Countermeasures against Differential Power Analysis for Hyperelliptic Curve Cryptosystems

Roberto M. Avanzi*

Institut für Experimentelle Mathematik
University of Duisburg-Essen (Essen site)
Ellernstraße 29 – 45326 Essen, Germany
mocenigo@exp-math.uni-essen.de

Abstract. In this paper we describe some countermeasures against differential side-channel attacks on hyperelliptic curve cryptosystems. The techniques are modelled on the corresponding ones for elliptic curves. The first method consists in picking a random group isomorphic to the one where we are supposed to compute, transferring the computation to the random group and then pulling the result back. The second method consists in altering the internal representation of the divisors on the curve in a random way. The impact of the recent attack of L. Goubin is assessed and ways to avoid it are proposed.

Keywords. Public-key cryptography, Side-channel attacks, Differential power analysis (DPA), Timing attacks, Hyperelliptic curves, Smart cards.

1 Introduction

The use of Jacobian varieties of hyperelliptic curves in discrete logarithm cryptosystems was proposed by N. Koblitz as early as 1988 [17,18] as an alternative to elliptic curves. Hyperelliptic curves are a generalisation of elliptic curves: the latter are just the hyperelliptic curves of genus one.

Until very recently, however, elliptic curve cryptosystems (short: `ecc`) have been perceived as faster than hyperelliptic systems (short: `hecc`, but some other authors prefer abbreviations like `hec` or `hcc`) of genus at least two and offering comparable security. An important milestone in the road to change this perception happened in September 2002: at the ECC 2002 Workshop in Essen, K. Nguyen of Philips Research reported on his implementation on a hardware simulator of T. Lange's projective formulae for genus 2 [25]. This showed for the first time that the performance of `hecc` can be competitive, even for smart card applications. Shortly afterwards J. Pelzl, T. Wollinger, J. Guajardo and C. Paar [38] obtained efficient formulae for genus 3 hyperelliptic Jacobians in all characteristics improving on the work of [23].

* The work described in this paper has been supported by the Commission of the European Communities through the IST Programme under Contract IST-2001-32613 (see http://www.arehcc.com or http://www.arehcc.org).

C.D. Walter et al. (Eds.): CHES 2003, LNCS 2779, pp. 366–381, 2003.

This raises immediately the issue of the security of hecc against side-channel attacks, first introduced in the form of timing attacks in [20] and then simple and differential power analysis (SPA and DPA) [21,22]. These attacks measure some leaked information of a cryptographic device (e.g. timing, power consumption, electromagnetic radiation) while it processes its inputs. For historical reasons we just write DPA also when exploiting leaked information other than power consumption. If a single input is used, the process is referred to as a *Simple Power Analysis* (SPA), and if several different inputs are used together with statistical tools, it is called *Differential Power Analysis* (DPA). We are concerned here with the second type of analysis.

SPA attempts to recover the secret scalar from one observation of the sequence of operations: For example, in a simple double-and-add algorithm the number of consecutive group doublings minus one is the amount of zeros between two ones in the binary representation of the scalar. For ecc there exist two anti-SPA strategies.

The first strategy aims at making the sequence of group operations seemingly independent from the scalar. In the "double-and-add-always" [7] scalar multiplication method an addition is performed after each doubling, even if the corresponding digit of the scalar is zero: This can be done of course in any group, including the Jacobians of hyperelliptic curves. For curves in odd characteristic admitting a particular form, the "Montgomery" method [33,37] allows a very fast computation where the y-coordinate is not used. Analogues of this idea exists for binary curves [1,30] and for all elliptic curves over prime fields [3,8].

The second strategy relies on indistinguishable addition and doubling formulae. They exist for many classes of curves, such as those in Hessian [15,40] or in Jacobi Form [28]. E. Brier and M. Joye found such formulae for elliptic curves over all fields [3]. Another way of pursuing this strategy is to insert dummy operations: for an even characteristic example see [2].

At the moment of this writing little has been done to protect specifically a hecc against SPA. The only currently known methods are the generic ones such as: (i) the insertion of dummy group additions in the scalar multiplication algorithm (as in the "double-and-add-always" method) or (ii) the insertion of dummy field operations in the addition and doubling formulae. T. Lange [27] remarked that the latter can be realized easily and efficiently with the genus 2 affine formulae: this is particularly important for the applications, since the formulae are simpler than in the genus 3 and 4 cases, and the security of genus 2 curves is better understood.

Henceforth we shall always assume that the scalar multiplication algorithm has been made immune from SPA by at least one of these two techniques.

In a DPA the side-channel information collected upon processing of several different inputs is correlated with the value of a boolean function χ of the internal representation of the operands in the cryptographic hardware. The attacker, which is assumed to know the algorithm, *guesses* that the hardware will perform a specific operation at a given point – for example which operand from a table is reused, or which branch is taken – depending on some part of the secret information to be elicited. The inputs are then sorted in two sets according to the values of χ on the corresponding guessed outputs. If the statistical correlation with the leaked information is good, the guess was correct. This leads to attacks which require time linear in the length of the cryptographic operation. We refer the reader to [20,21,22] for more details. Short descriptions can also be found in [7, § 3] and [16, §§ 3.2 and 3.3].

The present work is a first attempt to harden `hecc` *against DPA. In the next section we develop hyperelliptic curve analogues of Coron's third countermeasure [7] (point randomisation) and of the curve randomisation method of M. Joye and C. Tymen [16]. The impact of the recent results of L. Goubin [13] is discussed. We also discuss the applicability of such techniques in light of: (i) the state of the art of explicit formulae for divisor addition and (ii) security results for specific classes of varieties. An appendix contains an example of explicit transformations for the curve randomisations in genus* 2.

2 The Techniques

2.1 Curve Randomisation

An excellent, low brow introduction to the subject of hyperelliptic curves, with a detailed derivation of the facts used below, is given in [31]: Our notation is slightly different, but conforms to that of [24,25,26,27,38].

The idea behind curve randomisation techniques is to "scramble" all the bits of the computation in a (hopefully) unpredictable way. It consists in picking a random group isomorphic to the one on which the cryptosystem is based, transferring the computation to it and then pulling the result back.

