Efficient Explicit Formulae for Genus 3 Hyperelliptic Curve Cryptosystems

Xinxin Fan, Thomas Wollinger, and Guang Gong, *Member, IEEE*

Abstract— **The ideal class groups of hyperelliptic curves (HECs) can be used in cryptosystems based on the discrete logarithm problem. Recent developments of computational technologies for scalar multiplications of divisor classes have shown that the performance of hyperelliptic curve cryptosystems (HECC) is compatible to that of elliptic curve cryptosystems (ECC). Especially, genus 3 HECC are well suited for all kinds of embedded processor architectures, where resources such as storage, time or power are constrained, because of their short operand sizes. In this paper, we investigate the efficient explicit formulae for genus 3 HECs over both prime fields and binary fields, and analyze how many field operations are needed. First, we improve the explicit formulae for genus 3 HECs over binary fields using the theta divisors which can save about $20\% \sim 50\%$ multiplications for four cases, and extend the method to genus 3 HECs over prime fields. We then discuss acceleration of the divisor class doubling for genus 3 HECs over binary fields. By constructing birational transformations of variables, we find four types of curves which can lead to much faster divisor class doubling and give the corresponding explicit formulae. Especially, for special genus 3 HECs over binary fields with $h(X) = 1$, we obtain the fastest explicit doubling formula which only requires $1I + 10M + 11S$. Thirdly, we propose the inversion-free explicit formulae for genus 3 HEC over both prime fields and binary fields by introducing one more coordinate to collect the common denominator of the usual six coordinates. Finally, comparisons with the known results in terms of field operations and an implementation of genus 3 HECC over three binary fields on a Pentium-4 processor are provided.**

Index Terms— **Genus 3 hyperelliptic curves, explicit formulae, Cantor's algorithm, Harley's algorithm, theta divisors, inversion-free, efficient implementation.**

I. Introduction

PUBLIC-key cryptography was introduced in 1976 by Diffie and Hellman [21]. The first practical realization followed in 1977 when Rivest, Shamir and Adleman proposed their now the most widely used RSA cryptosystem [94], in which security is based on the intractability of the integer factorization problem. Elliptic curve cryptography, first proposed in the work of Koblitz [60] and Miller [81], has received dramatically great attention in the past almost 20 years. The motivation is that there is no known sub-exponential algorithm to solve the discrete logarithm problem on a general elliptic curve. This means that a desired security level can be attained with significantly smaller keys in elliptic curve cryptosystems than those in their RSA counterparts. Furthermore, all the standard protocols in cryptography which make use of the discrete logarithm problem in finite fields, such as Diffie-Hellman key exchange, ElGamal encryption and digital signature [25] and Digital Signature Algorithm (DSA) [32], have their analogues in the elliptic curve case.

In 1988, Koblitz proposed for the first time to use the Jacobian of a hyperelliptic curve defined over a finite field to implement cryptographic protocols based on the discrete logarithm problem [61]. Due to the work of Cantor [13] (for odd characteristic only) and Koblitz [62] (for a generalization to even characteristic), it is possible to perform efficient operations in the ideal class group of a hyperelliptic curve (Cantor's Algorithm). The field size of HECC is about $1/g$-th that of ECC where g is the genus of the underlying curves, because of the algebraic structure of HECs. For example, we can construct genus 3 HECC on 56-bit finite fields in order to achieve the same security level as 160-bit ECC. Thus genus 3 HECC can be implemented efficiently on a 64-bit CPU without any multi-precision arithmetic. Therefore, HECC are attractive from the implementation point of view. However, the group operations of HECC are more complex than those of ECC. Therefore, how to reduce the computational complexity of HECC group operations has recently become an active research topic in both academia and industry communities.

The first attempt to find efficient algorithms for group operations of HECC was done by Spallek [100] and Krieger [63]. The first practical formulae for genus 2 HECs, which are called Harley's Algorithm or Explicit Formulae, were obtained by Harley [39], [47]. The Harley's algorithm is an explicit representation of the Cantor's algorithm [13], [62] for genus 2 HEC defined over prime fields. Since then, tremendous effort has been made to extend and optimize the Harley's algorithm in order to make the performance of the HECC compatible to that of the ECC. The performance of HECC has been analyzed and implemented in all kinds of general-purpose processors and embedded processors [2]–[4], [6], [12], [22], [28]–[31], [39], [42]–[45], [47], [53]–[55], [57], [58], [64]–[72], [74], [77], [82], [84], [86], [88]–[92], [95]–[97], [99], [101]–[103], [108], [111], [112], and in many hardware platforms such as Field Programmable Gate Arrays (FPGAs) [7]–[9], [11], [17], [26], [50], [51], [56], [107], [109], [110]. Furthermore, using HECs to efficiently implement pairing-based cryptosystem has actively investigated recently [5], [14]–[16], [23], [24], [46], [48], [75]. However, most of those improvements focus on genus 2 HECC. In this paper, we accelerate the group operations for genus 3 HECC from three

Xinxin Fan and Guang Gong are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: x5fan@engmail.uwaterloo.ca, ggong@ece.uwaterloo.ca).

