u'_1) + t_4 + t_5 + 1 + z_1 + v_1$; $v'_0 = t_5 + z_0 + v_0$;	$2M$
Total		$I + 9M + 6S$

[1] For this approximation, it is assumed that one field inversion is as costly in time as 7 field multiplications. This approximation is based in several observations of implementations, e.g., in [PWGP03]

6 Conclusions

With this contribution, we introduce a major speed-up of the arithmetic on genus-2 hyperelliptic curves over fields of characteristic two. We were able to reduce the number of multiplications for doubling a divisor by approximately 47%. Since doubling is the crucial step for performing a divisor scalar multiplication, this improvement results in a performance gain of approximately 27% for HECC of genus 2.

References

[ADH94] L.M. Adlemann, J. DeMarrais, 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 *The 1st Algorithmic Number Theory Symposium — ANTS I*, pages 28 – 40, Berlin, May 1994. Springer-Verlag. LNCS 877.

[BCLW02] N. Boston, T. Clancy, Y. Liow, and J. Webster. Genus Two Hyperelliptic Curve Coprocessor. In *Workshop on Cryptographic Hardware and Embedded Systems — CHES 2002*, New York, 2002. Springer Verlag. LNCS 2523.

[Can87] D.G. Cantor. Computing in Jacobian of a Hyperelliptic Curve. In *Mathematics of Computation*, volume 48(177), pages 95 – 101, January 1987.

[EG02] A. Enge and P. Gaudry. A general framework for subexponential discrete logarithm algorithms. *Acta Arith.*, 102:83 – 103, 2002.

[Eng99a] A. Enge. Computing discrete logarithms in high-genus hyperelliptic jacobians in provably subexponential time. Preprint; Available at http://www.math.waterloo.ca/Cond0_Dept/CORR/corr99.html, 1999.

[Eng99b] A. Enge. The extended Euclidean algorithm on polynomials, and the computational efficiency of hyperelliptic cryptosystems, November 1999. Preprint.

[FR94] G. Frey and H.-G. Rück. A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. *Mathematics of Computation*, 62(206):865–874, April 1994.

[FS97] R. Flassenberg and S.Paulus. Sieving in function fields. Available at ftp://ftp.informatik.tu-darmstadt.de/pub/TI/TR/TI-97-13.rafla.ps.gz, 1997. To appear in Experimental Mathematics.

[Ful69] W. Fulton. *Algebraic Curves - An Introduction to Algebraic Geometry.* W. A. Benjamin, Inc., Reading, Massachusetts, 1969.

[Gau00] P. Gaudry. An algorithm for solving the discrete log problem on hyperelliptic curves. In Bart Preneel, editor, *Advances in Cryptology — EUROCRYPT 2000*, pages 19–34, Berlin, Germany, 2000. Springer-Verlag. LNCS 1807.

[GH00] P. Gaudry and R. Harley. Counting Points on Hyperelliptic Curves over Finite Fields. In W. Bosma, editor, *The 4th Algorithmic Number Theory Symposium — ANTS IV*, pages 297 – 312, Berlin, 2000. Springer Verlag. LNCS 1838.

[Har00] R. Harley. Fast Arithmetic on Genus Two Curves. Available at http://cristal.inria.fr/~harley/hyper/, 2000.

[KGM+02] J. Kuroki, M. Gonda, K. Matsuo, Jinhui Chao, and Shigeo Tsujii. Fast Genus Three Hyperelliptic Curve Cryptosystems. In *The 2002 Symposium on Cryptography and Information Security, Japan - SCIS 2002*, Jan.29-Feb.1 2002.

[KO63] A. Karatsuba and Y. Ofman. Multiplication of multidigit numbers on automata. *Sov. Phys. Dokl. (English translation)*, 7(7):595–596, 1963.

[Kob88] N. Koblitz. A Family of Jacobians Suitable for Discrete Log Cryptosystems. In Shafi Goldwasser, editor, *Advances in Cryptology — CRYPTO '88*, pages 94 – 99, Berlin, 1988. Springer-Verlag. LNCS 403.

[Kob89] N. Koblitz. Hyperelliptic Cryptosystems. In Ernest F. Brickell, editor, *Journal of Cryptology*, pages 139 – 150, 1989.