More formally, let $\mathcal{C}$ and $\mathcal{C}'$ be two hyperelliptic curves of genus $g \geqslant 1$ over a finite field $\mathbb{F}_q$. Suppose that $\phi : \mathcal{C} \to \tilde{\mathcal{C}}$ is an $\mathbb{F}_q$-isomorphism which is easily extended to an $\mathbb{F}_q$-isomorphism of the Jacobians $\phi : \mathcal{J}(\mathcal{C}) \to \mathcal{J}\big(\tilde{\mathcal{C}}\big)$. Let us further assume that ϕ, together with its inverse, is computable in a reasonable amount of time, *i.e.* small with respect to the time of a scalar multiplication. We do not require *a priori* the computation time of ϕ to be negligible with respect to a single group operation. Then instead of computing $Q = nD$ in $\mathcal{J}(\mathcal{C})(\mathbb{F}_q)$, where n is an integer and $D \in \mathcal{J}(\mathcal{C})(\mathbb{F}_q)$, we perform:

$$Q = \phi^{-1}\big(n\,\phi(D)\big) \tag{1}$$

so that the bulk of the computation is done in $\mathcal{J}\big(\tilde{\mathcal{C}}\big)(\mathbb{F}_q)$, or, since a picture is worth a thousand words, we note that the following diagram commutes

$$\begin{tikzcd}
\mathcal{J}(\mathcal{C})(\mathbb{F}_q) \arrow[r, "\text{multiplication by } n"] \arrow[d, "\phi"'] & \mathcal{J}(\mathcal{C})(\mathbb{F}_q) \\
\mathcal{J}\big(\tilde{\mathcal{C}}\big)(\mathbb{F}_q) \arrow[r, "\text{multiplication by } n"'] & \mathcal{J}\big(\tilde{\mathcal{C}}\big)(\mathbb{F}_q) \arrow[u, "\phi^{-1}"']
\end{tikzcd}$$

and we follow it along the longer path.

The countermeasure is effective if the representations of the images under ϕ of the curve coefficients and of the elements of $\mathcal{J}(\mathcal{C})(\mathbb{F}_q)$ are unpredictably different from those of their sources. This can be achieved by multiplying the quantities involved in a computation with randomly chosen numbers (but: see Subsection 2.3). We are going to show that, in the case of hyperelliptic curves, this can be done by a small number of elementary field operations.

We do not discuss the use of random field isomorphisms according to [16, § 4.2]. The treatment carries over with little or no changes, and the method is computationally heavy, considerably slowing down all field operations. It is not clear whether it can even be done on a smart card in the `ecc` case. In `hecc`, the ground field being smaller, it is possible that this countermeasure could be implemented. As there is a potential performance/security trade-off in even characteristic with curve randomisations (see § 2.1), especially in the genus 2 case, one might be tempted to reconsider the use of field isomorphisms: However, divisor randomisation (see § 2.2) makes them superfluous.

General curve isomorphisms. We now put in practice the idea just sketched. Let $g \geqslant 1$ be an integer, and $\mathbb{F}_q$ be a finite field. Let $\mathcal{C}, \tilde{\mathcal{C}}$ be two hyperelliptic curves of genus g defined by *Weierstrass equations*

$$\mathcal{C} \ : \ y^2 + h(x)y - f(x) = 0 \tag{2}$$

$$\tilde{\mathcal{C}} \ : \ y^2 + \tilde{h}(x)y - \tilde{f}(x) = 0 \tag{3}$$

over $\mathbb{F}_q$, where $f, \tilde{f}$ are monic polynomials of degree $2g+1$ in x and $h(x), \tilde{h}(x)$ are polynomials in x of degree at most g. $\mathcal{C}$ (and $\tilde{\mathcal{C}}$) has no singular affine points, *i.e.* there are no solutions $(x, y) \in \mathbb{F}_q \times \mathbb{F}_q$ which simultaneously satisfy the equation $y^2 + h(x)y - f(x) = 0$ and the partial derivative equations $2y + h(x) = 0$ and $h'(x)y - f'(x) = 0$. This is equivalent to saying that the discriminant of $4f + h^2$ does not vanish [29, Theorem 1.7]. Analogous conditions holds for $\tilde{\mathcal{C}}$. Denote by ∞ the non affine point in the projective completions of $\mathcal{C}$ and $\tilde{\mathcal{C}}$. All $\mathbb{F}_q$-isomorphisms of curves $\phi : \mathcal{C} \to \tilde{\mathcal{C}}$ are, by [29, Proposition 1.2], of the type

$$\phi \ : \ (x, y) \mapsto \big(s^{-2}x + b, s^{-(2g+1)}y + A(x)\big) \tag{4}$$

for some $s \in \mathbb{F}_q^{\times}$, $b \in \mathbb{F}_q$ and a polynomial $A(x) \in \mathbb{F}_q[x]$ of degree at most g. Upon substituting $s^{-2}x + b$ and $s^{-(2g+1)}y + A(x)$ in place of x and y in equation (3) and comparing with (2) we obtain

$$\begin{cases} h(x) = s^{2g+1}\Big(\tilde{h}\big(s^{-2}x + b\big) + 2\,A(x)\Big) \\ f(x) = s^{2(2g+1)}\Big(\tilde{f}\big(s^{-2}x + b\big) - A(x)^2 - \tilde{h}\big(s^{-2}x + b\big)A(x)\Big) \end{cases} \tag{5}$$

whose inversion is

$$\begin{cases} \tilde{h}(x) = s^{-(2g+1)}h(\hat{x}) - 2A(\hat{x}) \\ \tilde{f}(x) = s^{-2(2g+1)}f(\hat{x}) + s^{-(2g+1)}h(\hat{x})A(\hat{x}) - A(\hat{x})^2 \\ \qquad\qquad \text{where} \quad \hat{x} = s^2(x - b) \ . \end{cases} \tag{6}$$

Now ϕ is an isomorphism of $\mathcal{C}$ onto $\tilde{\mathcal{C}}$ and it induces an isomorphism (which we also call ϕ) of their Jacobians, which is also a group isomorphism. It is a well known fact that the Jacobian of a curve $\mathcal{C}$ is isomorphic to the ideal class group $\mathrm{Cl}^0(\mathcal{C})$, which is more suitable for direct computations, and for this reason we want to see see how ϕ operates on the elements of $\mathrm{Cl}^0(\mathcal{C})$.

