Efficient Doubling on Genus Two Curves over Binary Fields

Tanja Lange[1,⋆] and Marc Stevens[2,⋆]

[1] Institute for Information Security and Cryptology (ITSC),
Ruhr-Universität Bochum Universitätsstraße 150 D-44780 Bochum Germany
lange@itsc.ruhr-uni-bochum.de
http://www.ruhr-uni-bochum.de/itsc/
[2] Department of Mathematics and Computer Science,
Eindhoven University of Technology,
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
m.m.j.stevens@student.tue.nl

Abstract. In most algorithms involving elliptic and hyperelliptic curves, the costliest part consists in computing multiples of ideal classes. This paper investigates how to compute faster doubling over fields of characteristic two.

We derive explicit doubling formulae making strong use of the defining equation of the curve. We analyze how many field operations are needed depending on the curve making clear how much generality one loses by the respective choices. Note, that none of the proposed types is known to be weak – one only could be suspicious because of the more special types. Our results allow to choose curves from a large enough variety which have extremely fast doubling needing only half the time of an addition. Combined with a sliding window method this leads to fast computation of scalar multiples. We also speed up the general case.

Keywords: Hyperelliptic curves, fast arithmetic, explicit group operations, binary fields.

1 Introduction

Hyperelliptic curves of low genus obtained a lot of attention in the recent past for cryptographic applications. It is a rather recent result that they can compete with elliptic curves in terms of efficiency of the group law [Ava03, Lan04a]. The security of low genus hyperelliptic curves is assumed to be similar to that of elliptic curves of the same group size. Here, *low* really means genus $g \leq 3$ by [Gau00, Thé03, GT04, Nag04], and even for $g = 3$ some care has to be taken.

⋆ The work described in this paper has been supported in part by the European Commission through the IST Programme under Contract IST-2002-507932 ECRYPT. The information in this document reflects only the author's views, is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

H. Handschuh and A. Hasan (Eds.): SAC 2004, LNCS 3357, pp. 170–181, 2005.
© Springer-Verlag Berlin Heidelberg 2005

The main operation in protocols based on the *discrete logarithm problem* in an additively written group is the computation of scalar multiples of a group element. Using standard scalar multiplication methods this boils down to additions, doublings, and perhaps some precomputations.

In this paper we concentrate on genus two curves over fields of characteristic two and in detail on doubling formulae for the different types of curves. Obviously, choosing curves defined over $\mathbb{F}_2$ allows very efficient scalar multiplication as shown in the publications on Koblitz curves [GLS00, Lan04b]. However, there are only 6 different isogenie classes and, hence, the choice of curves is rather limited. So there is a trade-off between speed-up and special parameters.

In this article, we give a complete study of all cases of defining equation of the curve where we allow the curve to be *defined over the extension field*. In combination with a windowing method the best case achieves a performance only twice as slow as for Koblitz curves, the reason being that additions remain costly in both cases and there are more of them in the Koblitz curve setting. Clearly, this again is a special choice but the number of non-isomorphic curves has grown considerably.

So far only one very special type of curves has been considered [PWP04] and shown to lead to efficient doubling formulae. Our results improve their formulae and provide clear tables with *all types of defining equations* together with the number of operations and also give the doubling formulae.

After the submission of this paper the authors found a further work in special curves [BD04]. They obtain less efficient doublings, but also do a complete study of all kinds of curves. Even more recently, Duquesne (see [ACD+04]) made improvements for the case where $\deg h = 2$ and $h_0 \neq 0$.

We now briefly state the background needed on hyperelliptic curves and then give a complete study of the doubling formulae. Section 7 provides timings for the different cases giving evidence that the claimed speed up can actually be obtained. We end with some remarks on side channel attacks.

2 Basic Notations and Preliminaries

We refer the interested reader to [FL03, Lor96, MWZ98, Sti93] for mathematical background. For the scope of this paper we only try to motivate the representation of the group elements and the group law as this is what we concentrate on in the remainder of the paper.

Let $\mathbb{F}_q, q = 2^\ell$, be a finite field of characteristic 2 and let C be a hyperelliptic curve defined over $\mathbb{F}_q$. In cryptography one usually deals with curves C given by

$$\begin{gathered} C: \quad Y^2 + h(X)Y = f(X) \\ h, f \in \mathbb{F}_q[X],\ f \text{ monic},\ \deg f = 2g+1,\ \deg h \leq g \end{gathered} \tag{1}$$

for which no point $(x, y) \in C$ satisfies both partial derivative equations. For characteristic 2 one needs to have a non-zero h to achieve this quality. The integer g appearing in (1) is called the *genus of* C. We concentrate on curves of genus 2.