[Kob98] N. Koblitz. *Algebraic Aspects of Cryptography*. Algorithms and Computation in Mathematics. Springer-Verlag, 1998.

[Kri97] Uwe Krieger. signature.c, February 1997. Diplomarbeit, Universität Essen, Fachbereich 6 (Mathematik und Informatik).

[Lan02a] T. Lange. Efficient Arithmetic on Genus 2 Hyperelliptic Curves over Finite Fields via Explicit Formulae. Cryptology ePrint Archive, Report 2002/121, 2002. http://eprint.iacr.org/.

[Lan02b] T. Lange. Inversion-Free Arithmetic on Genus 2 Hyperelliptic Curves. Cryptology ePrint Archive, Report 2002/147, 2002. http://eprint.iacr.org/.

[Lan02c] T. Lange. Weighted Coordinates on Genus 2 Hyperelliptic Curves. Cryptology ePrint Archive, Report 2002/153, 2002. http://eprint.iacr.org/.

[Lan03] T. Lange. Formulae for Arithmetic on Genus 2 Hyperelliptic Curves, 2003. Available at http://www.ruhr-uni-bochum.de/itsc/tanja/preprints.html.

[MCT01] K. Matsuo, J. Chao, and S. Tsujii. Fast Genus Two Hyperelliptic Curve Cryptosystems. In *ISEC2001-31, IEICE*, 2001.

[MDM+02] Y. Miyamoto, H. Doi, K. Matsuo, J. Chao, and S. Tsuji. A Fast Addition Algorithm of Genus Two Hyperelliptic Curve. In *SCIS, IEICE Japan*, pages 497 – 502, 2002. in Japanese.

[Mum84] D. Mumford. Tata lectures on theta II. In *Prog. Math.*, volume 43. Birkhäuser, 1984.

[MWZ96] A.J. Menezes, Y.-H. Wu, and R. Zuccherato. An elementary introduction to hyperelliptic curves. In N. Koblitz, editor, *Algebraic Aspects of Cryptography*, Berlin, Heidelberg, 1996. Springer Verlag.

[Nag00] K. Nagao. Improving group law algorithms for Jacobians of hyperelliptic curves. In W. Bosma, editor, *ANTS IV*, volume 1838 of *Lecture Notes in Computer Science*, pages 439 – 448, Berlin, 2000. Springer Verlag.

[Pel02] J. Pelzl. Hyperelliptic Cryptosystems on Embedded Microprocessors. Master's thesis, Fakultät für Elektrotechnik und Informationstechnik, Ruhr-Universität Bochum, September 2002. (Diplomarbeit).

[Pol78] J. M. Pollard. Monte carlo methods for index computation mod p. *Mathematics of Computation*, 32(143):918–924, July 1978.

[PWGP03] J. Pelzl, T. Wollinger, J. Guajardo, and C. Paar. Hyperelliptic Curve Cryptosystems: Closing the Performance Gap to Elliptic Curves. In C.D. Walter, C.K. Koc, and C. Paar, editors, *Workshop on Cryptographic Hardware and Embedded Systems — CHES 2003*, pages 351–365, Berlin, 2003. Springer Verlag. LNCS 2779.

[PWP03] J. Pelzl, T. Wollinger, and C. Paar. Low Cost Security: Explicit Formulae for Genus-4 Hyperelliptic Curves. In *Tenth Annual Workshop on Selected Areas in Cryptography — SAC 2003*. Springer-Verlag, 2003. LNCS.

[Rüc99] H.-G. Rück. On the discrete logarithm in the divisor class group of curves. *Mathematics of Computation*, 68(226):805–806, 1999.

[Sma99] N.P. Smart. On the Performance of Hyperelliptic Cryptosystems. In *Advances in Cryptology - EUROCRYPT '99*, volume 1592, pages 165 – 175, Berlin, 1999. Springer-Verlag. LNCS 1592.

[SS98] 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*, pages 80 – 94, Berlin, 1998. Springer Verlag. LNCS 1514.

[SS00] Y. Sakai and K. Sakurai. On the Practical Performance of Hyperelliptic Curve Cryptosystems in Software Implementation. In *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, volume E83-A NO.4, pages 692 – 703, April 2000. IEICE Trans.