D. Mumford [35] has introduced a representation of the elements of the latter group as polynomial pairs, for which D. Cantor [4] provided an explicit arithmetic algorithm. Any divisor can be written as $D = \sum_{P\in\mathcal{S}} m_P P - (\sum_{P\in\mathcal{S}} m_P)\infty$ for a finite subset of points $\mathcal{S}$ of $\mathcal{C}\big(\overline{\mathbb{F}}_q\big)$ called the *support* of D, the m_i being positive integers, and the *degree* of D is the integer $\deg(D) = \sum_{P\in\mathcal{S}} m_P$. Let D be the unique principal divisor of degree at most g in a given divisor class on $\mathcal{C}$. Then (the ideal class associated to) D is represented by a unique pair of polynomials $U(t), V(t) \in \mathbb{F}_q[t]$ with $g \geqslant \deg_t U > \deg_t V$, U monic and such that:

$$\begin{cases} U(t) = \prod_{P\in\mathcal{S}} (t - x_P)^{m_P} \\ V(x_P) = y_P \quad \text{for all } P \in \mathcal{S} \\ U(t) \ \text{ divides } \ V(t)^2 + V(t)h(t) - f(t) \ . \end{cases} \tag{7}$$

We say that the pair $[U(t), V(t)]$ represents the *reduced divisor* D. It is $\deg(D) = \deg(U)$.

It is clear that we want to find a pair of polynomials $\tilde{U}(t), \tilde{V}(t) \in \mathbb{F}_q[t]$ which satisfy similar conditions, but for the divisor $\phi(D) = \sum_{P\in\mathcal{S}} m_P \phi(P) - (\sum_{P\in\mathcal{S}} m_P)\infty$ in place of D. In other words, we must have:

$$\begin{tikzcd} D = \sum_{P\in\mathcal{S}} m_P P - \Big(\sum_{P\in\mathcal{S}} m_P\Big)\infty \arrow[r, "\phi"] \arrow[d, equal] & \sum_{P\in\mathcal{S}} m_P \phi(P) - \Big(\sum_{P\in\mathcal{S}} m_P\Big)\infty = \phi(D) \arrow[d, equal] \\ \big[U(t), V(t)\big] \arrow[r, "\phi"] & \big[\tilde{U}(t), \tilde{V}(t)\big] \end{tikzcd}$$

This is very straightforward to obtain. Clearly

$$\begin{aligned} \tilde{U}(t) &= \prod_{P\in\mathcal{S}} \big(t - x_{\phi(P)}\big)^{m_P} = \prod_{P\in\mathcal{S}} \big(t - s^{-2}x_P - b\big)^{m_P} \\ &= s^{-2\sum_{P\in\mathcal{S}} m_P} U\big(s^2(t-b)\big) = s^{-2\deg_t U} U\big(s^2(t-b)\big) \ . \end{aligned} \tag{8}$$

Then, $\tilde{V}$ must satisfy $\tilde{V}(x_{\phi(P)}) = y_{\phi(P)}$ for all $P \in \mathcal{S}$, in other words

$$\tilde{V}(s^{-2}x_P + b) = s^{-(2g+1)}y_P + A(x_P) = s^{-(2g+1)}V(x_P) + A(x_P)$$

i.e.

$$\tilde{V}(t) = s^{-(2g+1)}V\big(s^2(t-b)\big) + A\big(s^2(t-b)\big) \ . \tag{9}$$

Equations (8) and (9) give the correct $\tilde{U}(t)$ and $\tilde{V}(t)$. This follows from the uniqueness of the representation of reduced divisors: In fact $\tilde{U}(t)$ and $\tilde{V}(t)$ are defined over $\mathbb{F}_q$, $\deg \tilde{V} = \deg V < \deg U = \deg \tilde{U}$, and it is straightforward to verify that $\tilde{U}(t)$ divides $\tilde{V}(t)^2 + \tilde{V}(t)\tilde{h}(t) - \tilde{f}(t)$.

Odd characteristic. Here we consider the case where $\mathbb{F}_q$ is a finite field of odd characteristic. We assume that $h(x) = \tilde{h}(x) = 0$, since we can transform the equations by the variable change $y \mapsto y - h(x)/2$ and $y \mapsto y - \tilde{h}(x)/2$. The advantage in doing so is that Cantor's algorithm will run faster, and for the same reason explicit formulae for odd characteristic have only been developed under this assumption. Then the equations of $\mathcal{C}, \tilde{\mathcal{C}}$ are of the form

$$\mathcal{C} \;:\; y^2 - f(x) = 0 \tag{10}$$

$$\tilde{\mathcal{C}} \;:\; y^2 - \tilde{f}(x) = 0 \tag{11}$$

which imply, by (6), that $A(x) = 0$.

If, furthermore, $\operatorname{char} \mathbb{F}_q \nmid 2g+1$, we can assume that the second most significant coefficient of $f(x)$ (and of $\tilde{f}(x)$), *i.e.* the coefficient f_{2g} of x^{2g}, vanishes too, since we can perform the variable change $x \mapsto x - f_{2g}/(2g+1)$. In this case, moreover, by (6) it must be $b = 0$, so the isomorphism ϕ takes the simple form

$$\phi \;:\; (x, y) \mapsto \left(s^{-2}x, s^{-(2g+1)}y\right) \tag{12}$$

where $s \in \mathbb{F}_q^{\times}$. (For simplicity, we shall consider only isomorphisms of this kind, even if $\operatorname{char} \mathbb{F}_q \mid 2g+1$.) The formula for $\tilde{f}$ is

$$\tilde{f}(x) = s^{-2(2g+1)} f\left(s^2 x\right) \;.$$

This randomisation modifies all non-zero coefficients of the Weierstrass equation (that is, all those who are used in the computation) and of the two polynomials representing a reduced divisor (except for the leading coefficient of U, which must remain equal to 1), namely

$$\tilde{U}(t) = s^{-2\deg_t U} U\left(s^2 t\right), \quad \tilde{V}(t) = s^{-(2g+1)} V\left(s^2 t\right) \;.$$

Explicit description, an implementation trick. The method is very fast. First, we pick a random $s \in \mathbb{F}_q^{\times}$ and compute its multiplicative inverse. They are both needed: s^{-1} for ϕ and s for ϕ^{-1}. We make the computation of ϕ explicit. If

$$f(x) = x^{2g+1} + \sum_{i=0}^{2g-1} f_i x^i$$

then

$$\tilde{f}(x) = x^{2g+1} + \sum_{i=0}^{2g-1} s^{2i-2(2g+1)} f_i x^i \;.$$

For $U(t)$ and $V(t)$ in the general case it is

$$U(t) = t^g + \sum_{i=0}^{g-1} U_i t^i \quad \text{and} \quad V(t) = \sum_{i=0}^{g-1} V_i t^i$$

so that

$$\tilde{U}(t) = t^g + \sum_{i=0}^{g-1} s^{2i-2g} U_i t^i \quad \text{and} \quad \tilde{V}(t) = \sum_{i=0}^{g-1} s^{2i-(2g+1)} V_i t^i \ .$$