The group one uses for cryptographic applications is the ideal class group $\mathrm{Cl}(C/\mathbb{F}_q)$ of C over $\mathbb{F}_q$. This is the quotient of the group of fractional ideals of $\mathbb{F}_q[X,Y]/(Y^2+h(X)Y+f(X))$ by the group of principal ideals. Like in the case of quadratic imaginary fields in each ideal class one finds an ideal generated by two polynomials $\langle u(X), v(X)+Y\rangle$.

There is a unique ideal of minimal degree in each class. Actually, each element D of $\mathrm{Cl}(C/\mathbb{F}_q)$ can be represented by an ordered pair of polynomials $D=[u(X),v(X)]$, with $u,v\in\mathbb{F}_q, \deg v < \deg u \leq g$ and u monic satisfying $u|v^2+hv+f$.

3 Group Law

The group operation in $\mathrm{Cl}(C/\mathbb{F}_q)$ is performed by first computing the product of the ideals and then reducing it modulo the principal ideals. This is the principle behind Cantor's algorithm [Can87, Kob89].

Obviously, this algorithm has to depend on the properties of the input – to derive explicit formulae one needs to study additions independently from doublings. For a complete study of all possible inputs together with formulae we refer to [Lan04a]. In this paper we concentrate on doublings for genus 2 curves in the most frequent case where the input $[u,v]$ has full degree and u and h do not have a root in common. Accordingly, we assume from now on

$$D=[u,v],\quad \deg u = 2,\quad \mathrm{res}(h,u)\neq 0.$$

Put $u=x^2+u_1x+u_0,\ v=v_1x+v_0$. Composing $[u,v]$ with itself should result in a class $[u_{\mathrm{new}},v_{\mathrm{new}}]$, where

$$\begin{aligned} u_{new} &= u^2,\\ v_{new} &\equiv v \mod u,\\ u_{new} &\mid v_{new}^2+v_{new}h+f. \end{aligned}$$

Then this class is reduced to obtain $[u',v']=2[u,v]$.

We fix the notation to refer to the coefficient of X^i in a polynomial $p(X)$ as p_i.

The following expressions follow those in [Lan04a] and are explained there. We slightly modified them for the way we will apply them.

$$\begin{aligned} \tilde{v} &\equiv h \mod u\\ r &= \mathrm{res}(\tilde{v},u)\\ inv' &\equiv r\tilde{v}^{-1} \mod u \equiv \tilde{v}_1x+\tilde{v}_0+u_1\tilde{v}_1 \mod u\\ k &= (f+hv+v^2)/u = k'+u(x+f_4),\ \text{with } k'\equiv k \mod u\\ s' &\equiv k'\ inv' \mod u\\ s'' &= s' \text{ made monic}\\ l'' &= s''u\\ u' &= s''^2+(kr^2/{s_1'}^2+hs''r/s_1')/u\\ v' &\equiv h+(l''s_1'/r+v) \mod u' \end{aligned}$$

Remark 1. In the actual formulae we do not follow these steps literally. It turns out to be more efficient to perform the inversion of r and s_1' jointly using Montgomery's trick.

Going into the details of these expressions one notices that the actual execution of the steps depends on the coefficients of the curve. We will present formulae for three different cases: $\deg h = 1$, $\deg h = 2$ with obtainable $h_0 = 0$ and $\deg h = 2$ in general. In the two first cases we have $h_0 = 0$ and r will simplify (to the form $r = u_0\tilde{r}$ for some $\tilde{r}$). This allows us to cancel r in the expressions, so its inverse is not needed anymore. This is how the major speedup is obtained in the formulae. In the case of general h we need the inversion of r and perform the inversion of r and s_1' jointly as explained above.

We now study the different expressions for h separately, always performing isomorphic transformations first to achieve as many zero coefficients as possible. In characteristic 2, curves with constant h are known to be supersingular. This makes them weak under the Frey-Rück attack [FR94, Gal01] and, hence, they should be avoided for DL systems.[1] Note, that in any case one needs to make sure that the extension field of $\mathbb{F}_q$ the Tate pairing maps to has large enough degree to avoid this attack.