[SSI98] Y. Sakai, K. Sakurai, and H. Ishizuka. Secure Hyperelliptic Cryptosystems and their Performance. In *Public Key Cryptography*, pages 164 – 181, Berlin, 1998. Springer-Verlag. LNCS 1431.

[Tak02] M. Takahashi. Improving Harley Algorithms for Jacobians of Genus 2 Hyperelliptic Curves. In *SCIS, IEICE Japan*, 2002. in Japanese.

[Thé03] N. Thériault. Index calculus attack for hyperelliptic curves of small genus. In *Advances in Cryptology - ASIACRYPT '03*, Berlin, 2003. Springer Verlag. LNCS.

[vOW99] P. C. van Oorschot and M. J. Wiener. Parallel collision search with cryptanalytic applications. *Journal of Cryptology*, 12(1):1–28, Winter 1999.

[Wol01] T. Wollinger. Computer Architectures for Cryptosystems Based on Hyperelliptic Curves, 2001. Master Thesis, Worcester Polytechnic Institute.

[WP02] T. Wollinger and C. Paar. Hardware Architectures proposed for Cryptosystems Based on Hyperelliptic Curves. In *Proceedings of the 9th IEEE International Conference on Electronics, Circuits and Systems - ICECS 2002*, volume III, pages 1159 – 1163, September 15-18 2002.

A Explicit Formulae for Genus Two HEC

The explicit formulae for the group operations adding and doubling on a HEC of genus 2 over $\mathbb{F}_{2^n}$ are presented in Tables 2 and 3, respectively.

Table 2. Explicit formulae for adding two divisors on a HEC of genus two over $\mathbb{F}_{2^n}$ [Lan03]

Input	Weight two reduced divisors $D_1 = (u_1, v_1)$ and $D_2 = (u_2, v_2)$ with $u_1 = x^2 + u_{11}x + u_{10}$; $u_2 = x^2 + u_{21}x + u_{20}$; $v_1 = v_{11}x + v_{10}$; $v_2 = v_{21}x + v_{20}$; furthermore: $h = x$ and $f = x^5 + f_1x + f_0$	
Output	A weight two reduced divisor $D' = (u', v') = D_1 + D_2$ with $u' = x^2 + u'_1x + u'_0$; $v' = v'_1x + v'_0$;	
Step	Procedure	Cost
1	Compute resultant r of u_1 and u_2: $z_1 = u_{11} - u_{21}$, $z_2 = u_{20} - u_{10}$, $z_3 = u_{11}z_1 + z_2$ $r = z_2z_3 + {z_1}^2u_{10}$	$3M + 1S$
2	Compute almost inverse $inv = r/u_2 \bmod u_1$: $inv_1 = z_1$, $inv_0 = z_2$	–
3	Compute $s' = rs \equiv (v_1 - v_2)inv \bmod u_1$: $w_1 = v_{10} - v_{20}$, $w_2 = v_{11} - v_{21}$, $w_3 = inv_0w_1$, $w_4 = inv_1w_2$ $s'_1 = (inv_0 + inv_1)(w_1 + w_2) - w_3 - w_4(1 + u_{11})$, $s'_0 = w_3 - u_{10}w_4$ If $s_1 = 0$ perform Cantor	$5M$
4	Compute $s'' = x + s_0/s_1 = x + s'_0/s'_1$ and s_1: $w_1 = (rs'_1)^{-1}$, $w_2 = rw_1 (= 1/s'_1)$, $w_3 = {s'_1}^2w_1 (= s_1)$ $w_4 = rw_2 (= 1/s_1)$, $w_5 = w_4^2$, $s''_0 = s'_0w_2$	$I + 5M + 2S$
5	Compute $l' = s''u_2 = x^3 + l'_2x^2 + l'_1x + l'_0$: $l'_2 = u_{21} + s''_0$, $l'_1 = u_{20} + u_{21}s''_0$, $l'_0 = u_{20}s''_0$	$2M$
6	Compute $u' = (s(l + h + 2v_1) - k)u_1^{-1} = x^2 + u'_1x + u'_0$: $u'_1 = s''_0 + l'_2 - u_{11} - w_5$ $u'_0 = (s''_0 - u_{11})(l'_2 - u_{11}) - u_{10} + l'_1 + w_4 + (u_{11} + u_{21})w_5$	$2M$
7	Compute $v' \equiv -(h + l + v_2) \bmod u'$: $w_1 = l'_2 - u'_1$, $w_2 = u'_1w_1 + u'_0 - l'_1$, $v'_1 = w_3w_2 - v_{21} - 1$ $w_4 = u'_0w_1 - l'_0$, $v'_0 = w_3w_4 - v_{20}$	$4M$
Total		$I + 21M + 3S$