To apply ϕ to the equation of the curve and to the basis divisor $[U(t), V(t)]$ we proceed as follows: Assume we have already s and s^{-1}. We compute s^{-k} for $k = 2, 3, \ldots 2g+1$ and $k = 2(g+1), 2(g+2), \ldots, 2(2g+1)$. This requires $3g+1$ multiplications (some can be replaced with squarings). For even k we compute $\tilde{f}_{2g+1-k/2} = s^{-k} f_{2g+1-k/2}$ (if $k \neq 2$) and $\tilde{U}_{g-k/2} = s^{-k} U_{g-k/2}$ (with $k \leqslant 2g$). If k is odd and $\leqslant 2g+1$ we multiply $V_{g-(k-1)/2}$ by s^{-k} to obtain $\tilde{V}_{g-(k-1)/2}$. Computing $\tilde{f}, \tilde{U}$ and $\tilde{V}$ requires $4g$ multiplications, hence the total amount of operations required to apply ϕ is $7g+1$ multiplications. Computing ϕ^{-1} requires only $4g$ multiplications in $\mathbb{F}_q$, bringing the total to $11g+1$.

In the cases $g = 2$, resp. 3 this randomisation needs 23, resp. 34 field multiplications (and possibly one inversion), which compares favorably to the costs of one group addition: in the genus 2 case, according to T. Lange [24] one group addition requires 25 multiplications and 1 inversion, and in the genus 3 case J. Pelzl *et al.* [38] need 76 multiplications and 1 inversion.

We mention an implementation trick to save an inversion each time the device is used at the price of a multiplication. During the initialisation of the device, a set $(\kappa_i, \kappa_i^{-1})$ of randomly chosen elements of $\mathbb{F}_q^\times$ together with their inverses is stored in the E^{2}PROM. Before each cryptographic operation, two random indices $i \neq j$ are picked, and the i-th pair is replaced by $(\kappa_i \cdot \kappa_j, \kappa_i^{-1} \cdot \kappa_j^{-1})$. The result is used as the (s, s^{-1}) for the curve randomisation in the current session.

Partial conclusions. Curve randomisation in odd characteristic is a fast countermeasure. The total amount of operations required to apply this technique is either comparable with that of a single group operation or much smaller.

Even characteristic. The discussion in § 2.1 applies in particular to the case of even characteristic. Let $d = [\mathbb{F}_q : \mathbb{F}_2]$. Since in this case one must have $h(x)\tilde{h}(x) \neq 0$ in equations (2) and (3), it is clear that applying the isomorphisms in general will not be as efficient as in the odd characteristic case.

In place of the fully general isomorphisms (4) we assume $b = 0$ and $A(x) = 0$, and proceed as at the end of § 2.1. The isomorphisms of the form

$$\phi \ : \ (x, y) \mapsto \left(s^{-2}x, s^{-(2g+1)}y\right) \tag{12'}$$

for generic $s \in \mathbb{F}_{2^d} \setminus \mathbb{F}_2$ randomise the coefficients of the equation as follows

$$\begin{cases} \tilde{h}(x) = s^{-(2g+1)} h(s^2 x) \\ \tilde{f}(x) = s^{-2(2g+1)} f(s^2 x) \ . \end{cases} \tag{13}$$

As in § 2.1 we make this explicit: if

$$f(x) = x^{2g+1} + \sum_{i=0}^{2g-1} f_i x^i \quad \text{and} \quad h(x) = \sum_{i=0}^{g} h_i x^i$$

then

$$\tilde{f}(x) = x^{2g+1} + \sum_{i=0}^{2g-1} s^{2i-2(2g+1)} f_i x^i \quad \text{and} \quad \tilde{h}(x) = \sum_{i=0}^{g} s^{2i-(2g+1)} h_i x^i$$

and the formulae for $\tilde{U}, \tilde{V}$ are the same as in § 2.1. All the coefficients of the equation and of the divisor are then multiplied by random constants. In even characteristic we must compute also the coefficients of $\tilde{h}(x)$ from those of $h(x)$. Hence, at most $g+1$ field operations more are required than in the odd characteristic case, bringing the cost of the computation of ϕ to at most $8g+2$ multiplications, after s has been randomly chosen and s^{-1} computed. The computation of ϕ^{-1} still requires $4g$ multiplications. The total cost of this randomisation is thus $12g+2$ field multiplications and one inversion: The implementation trick described in § 2.1 not necessary in even characteristic, inversion being much faster in this case.

Restricting h: h constant. In even characteristic often the coefficients of $h(x)$ are restricted for performance reasons. In this paragraph we consider the case where $h(x)$ is a non-zero constant. Equation (6) implies that $\tilde{h}(x)$ will also be a non-zero constant.

It is an established fact in algebraic geometry that curves of equation $y^2 + cy = f(x)$ with $\deg f = 5$ and $c \neq 0$ are supersingular [11, Theorem 9] and so are not suitable for the cryptographic applications we have in mind.

On the other hand there are no hyperelliptic supersingular curves of genus 3 in characteristic 2 [39], so curves of the form $y^2 + cy = f(x)$ where $\deg f = 7$ and $c \neq 0$ do not appear to be weak provided that parameters as extension degree and group order are suitably chosen. Now, even though in [38] a very fast doubling formula is given for the doubling in the case $h(x) = 1$, J. Pelzl has privately communicated to us that in the generic case where $h(x)$ is a non-zero constant $h(x) = c \in \mathbb{F}_{2^d}$ doubling speed can still be improved dramatically. Trivially, $\tilde{h}(x) = s^{-(2g+1)} c = s^{-7} c$. This makes the genus 3 case important.

Restricting h: h non-constant but defined over $\mathbb{F}_2$. Another technique for gaining performance is to choose $h(x)$ non-constant but defined over $\mathbb{F}_2$ (see for example [25] and [26]). By (6) this leads to the question: *if* $h(x) \in \mathbb{F}_2[x]$, *for which elements* $b \in \mathbb{F}_q$ *and* $s \in \mathbb{F}_q^\times$ *is it* $\tilde{h}(x) = s^{-(2g+1)} h\big(s^2(x-b)\big) \in \mathbb{F}_2[x]$*?*

The leading coefficient of $\tilde{h}(x)$ equals s^{-r} where $r = (2g+1) - 2\deg h$, and since it cannot vanish, it is 1, *i.e.* $s^r = 1$. Now r is an odd positive integer bounded by $2g-1$. The cryptosystem must withstand P. Gaudry's low genus algorithm for computing discrete logarithms in hyperelliptic Jacobians [12]. Hence g must be small, in fact $g \leqslant 4$, so $r \leqslant 7$. This implies that s can take only very few possible values, making superfluous the effort of randomising it.