4 Case deg $h = 1$

In this section we assume $\deg h = 1$. One can obtain an isomorphic curve where $f_4 = h_0 = 0$ and h_1 is divided by any cube a^3. In this case it is much more useful to have h_0 zero (at the cost of a non-zero f_3) as mentioned above. It is suggested to choose the cube a^3 such that $\frac{a^3}{h_1}$ is 'small'. This allows the multiplications with it to be performed via additions and thus they are almost for free. If, as usual, one chooses $\mathbb{F}_{2^n}$ with n odd there are no non-trivial cube roots of unity. Hence, there is always an a such that $a^3 = h_1$. For even n this happens with probability 1/3. This isomorphic curve is obtained using the following change of variables and dividing the equation by a^{10}:

$$Y \leftarrow a^5\tilde{Y} + a^4\sqrt{f_4 + \frac{h_0}{h_1}}\tilde{X}^2, \quad X \leftarrow a^2\tilde{X} + \frac{h_0}{h_1}$$

Hence, we obtain a curve of the form $Y^2+h_1XY = X^5+f_3X^3+f_2X^2+f_1X+f_0$, usually with $h_1 = 1$. Adding a linear factor to the substitution of $\tilde{Y}$ one can achieve $f_2 = 0$ with probability 1/2. A constant term leads to $f_1 = 0$. Hence, there are only two free parameters f_3, f_0 as opposed to three in the general case showing that the type is indeed special.

[1] These curves have found an application in pairing based cryptography. The explicit formulae for this case together with information necessary to compute the pairing are the topic of an upcoming paper.

Table 1. Doubling $\deg h = 1$, $\deg u = 2$

Doubling $\deg h = 1$, $\deg u = 2$				
Input	$[u, v], u = x^2 + u_1 x + u_0, v = v_1 x + v_0$; h_1^2, h_1^{-1}			
Output	$[u', v'] = 2[u, v]$			
Step	Expression	$h_1 = 1$	h_1^{-1} small	h_1 arbitrary
1	compute rs_1:	3S	3S	3S
	$z_0 = u_0^2$, $k_1' = u_1^2 + f_3$;			
	$w_0 = f_0 + v_0^2 (= rs_1'/h_1^3)$;			
	If $w_0 = 0$ see below			
2	compute $1/s_1$ and s_0'':	I, 2M	I, 2M	I, 2M
	$w_1 = (1/w_0) z_0 (= h_1/s_1)$;			
	$z_1 = k_1' w_1$, $s_0'' = z_1 + u_1$;			
3	compute u':	2S	S, 2M	S, 2M
	$w_2 = h_1^2 w_1$, $u_1' = w_2 w_1$;			
	$u_0' = s_0''^2 + w_2$;			
4	compute v':	S, 3M	S, 3M	S, 5M
	$w_3 = w_2 + k_1'$;			
	$v_1' = h_1^{-1}(w_3 z_1 + w_2 u_1' + f_2 + v_1^2)$;			
	$v_0' = h_1^{-1}(w_3 u_0' + f_1 + z_0)$;			
total		I, 6S, 5M	I, 5S, 7M	I, 5S, 9M
Special case $s = s_0$				
$2'$	compute s and precomputations:	1M	1M	2M
	$s_0 = (1/h_1) k_1'$, $w_1 = u_0 s_0 + v_0$;			
$3'$	compute u':	S	S	S
	$u_0' = s_0^2$;			
$4'$	compute v':	2M	2M	2M
	$w_2 = s_0(u_1 + u_0') + v_1 + h_1$;			
	$v_0' = u_0' w_2 + w_1$;			
total		4S, 3M	4S, 3M	4S, 4M

With the new curve coefficients the expressions r and s will simplify to:

$$\begin{aligned} r &= h_1^2 u_0, \\ s_1' &= h_1 k_0', \\ s_0' &= u_1 s_1' + h_1 u_0 u_1. \end{aligned}$$

Since $f + hv + v^2 = uk' + u^2 x$ we also have that

$$\begin{aligned} f_0 + v_0^2 &= u_0 k_0' \quad (= rs_1'/h_1^3) \\ f_1 + u_0^2 + h_1 v_0 &= u_1 k_0' + u_0 k_1' \\ f_2 + v_1(h_1 + v_1) &= k_0' + u_1 k_1' \\ u_1^2 + f_3 &= k_1' \end{aligned}$$

making it very cheap to calculate rs'_1 as the exact coefficients of k' are not necessary. We present the doubling formulae for this case in Table 1. The operations are counted for the case that $h_1 = 1$, h_1^{-1} is 'small' (multiplications with h_1^{-1} are not counted), and arbitrary h_1. Both h_1^2 and h_1^{-1} are precomputed. In Step 2 the inversion and multiplication with z_0 can also be replaced by a division as the inverse is not used later on.