Table 3. Explicit formulae for doubling a divisor on a HEC of genus two over $\mathbb{F}_{2^n}$ [Lan03]

Input	Weight two reduced divisors $D = (u, v)$ with $u = x^2 + u_1x + u_0$; $v = v_1x + v_0$; furthermore: $h = h_2x^2 + h_1x + h_0$; where $h_i \in \{0, 1\}$; $f = x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$; where $f_4 \in \{0, 1\}$;	
Output	A weight two reduced divisor $D' = (u', v') = [2]D$ with $u' = x^2 + u'_1x + u'_0$; $v' = v'_1x + v'_0$;	
Step	Procedure	Cost
1	Compute resultant r of u and $h + 2v$: let $\tilde{v} \equiv h + 2v \bmod u$: $\tilde{v}_1 = h_1 + 2v_1 - h_2u_1$; $\tilde{v}_0 = h_0 + 2v_0 - h_2u_0$; $w_0 = v_1^2$; $w_1 = u_1^2$; $w_2 = \tilde{v}_1^2$; $w_3 = u_1\tilde{v}_1$; $r = u_0w_2 + \tilde{v}_0(\tilde{v}_0 - w_3)$;	$3M + 2S$
2	Compute almost inverse $inv \equiv r/\tilde{v} \bmod u_1$: $inv_1 = -\tilde{v}_1$; $inv_0 = \tilde{v}_0 - w_3$;	$-$
3	Compute $k \equiv [(f - hv - v^2)/u] \bmod u$: $w_3 = f_3 + w_1$; $w_4 = 2u_0$; $k_1 = 2(w_1 - f_4u_1) + w_3 - w_4 - v_1h_2$; $k_0 = u_1(2w_4 - w_3 + f_4u_1 + v_1h_2) + f_2 - w_0 - 2f_4u_0 - v_1h_1 - v_0h_2$;	$1M$
4	Compute $s' = kinv \bmod u$: $w_0 = k_0inv_0$; $w_1 = k_1inv_1$; $s'_1 = (inv_0 + inv_1)(k_0 + k_1) - w_0 - w_1 - u_1w_1$; $s'_0 = w_0 - u_0w_1$; If $s'_1 = 0$ perform Cantor's Algorithm	$5M$
5	Compute $s = x + s'_0/s'_1$ and s_1: $w_1 = (rs'_1)^{-1}$; $w_2 = rw_1(= 1/s'_1)$; $w_3 = {s'_1}^2w_1(= s_1)$; $w_4 = rw_2(= 1/s_1)$; $w_5 = w_4^2$; $s_0 = s'_0w_2$;	$I + 5M + 2S$
6	Compute $l = su$: $l_2 = u_1 + s_0$; $l_1 = u_1s_0 + u_0$; $l_0 = u_0s_0$	$2M$
7	Compute $u' = [l^2 + w_4l(2v + h) - w_5(f - vh - v^2)]/u^2$: $u'_1 = 2s_0 + w_4h_2 - w_5$; $u'_0 = s_0^2 + w_4(h_2(s_0 - u_1) + 2v_1 + h_1) + w_5(2u_1 - f_4)$;	$2M + 1S$
8	Compute $v' \equiv -(h + w_3l + v) \bmod u'$: $w_1 = l_2 - u'_1$; $w_2 = u'_1w_1 + u'_0 - l_1$; $v'_1 = w_2w_3 - v_1 - h_1 + h_2u'_1$; $w_4 = u'_0w_1 - l_0$; $v'_0 = w_3w_4 - v_0 - h_0 + h_2u'_0$;	$4M$
Total	 with $h(x) = h_1x + h_0$ and $f_2 = f_3 = 0$	$I + 20M + 5S$ $I + 17M + 5S$