Remark: *In order to make Weil Descent attacks [9,10] infeasible, the extension degree d is usually taken to be a large prime number $p \gtrapprox 160/g$ or twice a prime $p \gtrapprox 80/g$. Recall also that $g \leqslant 4$. The possible values of s are limited to the roots of irreducible factors of $X^r - 1$ of degree dividing d. If $d = p \gtrapprox 160/g \geqslant 40$, which is also the preferred case, then $s = 1$. If $d = 2p$ with $p \gtrapprox 80/g \geqslant 20$, s can only be a root of a factor of $X^r - 1$ of degree at most 2 and irreducible over $\mathbb{F}_2$. A quick verification of such factors (recall that r is odd and $\leqslant 7$) implies that either $s = 1$ or $r = 3$ and $s^2 + s + 1 = 0$. If two coefficients of $h(x)$ are equal to 1, forcing the corresponding coefficients of $\tilde{h}(x)$ to be also equal to 1 implies always $s = 1$.*

Let σ be the Frobenius automorphism of $\mathbb{F}_q/\mathbb{F}_2$, *i.e.* $\alpha \mapsto \alpha^2$. Now $\tilde{h}(x) = h(x-b) \in \mathbb{F}_2[x]$, hence $h(-b^{\sigma^j}) = h(-b)^{\sigma^j} = h(-b) \in \mathbb{F}_2$ for all j. In other words all distinct conjugates of $-b$ are roots of $h(x) - h(-b) = 0$, and if $b \not\in \mathbb{F}_2$ there are at least $p \gtrapprox 80/g \geqslant 20$ of such conjugates, including $-b$. But the degree of h, as we already know, is at most $g \leqslant 4$, and this forces $b \in \mathbb{F}_2$. There are only two choices for b, making useless to consider its randomisation.

We see that the isomorphisms we can use are of the form

$$\phi \ : \ (x, y) \mapsto \big(x, y + A(x)\big)$$

where the polynomial $A(x) \in \mathbb{F}_q[x]$ has degree at most g. The situation is similar to that for elliptic curves as described in the already cited paper of M. Joye and C. Tymen: we can efficiently randomise only one of the two polynomials (V, whereas U will be left untouched), or, in other words, only a half of the coordinates. In fact, by (6) not all coefficients of f are randomised in $\tilde{f}$, increasing the likelihood of successful bit-correlations if this countermeasure is used alone.

Partial conclusions. We conclude that for genus 2 hyperelliptic curves in characteristic 2, curve randomisation is not adequate if one wants to force the coefficients of $\tilde{h}$ to lie outside $\mathbb{F}_2$.

In the genus 3 case curves of equation $y^2 + cy = f(x)$ can be randomised obtaining good performance and security.

In all other cases, we recommend other techniques, such as divisor randomisation, which also works in odd characteristic. We sketch it in the next section in the case of genus 2.

2.2 Divisor Randomisation in Genus 2

Divisor randomisation works by randomising the bits of the representation of a reduced divisor, which can be either the base group element of the cryptosystem or any intermediate result of the computation of a scalar multiplication. This technique does not scramble the bits of the internal representations of the coefficients of the curve. It can be used whenever a group element can be represented in several different ways. Notable examples are the projective coordinates on elliptic curves: two triples (X, Y, Z) and (X', Y', Z') represent the same point if and only if there exists a non zero element s in the base field such that $X = sX'$, $Y = sY'$ and $Z = sZ'$. With Jacobian coordinates [5], two triples (X, Y, Z) and (X', Y', X') represent the same point if and only if $X = s^2X'$, $Y = s^3Y'$ and $Z = sZ'$.

Recently, alternative coordinate systems for genus 2 hyperelliptic curves have been proposed: An inversion-free system by Miyamoto *et al.* [32] which operates on the hyperelliptic analogue of projective coordinates, later extended and improved by Lange [25], who also developed an analogue of Jacobian coordinates, called the *new (or weighted) coordinates* [26]. We are not aware of similar coordinate systems for genus 3 curves. Furthermore, as the genus of the considered curve increases, the size of the base field decreases, and the cost of a field inversion relative to a field multiplication also decreases quickly. This makes inversion-free formulae in genus at least 3, not so desirable from the point of view of raw performance, because they trade a single inversion for a lot more multiplications than the affine formulae.

In projective coordinates a divisor class D with associated reduced polynomial pair $[U(t), V(t)]$ is represented as a quintuple $[U_1, U_0, V_1, V_0, Z]$ where

$$U(t) = t^2 + \frac{U_1}{Z}\,t + \frac{U_0}{Z} \quad \text{and} \quad V(t) = \frac{V_1}{Z}\,t + \frac{V_0}{Z} \ .$$

The randomisation in this case consists in picking a random $s \in \mathbb{F}_q^\times$ and by performing the following replacement

$$[U_1, U_0, V_1, V_0, Z] \mapsto [sU_1, sU_0, sV_1, sV_0, sZ] \ .$$

In new (weighted) coordinates a divisor class is represented by means of six coordinates $[U_1, U_0, V_1, V_0, Z_1, Z_2]$ where

$$U(t) = t^2 + \frac{U_1}{Z_1^2}\,t + \frac{U_0}{Z_1^2} \quad \text{and} \quad V(t) = \frac{V_1}{Z_1^3 Z_2}\,t + \frac{V_0}{Z_1^3 Z_2} \ .$$

To blind the base point or an intermediate computation, two elements s_1, s_2 are picked in $\mathbb{F}_q^\times$ at random and the following substitution is performed:

$$[U_1, U_0, V_1, V_0, Z_1, Z_2] \mapsto [s_1^2U_1, s_1^2U_0, s_1^3s_2V_1, s_1^3s_2V_0, s_1Z_1, s_2Z_2] \ .$$

If (some or all of) the additional coordinates z_1, z_2, z_3 and z_4 are used – which satisfy $z_1 = Z_1^2$, $z_2 = Z_2^2$, $z_3 = Z_1Z_2$ and $z_4 = z_1z_2$ – then they must also be updated: the fastest way is to recompute them from Z_1 and Z_2 by two squarings and two multiplications.