5 Case deg $h = 2$

If h is of degree two then in general we cannot make any of its coefficients zero, however, it is possible to make a change of coordinates to obtain $h_2 = 1$ and $f_3 = f_2 = 0$. The case $h_0 = 0$ allows for a significant speedup, however, we can only obtain h_0 zero if there exists a b such that $b^2 + bh_1 = h_0$ and this will be at the cost of a non-zero f_4. If there is no such b, i. e. $\mathrm{Tr}(h_0/h_1^2) \neq 0$, then choose $b = f_4$ making f_4 zero. This can be done by the following change of variables and dividing the resulting equation by h_2^{10}.

$$Y \leftarrow h_2^5\tilde{Y} + f_3h_2\tilde{X} + \frac{f_3(f_3 + h_1h_2 + f_3h_2^2) + f_2h_2^2}{h_2^3}, \quad X \leftarrow h_2^2\tilde{X} + b$$

5.1 Case deg $h = 2, h_0 = 0$

First, we assume that we have obtained $h_0 = 0$ leading to an equation

$$Y^2 + (X^2 + h_1X)Y = X^5 + f_4X^4 + f_1X + f_0,$$

Using a quadratic term in the transformation of Y, one can additionally obtain $f_4 = 0$ with probability 1/2, namely if $\mathrm{Tr}((b + f_4)/h_2^2) = 0$, with b as above. If, as usual, one chooses $\mathbb{F}_{2^n}$ with n odd then one can always obtain either $f_4 = 0$ or $f_4 = 1$. Accordingly, one has three free parameters h_1, f_1, f_0.

Then the expressions for r and s will simplify to:

$$\begin{aligned} r &= u_0(u_0 + h_1^2 + h_1u_1) \\ s'_1 &= h_1k'_0 + u_0k'_1 + u_1k'_0 \\ s'_0 &= u_1s'_1 + u_0k'_0 + h_1u_0k'_1 \end{aligned}$$

And since $f + hv + v^2 = uk' + u^2(x + f_4)$ we also have that

$$\begin{aligned} f_0 + v_0^2 + h_0v_0 + f_4u_0^2 &= u_0k'_0 \\ f_1 + u_0^2 + h_1v_0 + h_0v_1 &= u_1k'_0 + u_0k'_1 \\ v_0 + v_1(h_1 + v_1) + f_4u_1^2 &= k'_0 + u_1k'_1 \\ u_1^2 + v_1 &= k'_1. \end{aligned}$$

Table 2. Doubling, $\deg h = 2$, $h_0 = 0$, $\deg u = 2$

Doubling, $\deg h = 2$, $h_0 = 0$, $\deg u = 2$			
Input	$[u, v], u = x^2 + u_1 x + u_0, v = v_1 x + v_0$; h_1^2		
Output	$[u', v'] = 2[u, v]$		
Step	Expression	h_1 small	h_1 arbitrary
1	compute k_1' and precomputations: $z_0 = u_0^2$, $z_1 = u_1^2$, $w_0 = v_1(h_1 + v_1)$; $k_1' = z_1 + v_1$, $z_2 = h_1 u_1$, $z_3 = f_4 u_1$;	3S, M	2S, 3M
2	compute resultant $r =$res$(\tilde{v}, u)$: $\tilde{r} = u_0 + h_1^2 + z_2 = (r/u_0)$;		
3	compute s_1' and almost s_0': $w_2 = u_1(k_1' + z_3) + w_0$, $w_3 = v_0 + h_1 k_1'$; $s_1' = f_1 + z_0 + h_1 w_2$; $m_0 = w_2 + w_3 (= (s_0' - u_1 s_1')/u_0)$; If $s_1' = 0$ see below	M	3M
4	compute $s'' = x + s_0/s_1$ and s_1: $w_2 = 1/(s_1')(= 1/r s_1)$, $w_3 = u_0 w_2$; $w_4 = \tilde{r} w_3 (= 1/s_1)$, $w_5 = w_4^2$; $s_0'' = u_1 + m_0 w_3$;	I, S, 3M	I, S, 3M
5	compute u': $z_4 = f_4 w_4$, $u_1' = w_4 + w_5$; $u_0' = s_0''^2 + w_4(s_0'' + h_1 + u_1 + z_4)$;	S, 2M	S, 2M
6	compute v': $z_5 = w_2(m_0^2 + k_1'(s_1' + h_1 m_0))$; $z_6 = s_0'' + h_1 + z_4 + z_5$; $v_0' = v_0 + z_2 + z_1 + w_4(u_0' + z_3) + s_0'' z_6$; $v_1' = v_1 + w_4(u_1' + s_0'' + f_4 + u_1) + z_5$;	S, 5M	S, 6M
total		I, 6S, 12M	I, 5S, 17M
Special case $s = s_0$			
$3'$	compute s and precomputations: $w_1 = 1/\tilde{r}$, $s_0 = m_0 w_1$; $w_2 = u_0 s_0 + v_0 + h_0$;	I, 2M	I, 2M
$4'$	compute u': $u_0' = s_0^2 + s_0$;	S	S
$5'$	compute v': $w_1 = s_0(u_1 + u_0') + u_0' + v_1 + h_1$; $v_0' = u_0' w_1 + w_2$;	2M	2M
total		I, 4S, 6M	I, 3S, 10M