Thomas Wollinger is with the Department of Electrical Engineering and Information Sciences, Communication Security Group (COSI), Ruhr-Universitaet Bochum, Germany, Universitaetsstrasse 150, 44780 Bochum, Germany, (e-mail: wollinger@crypto.rub.de)

aspects: using the Theta divisors, optimizing explicit doubling formulae for genus 3 curves over binary field using birational transformations and employing inversion-free arithmetic.

Our Main Contributions[1]

In [54] and [55], Katagi *et al.* introduced the concept of the theta divisors (or degenerate divisors) and explained how to use the theta divisors positively for speeding up the scalar multiplication of the HECC, saving the memory space in storing the base divisor, and thwarting the side-channel attacks. They showed explicit formulae related to the theta divisors of genus 3 HEC defined over binary fields for five cases. In this paper, we present optimizations for their formulae and generalize those techniques to prime fields. Our improvements save about $20\% \sim 50\%$ multiplications compared to Katagi *et al.*'s formulae in four cases. All of these formulae are useful not only for genus 3 HECC but also for pairing-based cryptosystem using genus 3 HECs.

Next, we accelerate the divisor doubling for genus 3 HECC over binary fields using special types of curves. We generalize Tanja *et al.*'s idea to the genus 3 case and improve the results in [45]. By constructing birational transformations of variables, we derive explicit doubling formulae for four types of special curves. For each type of curves, we analyze how many field operations are needed. So far no attack on any of all the curves suggested in this paper is known, even though some cases are very special. Depending on the degree of h, our explicit formulae only require $1I+10M+11S$, $1I+13M+13S$, $1I+20M+12S$ and $1I+26M+11S$ for divisor class doublings in the best case, respectively. Especially, for the case of $h(X)=1$, we obtain the fastest explicit doubling formula, and for the case of deg $h=1$, our explicit formula improve the recent results in [45] significantly by saving $31M$ at the cost of extra $7S$. In addition, we discuss some cases which are not included in [45]. Our results allow to choose curves from a large variety, which have extremely fast doubling with needing only one-third of the time of an addition in the best case.

Thirdly, for some application environments of HECC, such as smart cards, Personal Digital Assistants (PDAs) and so on, where inversions are extremely time and space critical, we consider the projective coordinates for avoiding inversions at the cost of more multiplications and one more coordinate for genus 3 HECC. When genus 3 HECs are defined over a prime field $\mathbb{F}_p$, our inversion-free explicit formulae will cost respectively $123M+7S$, $104M+6S$, $107M+10S$ and $86M+6S$ for performing a group addition, mixed addition, doubling and affine doubling. If we use special genus 3 HECs with $h(x)=1$ over binary fields, our inversion-free explicit formulae need only $116M+8S$, $93M+10S$, $37M+16S$ and $23M+11S$ for a group addition, mixed addition, doubling and affine doubling, respectively.

Finally, based on our improvements, we analyze two scalar multiplication algorithms using theta divisors and provide comparisons with the case of using standard divisors for genus 3 HECC over both prime fields and binary fields. Moreover, we implemented efficient doubling explicit formulae in this contribution on a Pentium-4@2.8GHz processor to show the correctness and performance of our new derived formulae for the group operations.

The rest of this paper is organized as follows: Section II summarizes previous contributions addressing the improvements of the genus 3 HEC group operations. Section III gives a short introduction to the mathematical background of genus 3 HECs, presents the standard algorithm (Cantor's Algorithm) and Harley's algorithm to do arithmetic in the ideal class groups of genus 3 HECs. Section IV summarizes all kinds of tricks to derive explicit formulae. Section V deals with the explicit formulae using theta divisors. Section VI discusses efficient doubling for genus 3 HECs over binary fields. Section VII introduces inversion-free arithmetic for genus 3 HECC. Section VIII gives the experimental results of our new derived explicit doubling formulae. Finally, we end this contribution with a discussion of comparisons of our results with the known results and some conclusions in Section IX.