2.3 Goubin's Attack May Force Further Blinding

Recently L. Goubin [13] has pointed out a potential weakness of some ecc randomisation procedures, including Coron's third and Joye-Tymen's, when implemented on systems where the secret scalar is fixed and the base of the scalar multiplication (the message) can be chosen. Since our techniques generalise the above ones, it is natural to investigate how Goubin's ideas might affect our work.

His attack is based on the randomisation of 0 by multiplication by a constant or by field isomorphism being still 0. It relies also on the fact that the scalar multiplication algorithm has a fixed sequence of group operations for a given scalar – even after removing any dummy operations. (It should work also if the number of possible operations sequences for a given scalar is small enough.)

Suppose that the most significant bits $n_r, n_{r-1}, \dots, n_{j+1}$ of the secret scalar n are known and that we want to discover the next bit n_j. Assume also that a *chosen message attack* can be set up to obtain in a specific step of the scalar multiplication – namely the one corresponding to the processing of n_j – a point or a divisor with one or more coordinates equal to zero, provided that n_j has been guessed correctly (that divisor can be tD where D is the "message" and $t = (n_r, n_{r-1}, \dots, n_{j+1}, n_j)_2$). The side-channel trace correlation may reveal if the guess was correct even in presence of multiplicative randomisation procedures, because some multiplications by zero will occur in any case. In particular, this can affect the random isomorphisms of the form $\phi : (x, y) \mapsto \left(s^{-2}x, s^{-(2g+1)}y\right)$ and the divisor randomisation techniques of Subsection 2.2.

An approach to thwart Goubin's attack could use the more general isomorphisms (4) with b, $A(x) \neq 0$ to randomise also the vanishing coefficients of the divisors: this has the disadvantage of requiring curve equations in general form and thus slower formulae for the operations.

There is a development of Goubin's ideas which might be even more serious. We first fix some notation: Λ is the large prime order subgroup of $\mathrm{Cl}^0(\mathcal{C})(\mathbb{F}_q)$ used in the cryptosystem and ℓ its order.

A variant of Goubin's attack may exploit the fact that the basic explicit formulae for small genus hyperelliptic curves only deal with the most common cases (cfr. [24,27,38]). They do not hold if the divisors given as input to a group operation satisfy exceptional conditions, such as:

(i) If the reduced divisor D_i, for $i = 1, 2$, is represented by the polynomial pair $[U_i, V_i]$, then the greatest common divisor of U_1 and U_2 is non-constant or, equivalently, their resultant is vanishing.
In this case we say that D_1 and D_2 *collide*. This happens if the supports of D_1 and of either D_2 or $-D_2$ have at least one point in common.

(ii) $\deg(D_1) < g$ or, equivalently, $\deg(U_1) < g$ (this applies to additions as well as to doublings).

Such situations occur in practice with very small probability ($O(q^{-1})$ for curves over $\mathbb{F}_q$), hence no separate formulae for these cases are implemented and either

Cantor's algorithm or formulae with quite different characteristics are used[1]. Since the characterising properties of these divisors are of geometric nature, they are preserved under curve isomorphisms. Their occurrence may thus be induced at prescribed points to verify the guesses of the bits of the scalar. At least in theory, the attacker guesses that at some point the scalar multiplication algorithm adds D to tD (resp. doubles tD) and therefore chooses D to collide with tD (resp. $\deg(tD) < g$). We do not know, except for very simple cases (i.e. $|t| \leqslant g$), how to produce for a given t (with $\ell \nmid t$) a divisor D colliding with tD (reduced) – we suspect that it is in general a hard problem. On the other hand it is very easy to find D such that $\deg(tD) < g$: just pick any $D' \in \mathit{\Lambda}$ with $\deg(D') < g$, find an integer s with $s\,t \equiv 1 \pmod{\ell}$ and put $D = sD'$. Then $tD = D'$ and, if the doubling formula for the exceptional case is distinguishable from the generic one, the attack can be launched.

This represents an obvious danger with affine coordinates: if one of the above exceptional conditions occurs, the most common case formulae cannot be used, to avoid a division by zero. With inversion-free coordinate systems the situation is only apparently different: one can just use only the most common case formulae and check at the end of the scalar multiplication if the divisor belongs to the curve – but also in that case anomalous behaviour of the device at the end of the scalar multiplication could be detected.

We therefore need additional scalar and message blinding methods.

We briefly discuss scalar blinding methods. Their purpose is to render unpredictable the addition chain used in the scalar multiplication, thus preventing the attacker to guess for which integers t group operations of the type $D + tD$ or $2(tD)$ are actually performed.

The first is Coron's first countermeasure [7], *i.e.* the replacement of the scalar n with $n+k\ell$ in nD for a random integer k. This technique can be traced back to [20], and was shown [36] to leave a bias in the least significant bits of the scalar. B. Möller [34] combines it (only in the `ecc` case) with an idea of C. Clavier and M. Joye, and suggests the computation of $nD = (n + k_1 + k_2\ell)D - k_1D$, where k_1 and k_2 are two suitably sized random numbers: k_1 and k_2 should be large enough to make L. Goubin's attack not palatable, yet not too big, to leave the overhead tolerable (for example $k_1, k_2 \approx 2^{32}$ are good choices if $\ell \approx 2^{160}$).

For a completely different technique see [41].

For message blinding, a `hecc` analogue of Coron's second method [7] consists in replacing the computation of nD with that of $n(D + R) - S$, where $R \in \mathit{\Lambda}$ is a secret divisor for which $S = nR$ is known. A set of secret divisor pairs $(R_i, S_i) \in \mathit{\Lambda} \times \mathit{\Lambda}$ with $S_i = nR_i$ can be stored in the smart card at initialisation time, and at each run both elements of a randomly chosen pair are multiplied by the same small signed scalar and added to the respective elements of another pair. The result is then used to randomise the scalar multiplication.

[1] To provide explicit *and* indistinguishable formulae for all cases would be a formidable feat – and would probably slow down considerably the cryptosystem.

Suppose that a computation involving tD has to be done during the scalar multiplication, either $D+tD$ or $2(tD)$, and that either D collides with tD (this is relevant only to the addition $D+tD$) or $\deg(tD) < g$ – as wished by the attacker. If D has been replaced at the beginning by $D+R$ for a randomly chosen point R, then $D+R$ and $t(D+R)$ will collide with probability $O(q^{-1})$ (this is actually a conjecture which has been extensively confirmed experimentally on small curves), resp. $\deg(t(D+R)) = g$ also with probability $O(q^{-1})$: The last statement holds because $t(D+R)$ is in practice a random point, which implies also that, even if tD had some zero coordinates, a fixed coordinate of $t(D+R)$ would be zero with probability q^{-1}.