Table 2 presents the operations for the case of $h_0 = 0$. In the formulae there are two counted multiplications with f_4 and five with h_1 which are cheaper or for free when f_4 resp. h_1 is 'small'. Furthermore, h_1^2 is precomputed.

Table 3. Doubling, $\deg h = 2$, $\deg u = 2$

Doubling, $\deg h = 2$**,** $\deg u = 2$				
Input	$[u, v], u = x^2 + u_1 x + u_0, v = v_1 x + v_0$; h_0^2			
Output	$[u', v'] = 2[u, v]$			
Step	Expression	h_1, h_0 small	h_1 small	h_i arbitrary
1	compute k_1' and precomputations:	3S	3S	2S,2M
	$z_0 = u_0^2$, $z_1 = u_1^2$, $w_0 = v_1(h_1 + v_1)$;			
	$k_1' = z_1 + v_1$, $w_1 = h_1 u_0$;			
2	compute resultant $r = \mathrm{res}(\tilde{v}, u)$:	1M	2M	2M
	$w_2 = h_0 u_1$, $r = h_0^2 + z_0 + (h_1 + u_1)(w_1 + w_2)$;			
3	compute s_1' and almost s_0':	1S,2M	1S,4M	1S,5M
	$s_1' = f_1 + z_0 + h_0 z_1 + h_1(u_1 k_1' + w_0)$;			
	$m_0 = f_0 + w_1 k_1' + h_0 w_0 + v_0^2 (= s_0' - u_1 s_1')$;			
	If $s_1' = 0$ see below			
4	compute $s'' = x + s_0/s_1$ and s_1:	I, 2S, 5M	I, 2S, 5M	I, 2S, 5M
	$w_1 = 1/(r s_1') (= 1/r^2 s_1)$, $w_2 = r w_1 (= 1/s_1')$;			
	$w_3 = s_1'^2 w_1 (= s_1)$;			
	$w_4 = r w_2 (= 1/s_1)$, $w_5 = w_4^2$, $s_0'' = u_1 + m_0 w_2$;			
5	compute l':	2M	2M	2M
	$l_2' = u_1 + s_0''$, $l_1' = u_1 s_0'' + u_0$, $l_0' = u_0 s_0''$;			
6	compute u':	S, M	S, M	S, M
	$u_0' = s_0''^2 + w_4(s_0'' + u_1 + h_1)$;			
	$u_1' = w_4 + w_5$;			
7	compute v':	4M	4M	4M
	$w_1 = l_2' + u_1'$, $w_2 = u_1' w_1 + u_0' + l_1'$;			
	$v_1' = w_2 w_3 + v_1 + h_1 + u_1'$;			
	$w_2 = u_0' w_1 + l_0'$, $v_0' = w_2 w_3 + v_0 + h_0 + u_0'$;			
total		I,7S,15M	I,7S,18M	I, 6S, 21 M
Special case $s = s_0$				
$3'$	compute s and precomputations:	I,2M	I,2M	I,2M
	$w_1 = 1/r$, $s_0 = m_0 w_1$, $w_2 = u_0 s_0 + v_0 + h_0$;			
$4'$	compute u':	S	S	S
	$u_0' = s_0^2 + s_0$;			
$5'$	compute v':	2M	2M	2M
	$w_1 = s_0(u_1 + u_0') + u_0' + v_1 + h_1$, $v_0' = u_0' w_1 + w_2$;			
total		I,5S,7M	I,5S,10M	I,4S,13M