II. Previous Work for Genus 3 HEC

In this section, we review previous improvements of group operations for the genus 3 HECC. In the rest of this paper I represents a field inversion, M a field multiplication, and S a field squiring. In some references, the authors did not distinguish between multiplications and squarings, which is denoted as M/S.

Cantor's algorithm applies to hyperelliptic curves of arbitrary genus. In [84], Nagao accelerated the polynomial arithmetic for Cantor's algorithm and evaluated the computational cost of the improved group operations for genus $2 \leq g \leq 10$. For genus 3 curves over prime fields with $f_i \in \mathbb{F}_2$, Nagao's improvements require $2I+154M/S$ and $2I+132M/S$ for a group addition and doubling, respectively.

Since Harley obtained the first practical explicit formulae [39], [47], most of the improvements concentrate on genus 2 curves [12], [22], [44], [53], [65]–[72], [74], [77], [82], [91], [92], [101]–[103]. For genus 3 curves, the work on improvements for the group operations has been conducted since 2002. In [64], Kuroki *et al.* extended for the first time the Harley's algorithm to genus 3 curves over prime fields and employed the methods from [47], [82] to make further acceleration. The computational cost of their algorithm is $1I+81M/S$ for an addition and $1I+74M/S$ for a doubling. The proposed algorithms were implemented on an Alpha Workstation 21264@667MHz, which take 932 μs for a 160-bit scalar multiplication on a divisor class group. In [88], [89], Pelzl *et al.* further optimized the formulae of [64] and generalized those to arbitrary characteristic. When using special genus 3 curves over binary fields with $h(x)=1$, their explicit formulae can obtain the best results at that time, which require $1I+65M+6S$ and $1I+14M+11S$ for an group addition and a group doubling, respectively. Furthermore, the authors made the first thorough comparisons of the performance of their explicit formulae on different platforms including a Pentium processor and three embedded processors (ARM, ColdFire and PowerPC) [112]. Their improvements and implementations are also summarized in [108], [111]. In [57], Kitamura *et al.* studied fast software

[1] Part of this research was presented in [30], [31]

implementations of group operations for genus 3 curves over binary fields by using the SIMD operations to parallelize the steps in Harley's algorithm. This results in 11% faster than conventional implementations. In 2004, Avanzi [2] gave a comprehensive comparisons for implementing the explicit formulae for prime fields of cryptographically relevant sizes.

A further speed-up for HECs of genus 3 of odd characteristic was achieved in 2004 by Gonda, Matsuo, Aoki, Chao and Tsujii [42], [43]. The authors suggested to use Toom's multiplication and the virtual polynomial multiplication (for more details see Section IV) for improving the results presented in [88], [89], and refined details of these implementations. This algorithm takes only $I + 70M/S$ for an addition and $1I + 71M/S$ for a doubling. In addition, their implementation results show the excellent performance of genus 3 HECC when implemented on 64-bit CPUs. In their implementation, a 160-bit scalar multiplication can be done within $172\mu s$ on a 64-bit CPU Alpha EV68@1.25GHz. In [45], Guyot *et al.* proposed efficient algorithms to compute the resultant of two polynomials and of the inverse of one polynomial modulo another, and improved the overall complexity of the addition and doubling algorithms for both even and odd characteristics. Their explicit formulae are applicable to almost all hyperelliptic curves of genus 3. In 2005, a novel efficient implementation of HECC was proposed by Katagi *et al.* [54], [55]. The authors utilized theta divisors to achieve a fast scalar multiplication and developed a window-based method using theta divisors that is secure against side-channel attacks. However, they only analyzed the details of their proposed method for genus 2 and 3 HECC over binary fields.

During the preparation of this paper, we noted two new results which we state below and will be used in our paper. Avanzi *et al.* [3] and Nyukai *et al.* [86] found independently that the approaches published in the previous explicit formulae which compute a pseudo-inverse via computation of the resultant are not optimal, and proposed a much better method for computation of the pseudo-inverse by using Cramer's rule and expanding the resultant. Their method can save one multiplication compared to that in [45]. In [3], the authors applied this method to the special genus 3 and 4 HECs over binary fields with $h(x) = 1$. In addition, in terms of the characteristics of the field multiplication over binary fields, they presented sequential multiplications which repeatedly use the results of precomputations of field multiplications for a set of multiplications with one of terms in common. Using the sequential multiplications, a group addition takes $1I + 47.7M + 6S$ and a group doubling takes $1I + 9.3M + 11S$ for genus 3 curves over binary fields with $h(x) = 1$. While in [86] the authors used the proposed methods of computing the pseudo-inverse to improve the results of [42], [43] and [90] for genus 3 and 4 HECs, respectively. They implemented their improved explicit formulae for genus 3 HECs over the prime field $\mathbb{F}_{2^{61}-1}$ again in the 64-bit CPU Alpha EV68@1.25GHz and showed that a 160-bit scalar multiplication can be done within $163\mu s$ on that CPU.