We infer that this type of message blinding (which, if used alone, might arouse suspicion) thwarts Goubin's attack. Due to a similar underlying philosophy, additional message blinding should be effective also against some `hecc` analogue of the "exceptional procedure attack" [14].

***To prevent a variant of Goubin's attack**, we recommend to use at least an additional scalar or message blinding method besides our randomisation procedures. The hyperelliptic analogue of Coron's second countermeasure, being less expensive than scalar blinding, looks particularly attractive. The isomorphism ϕ need not be of the most general type described in 2.1, but the conclusions of 2.1 and the caveats of 2.1 still apply.*

3 Conclusions

We proposed two methods to blind the base divisor class for hyperelliptic curve cryptosystems, in order to provide resistance against DPA.

The first method consists in transferring the critical computation to the Jacobian of a different randomly chosen isomorphic curve. It can be applied to curves of all genera.

The second method is a hyperelliptic analogue of Coron's third countermeasure. It applies only to families of curves for which we know explicit formulae for hyperelliptic analogues of elliptic curve projective and Jacobian coordinates. Explicit examples in the case of genus 2 have been worked out in detail.

These techniques are easy to implement and do not impact the performance significantly. In fact their cost is at most that of a single group addition.

In conjunction with suitable additional scalar and message blinding techniques, they can be made resistant against Goubin's recent chosen message attack, as well as against a possibly more serious variant of the latter based on the structure of the divisors on hyperelliptic curves.

Acknowledgement. The author wishes to express his gratitude to Tanja Lange, Kim Nguyen, Bertrand Byramjee and the anonymous referees for their valuable remarks and suggestions. The author thanks also Alessandro Baldaccini for partially reviewing his English.

References

1. G.B. Agnew, R.C. Mullin and S.A. Vanstone, *An Implementation of Elliptic Curve Cryptosystems over $F_{2^{155}}$.* IEEE Journal on Selected Areas in Communications, vol. 11, n. 5, pp. 804–813, 1993.
2. A. Bellezza, *Countermeasures against Side-Channel Attacks for Elliptic Curve Cryptosystems.* Cryptology ePrint Archive, Report 2001/103.
   Available from: `http://eprint.iacr.org/`
3. E. Brier and M. Joye, *Weierstrass Elliptic Curves and Side-Channel Attacks.* In: *Proceedings of PKC'2002*, LNCS 2274, pp. 335–345. Springer-Verlag, 2002.
4. D. Cantor, *Computing in the jacobian of a hyperelliptic curve.* Mathematics of Computation, **48** (1987), pp. 95–101.
5. D.V. Chudnovsky and G.V. Chudnovsky, *Sequences of numbers generated by addition in formal groups and new primality and factoring tests*, Advances in Applied Mathematics, **7** (1987), pp. 385–434.
6. C. Clavier and M. Joye, *Universal exponentiation algorithm - a first step towards provable SPA-resistance.* In: *Proceedings of CHES 2001*, LNCS 2162, pp. 300–308. Springer-Verlag, 2000.
7. J.-S. Coron, *Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems.* In: *Proceedings of CHES '99*, LNCS 1717, pp. 292–302. Springer-Verlag, 1999.
8. W. Fischer, C. Giraud, E.W. Knudsen and J.-P. Seifert, *Parallel Scalar Multiplication on General Elliptic Curves over F_p hedged against Differential Side Channel Attacks.* Cryptology ePrint Archive, Report 2002/007, 2002.
   Available from: `http://eprint.iacr.org/`
9. G. Frey, *How to disguise an elliptic curve (Weil descent).* Talk at ECC '98, Waterloo, 1998. Slides available from:
   `http://www.cacr.math.uwaterloo.ca/conferences/1998/ecc98/slides.html`
10. G. Frey, *Applications of arithmetical geometry to cryptographic constructions.* In: *Finite fields and applications (Augsburg, 1999)*, pp. 128–161. Springer-Verlag, 2001.
11. S.D. Galbraith, *Supersingular curves in cryptography.* In: *Proceedings of ASIACRYPT 2001*, LNCS 2248, pp. 495–513. Springer-Verlag 2001.
12. P. Gaudry, *An algorithm for solving the discrete log problem on hyperelliptic curves.* In: *Advances in Cryptology – Eurocrypt 2000*, LNCS 1807, pp. 19–34. Springer-Verlag, 2000.
13. L. Goubin, *A Refined Power-Analysis Attack on Elliptic Curve Cryptosystems.* In: *Proceedings of PKC 2003*, LNCS 2567, pp. 199–211. Springer-Verlag, 2003.
14. T. Izu and T. Takagi, *Exceptional Procedure Attack on Elliptic Curve Cryptosystems.* In: *Proceedings of PKC 2003*, LNCS 2567, pp. 224–239. Springer-Verlag, 2003.
15. M. Joye, J.-J. Quisquater, *Hessian Elliptic Curves and Side-Channel Attacks.* In: *Proceedings of CHES'2001*, LNCS 2162, pp. 412–420, Springer–Verlag, 2001
16. M. Joye and C. Tymen, *Protections against Differential Analysis for Elliptic Curve Cryptography – An Algebraic Approach.* In: *Proceedings of CHES 2001*, LNCS 2162, pp. 377–390. Springer-Verlag, 2001.
17. N. Koblitz, *A family of Jacobians suitable for discrete log cryptosystems.* In: *Advances in Cryptology – Proceedings of CRYPTO '88*, LNCS 403, pp. 94-99. Springer-Verlag, 1990.