5.2 Case $\deg h = 2, h_0 \neq 0$

For completeness we include the formulae for the general case $\deg h = 2, h_2 = 1, h_0 \neq 0$. Compared to the doubling formulae in [Lan04a] we manage to trade one multiplication for a squaring which is usually more efficient in characteristic 2. To this aim we need to include one fixed precomputation h_0^2 to the curve parameters.

For h of full degree with non-zero h_0 we can transform to

$$Y^2 + (X^2 + h_1 X + h_0)Y = X^5 + f_1 X + f_0.$$

Accordingly h_2 and f_4 are not mentioned in the formulae.

If one is willing to choose either (or both) h_1 or h_0 'small', we can get much more operations for free.

Here we only used that $f + hv + v^2 = uk' + u^2(x + f_4)$ to calculate s' cheaper and that $s_0' = u_1 s_1' + m_0$ for some relatively simple m_0:

$$s_1' = f_1 + u_0^2 + h_0 u_1^2 + h_1(u_1 k_1' + v_1(h_1 + v_1))$$
$$m_0 = f_0 + h_1 u_0 k_1' + h_0 v_1(h_1 + v_1) + v_0^2$$

6 Summary

The previous sections showed a complete study of doubling formulae depending on the type of h. We summarize the findings in Table 4 listing only the general cases; for h of degree 1 and general h the case f_4 not small does not apply since then $f_4 = 0$.

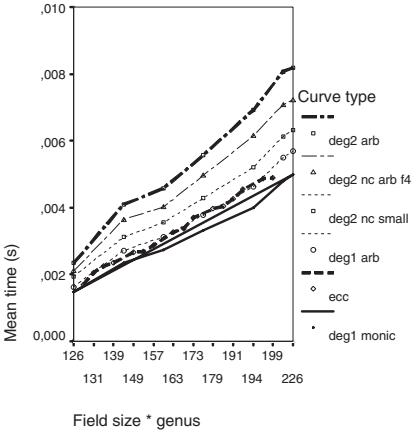
Table 4. Overview

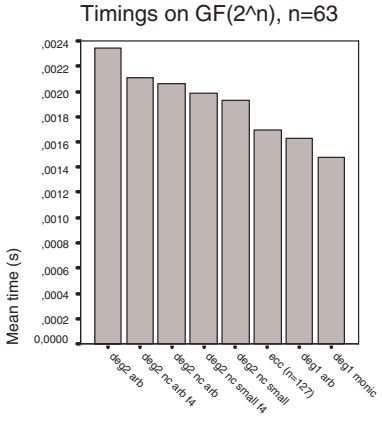
	$h = X$	$h = h_1 X$		$h = X^2 + h_1 X$		$h = X^2 + h_1 X + h_0$		
		h_1^{-1} small		h_1 small		h_1, h_0 small	h_1 small	
f_4 small	I, 6S, 5M	I, 5S, 7M	I, 5S, 9M	I, 6S, 10M	I, 5S, 15M	I, 7S, 15M	I, 7S, 18M	I, 6S, 21M
f_4 arb.	n. a.	n. a.	n. a.	I, 6S, 12M	I, 5S, 17M	n. a.	n. a.	n. a.