18. N. Koblitz, *Hyperelliptic Cryptosystems*, Journal of Cryptology **1** (1989), pp. 139–150.
19. N. Koblitz, *Algebraic aspects of cryptography.* Springer, 1998.
20. P. Kocher, *Timing attacks on implementations of Diffie-Hellman, RSA, DSS and other systems.* In: *Advances in Cryptology, Proceedings of Crypto'96*, LNCS 1109, pp. 104–113, Springer-Verlag, 1996.
21. P. Kocher, J. Jaffe and B. Jun, *Introduction to Differential Power Analysis and Related Attacks,* 1998.
    Available from: `http://www.cryptography.com/dpa/technical`
22. P. Kocher, J. Jaffe and B. Jun, *Differential Power Analysis.* In: *Proceedings of CRYPTO'99*, LNCS 1666, pp. 388–397. Springer-Verlag, 1999.
23. J. Kuroki, M. Gonda, K. Matsuo, J. Chao and S. Tsujii, *Fast Genus Three Hyperelliptic Curve Cryptosystems.* In: *The 2002 Symposium on Cryptography and Information Security, Japan - SCIS 2002*, Jan. 29–Feb. 1 2002.
24. T. Lange, *Efficient Arithmetic on Genus 2 Hyperelliptic Curves over Finite Fields via Explicit Formulae.* Cryptology ePrint Archive, Report 2002/121. Available from: `http://eprint.iacr.org/` – See also [27].
25. T. Lange, *Inversion-Free Arithmetic on Genus 2 Hyperelliptic Curves.* Cryptology ePrint Archive, Report 2002/147. Available from: `http://eprint.iacr.org/` – See also [27].
26. T. Lange, *Weighted Coordinates on Genus 2 Hyperelliptic Curves.* Cryptology ePrint Archive, Report 2002/153. Available from: `http://eprint.iacr.org/` – See also [27].
27. T. Lange, *Formulae for Arithmetic on Genus 2 Hyperelliptic Curves.* Preprint. Available from: `http://www.ruhr-uni-bochum.de/itsc/tanja/`
    It partially contains and extends the material of the previous three papers [24, 25,26].
28. P.-Y. Liardet, N.P. Smart, *Preventing SPA/DPA in ECC system using the Jacobi Form.* In: *Proceedings of CHES'2001*, LNCS 2162, pp. 401–411. Springer-Verlag, 2001.
29. P. Lockhart, *On the discriminant of a hyperelliptic curve.* Trans. Amer. Math. Soc. **342** (1994), no. 2, pp. 729–752.
30. J. López and R. Dahab, *Fast Multiplication on Elliptic Curves over* $GF(2^m)$ *without Precomputation.* In: *Proceedings of CHES'99*, LNCS 1717, pp. 316–327. Springer-Verlag, 1999.
31. A. Menezes, Y.-H. Wu and R. Zuccherato, *An Elementary Introduction to Hyperelliptic Curves.* In [19].
32. Y. Miyamoto, H. Doi, K. Matsuo, J. Chao and S. Tsuji, *A fast addition algorithm of genus two hyperelliptic curve.* In: *Proceedings of SCIS 2002*, IEICE Japan, pp. 497–502, 2002. In Japanese.
33. P.L. Montgomery, *Speeding the Pollard and Elliptic Curve Methods for Factorizations.* Mathematics of Computation, vol. 48, pp. 243–264, 1987.
34. B. Möller, *Securing Elliptic Curve Point Multiplication against Side-Channel Attacks.* In: *Proceedings of ISC '2001*, LNCS 2200, pp. 324–334. Springer-Verlag, 2001.
35. D. Mumford, *Tata Lectures on Theta II.* Birkhäuser 1984.
36. K. Okeya and K. Sakurai, *Power analysis breaks elliptic curve cryptosystems even secure against the timing attack.* In: *Progress in Cryptology - INDOCRYPT 2000*, LNCS 1977, pp. 178–190. Springer-Verlag, 2000.

37. K. Okeya, H. Kurumatani and K. Sakurai, *Elliptic curves with the Montgomery–form and their cryptographic applications.* In: *Public Key Cryptography PKC 2000*, LNCS 1751, pp. 238–257. Springer-Verlag, 2000.
38. J. Pelzl, T. Wollinger, J. Guajardo and C. Paar, *Hyperelliptic Curve Cryptosystems: Closing the Performance Gap to Elliptic Curves.* This Volume.
39. J. Scholten and H.J. Zhu, *Hyperelliptic curves in characteristic 2.* Inter. Math. Research Notices, **17** (2002), pp. 905–917.
40. N.P. Smart, *The Hessian Form of an Elliptic Curve.* In: *Proceedings of CHES '2001*, LNCS 2162, pp. 118–125, Springer-Verlag, 2001.
41. C.D. Walter, *MIST: An Efficient, Randomized Exponentiation Algorithm for Resisting Power Analysis.* In: *Topics in Cryptology – CT-RSA 2002, The Cryptographer's Track at the RSA Conference, 2002, San Jose, CA, USA, February 18-22, 2002, Proceedings*, LNCS 2271, pp. 53–66. Springer-Verlag, 2002.

Appendix: Explicit Transformations for the Curve Randomisation for Genus 2

As an example, in this appendix we write down the transformations for the curve randomisation method explained in Subsection 2.1, for $g = 2$. In view of the results of § 2.1, we consider here only the curve isomorphism of type (12) where the equations of $\mathcal{C}$ and $\tilde{\mathcal{C}}$ are given by

$$\mathcal{C} \ : \ y^2 + h(x)y - f(x) = 0 \quad \text{and} \quad \tilde{\mathcal{C}} \ : \ y^2 + \tilde{h}(x)y - \tilde{f}(x) = 0 \ .$$

The polynomials $f(x)$ and $h(x)$ are of the form

$$\begin{aligned} f(x) &= x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0 \quad \text{and} \\ h(x) &= h_2x^2 + h_1x + h_0 \ . \end{aligned}$$

Their images are

$$\begin{aligned} \tilde{f}(x) &= x^5 + s^{-2}f_4x^4 + s^{-4}f_3x^3 + s^{-6}f_2x^2 + s^{-8}f_1x + s^{-10}f_0 \quad \text{and} \\ \tilde{h}(x) &= s^{-1}h_2x^2 + s^{-3}h_1x + s^{-5}h_0 \ . \end{aligned}$$

If the base divisor is given by $D = [U(t), V(t)]$ with $\deg(U) = 2$,

$$U(t) = t^2 + U_1t + U_0 \quad \text{and} \quad V(t) = V_1t + V_0$$

then its image $[\tilde{U}(t), \tilde{V}(t)]$ is

$$\tilde{U}(t) = t^2 + s^{-2}U_1t + s^{-4}U_0 \quad \text{and} \quad \tilde{V}(t) = s^{-3}V_1t + s^{-5}V_0 \ .$$

If $\deg(U) = 1$, *i.e.* $U(t) = t + U_0$, then its image is $\tilde{U}(t) = t + s^{-2}U_0$, whereas the image of $V(t)$ is independent of the degree: in this case $V(t) = V_0$ and thus $\tilde{V}(t) = s^{-5}V_0$. The inverse transformation from $\phi(D)$ to D is obvious.

The total number of field operations is at most 26 multiplications in the even characteristic case, 23 multiplications in odd characteristic (because $h = 0$), and one inversion (but: see end of 2.1).