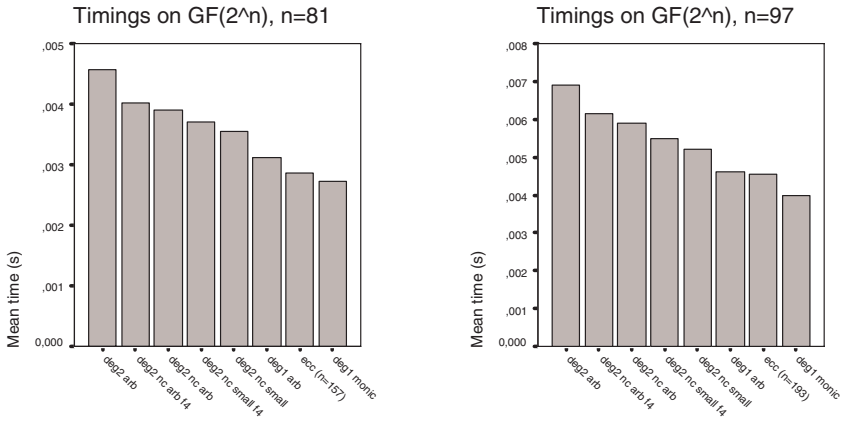
7 Experimental Results

We implemented our new formulae using the NTL library. We used a simple sliding windows method with window size 3 to perform the scalar multiplication in all tests. The extension fields over $\mathbb{F}_2$ were all defined by means of a trinomial. Magma was used to create good random curve equations.

We tested the different cases for $F = \mathbb{F}_{2^{83}}$ and $F = \mathbb{F}_{2^{97}}$ and we used 1 as synonym for 'small' which means that for $\deg h = 1$ the two cases $h_1 = 1$ and h_1 'small' were combined. We also included the elliptic curve case where the field is twice as big to have comparable security, here we also used the same sliding windows method.







All tests were performed on a AMD Athlon XP 2500+ laptop running Gentoo linux. We used the NTL library to perform the field arithmetic. For all field extensions we used a trinomial or a pentanomial for the field arithmetic. Specifically for $n = 63, 81, 97, 127, 193$ we used a trinomial and for $n = 157$ a pentanomial. For the three bar graphs we have chosen field sizes for HEC and ECC such that the group orders were very close and that the arithmetic could be done with a trinomial to make a fair comparison. However for $n = 81$ there was no such comparable field extension for ECC. Therefore we have chosen for a smaller group order ($n = 157$) and arithmetic based upon a pentanomial. The cases included in the graphs are:

deg2 arb:	The case where $\deg h = 2$ and $h_0 \neq 0$
deg2 nc arb f4:	The case where $\deg h = 2$, $h_0 = 0$, $f_4 \neq 0$;
deg2 nc arb:	The case where $\deg h = 2$, $h_0 = 0$, $f_4 = 0$;
deg2 nc small f4:	The case where $\deg h = 2$, $h_0 = 0$, $f_4 \neq 0$ and h_1 small;
deg2 nc small:	The case where $\deg h = 2$, $h_0 = 0$, $f_4 = 0$ and h_1 small;
deg1 arb:	The case where $\deg h = 1$;
deg1 monic:	The case where $\deg h = 1$ and $h_1 = 1$;
ecc:	ECC on the according field extension.

8 Conclusion and Outlook

We have given a complete study of doubling formulae reaching the minimal number of field operations in the respective cases and achieving a lower operation count compared to the special cases [PWP04, BD04] published so far.

The addition formulae depend far less on the equation of h and not on that of f. One can save one multiplication in case of $h_1 \in \{0, 1\}$; all other special choices allow to save at most some additions.

Accordingly, the operation counts for addition and doubling differ quite significantly, especially in the case of $h = X$, making sidechannel attacks feasible.

Following Coron's double-and-always-add countermeasure would lead to including many of the costly additions.

We assume first the setting of rather low storage capacities such that precomputations cannot be made. Then one uses the NAF of the scalar to minimize the Hamming weight. This means that every addition (ADD) is followed by at least two doublings. As doublings have become rather cheap now, we propose to follow the strategy of putting the fixed sequence of ... DBL, ADD, DBL, DBL, DBL, ADD, DBL, DBL, DBL, ... (or even four doublings following an addition). This can be achieved by inserting several dummy doublings and only very few dummy additions.

The situation looks much more friendly if we are allowed to store precomputed multiples of the base class D. Möllers windowing method [MÖ1] allows to obtain a uniform side channel by using only non-zero coefficients in the expansion.

In this article we restricted our attention to affine coordinates as in binary fields an inversion is not prohibitively expensive. It is planned to extend the formulae to inversion-free coordinate systems as well; our findings give new insight in even more efficient choices of the additional coordinates. Furthermore, the lower operation count obtained here for the special choices applies also to other coordinate systems. Projective and new coordinates bear the additional advantage that randomization techniques [Ava04] can be applied to avoid DPA, e. g. all coordinates can be multiplied by (powers of) a random integer leading to a different representation of the same ideal class. For affine coordinates one can randomize the curve equation by making a transformation to an isomorphic curve. This leaves invariant the classes of $\deg h = 1$ and $\deg h = 2$ but one cannot keep all best choices made above and hence, cannot achieve the lowest number of operations. As our publication details all possible cases one now has the choice to trade efficiency for a larger class of curves and hence better randomization.

References

[ACD+04] R. Avanzi, H. Cohen, C. Doche, G. Frey, T. Lange, K. Nguyen, and F. Vercauteren. *The Handbook of Elliptic and Hyperelliptic Curve Cryptography.* CRC, 2004. to appear.

[Ava03] R. M. Avanzi. Aspects of Hyperelliptic Curves over Large Prime Fields in Software Implementations. Cryptology ePrint Archive, Report 2003/253, 2003. to appear in CHES 2004.

[Ava04] R. M. Avanzi. Countermeasures Against Differential Power Analysis for Hyperelliptic Curve Cryptosystems. In *Proceedings of CHES 2003*, volume 2779 of *LNCS*, pages 366–381, 2004.

[BD04] B. Byramjee and S. Duqesne. Classification of genus 2 curves over $\mathbb{F}_{2^n}$ and optimization of their arithmetic. Cryptology ePrint Archive, Report 2004/107, 2004. `http://eprint.iacr.org/`.

[Can87] D. G. Cantor. Computing in the Jacobian of a hyperelliptic curve. *Math. Comp.*, 48:95–101, 1987.

[FL03] G. Frey and T. Lange. Mathematical Background of Public Key Cryptography. Technical Report 10, IEM Essen, 2003.

[FR94] 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.

[Gal01] S. D. Galbraith. Supersingular Curves in Cryptography. In *Advances in Cryptology – Asiacrypt 2001*, volume 2248 of *Lect. Notes Comput. Sci.*, pages 495–513. Springer, 2001.

[Gau00] P. Gaudry. An algorithm for solving the discrete log problem on hyperelliptic curves. In *Advances in Cryptology – Eurocrypt'2000*, Lect. Notes Comput. Sci., pages 19–34. Springer, 2000.

[GLS00] C. Günther, T. Lange, and A. Stein. Speeding up the Arithmetic on Koblitz Curves of Genus Two. In *Selected Areas in Cryptography – SAC 2000*, volume 2012 of *Lect. Notes Comput. Sci.*, pages 106–117. Springer, 2000.

[GT04] P. Gaudry and E. Thomé. A double large prime variation for small genus hyperelliptic index calculus. Cryptology ePrint Archive, Report 2004/153, 2004.

[Kob89] N. Koblitz. Hyperelliptic cryptosystems. *J. Cryptology*, 1:139–150, 1989.

[Lan04a] T. Lange. Formulae for Arithmetic on Genus 2 Hyperelliptic Curves. `http://www.itsc.ruhr-uni-bochum.de/tanja/preprints.html`, 2004. to appear in J. AAECC.

[Lan04b] T. Lange. Koblitz curve cryptosystems. *Finite Fields and Their Applications*, 2004. to appear.

[Lor96] D. Lorenzini. *An Invitation to Arithmetic Geometry*, volume 9 of *Graduate studies in mathematics*. AMS, 1996.

[MÖ1] B. Möller. Securing elliptic curve point multiplication against side-channel attacks. In *Proc. of ISC 2001*, pages 324–334, 2001.

[MWZ98] A. J. Menezes, Y.-H. Wu, and R. Zuccherato. An Elementary Introduction to Hyperelliptic Curves. In N. Koblitz, editor, *Algebraic Aspects of Cryptography*, pages 155–178. Springer, 1998.

[Nag04] K. Nagao. Improvement of Thériault Algorithm of Index Calculus for Jacobian of Hyperelliptic Curves of Small Genus. Cryptology ePrint Archive, Report 2004/161, 2004.

[PWP04] J. Pelzl, T. Wollinger, and C. Paar. Special Hyperelliptic Curve Cryptosystems of Genus Two: Efficient Arithmetic and Fast Implementation. In *Embedded Cryptographic Hardware: Design and Security*, 2004. to appear.

[Sti93] H. Stichtenoth. *Algebraic Function Fields and Codes*. Springer, 1993.

[Thé03] N. Thériault. Index calculus attack for hyperelliptic curves of small genus. In *Advances in cryptology – Asiacrypt 2003*, volume 2894 of *Lect. Notes Comput. Sci.*, pages 75–92. Springer, 2003.