The work at hand applies all of tricks available now to improve and optimize the explicit formulae for genus 3 HECs from three aspects. First, we improve explicit formulae for genus 3 curves using theta divisors over binary fields and derive the new formulae for the prime fields case. And then, we find efficient explicit doubling formulae for four types of genus 3 curves over binary fields. Finally, we propose the inversion-free arithmetic for genus 3 HECC. The comparisons of computation complexity of these new formulae with the known results are summarized in Tables I, IV and V.

III. Mathematical Background on Genus 3 Hyperelliptic Curves

In this section, we present a brief introduction to the theory of genus 3 hyperelliptic curves over finite fields of arbitrary characteristic, which is needed in the rest of this paper. For a detailed treatment, the reader is referred to [13], [19], [62], [80].

A. Genus 3 HECs and Their Divisor Class Groups

Let $\mathbb{F}_q$ be a finite field of characteristic p, $q = p^n$, and let $\overline{\mathbb{F}}_q$ denote the algebraic closure of $\mathbb{F}_q$. Let $\mathbb{F}_q(C)/\mathbb{F}_q$ be a quadratic function field defined via an equation

$$C : Y^2 + h(X)Y = F(X) \tag{1}$$

where $F(X) = X^7 + f_6X^6 + f_5X^5 + f_4X^4 + f_3X^3 + f_2X^2 + f_1X + f_0 \in \mathbb{F}_q[X]$ is a monic polynomial of degree 7, $h(X) = h_3X^3 + h_2X^2 + h_1X + h_0 \in \mathbb{F}_q[X]$ is a polynomial of degree at most 3, and there are no solutions $(x, y) \in \overline{\mathbb{F}}_q \times \overline{\mathbb{F}}_q$ which simultaneously satisfy the equation $y^2 + h(x)y = F(x)$ and the partial derivative equations $2y + h(x) = 0$ and $h^{'}(x)y - F^{'}(x) = 0$. The curve $C/\mathbb{F}_q$ associated with this function field is called a *hyperelliptic curve of genus 3 defined over* $\mathbb{F}_q$. For our purpose it is enough to consider a point P as an ordered pair $P = (x, y) \in \overline{\mathbb{F}}_q^2$ which satisfies $y^2 + h(x)y = F(x)$. Besides these tuples there is one point P_∞ at infinity. The inverse of P is defined as $-P = (x, -y - h(x))$. We call a point P that satisfies $P = -P$ a *ramification point*. Note that for the genus 3 HECs over prime fields, it suffices to let $h(X) = 0$ and to have $F(X)$ square free.

In contrast to ECC, points on a hyperelliptic curve do not form a group. Rather than points, divisors are employed. A divisor D of $C(\overline{\mathbb{F}}_q)$ is an element of the free abelian group over the points of $C(\overline{\mathbb{F}}_q)$, e.g. $D = \sum_{P \in C(\overline{\mathbb{F}}_q)} n_P P$ with $n_P \in \mathbb{Z}$ and $n_P = 0$ for almost all points P. The degree of a divisor D is defined as $deg(D) = \sum_{P \in C(\overline{\mathbb{F}}_q)} n_P$. We say that a divisor D is defined over $\mathbb{F}_q$ if $D^\sigma = D$, where $D^\sigma = \sum_{P \in C(\overline{\mathbb{F}}_q)} n_P P^\sigma$, for all automorphisms σ of $\overline{\mathbb{F}}_q$ over $\mathbb{F}_q$. The divisor class group $\mathcal{J}_C(\mathbb{F}_q)$ is defined by the quotient group $\mathbb{D}_0/\mathbb{P}$, where $\mathbb{D}_0$ is a group of degree zero divisors and $\mathbb{P}$ is a group of principal divisors on C, which is a finite formal sum of the zeros and poles.

The divisor class group $\mathcal{J}_C(\mathbb{F}_q)$ of C forms a finite Abelian group and therefore we can construct cryptosystems whose security is based on the discrete logarithm problem on the Jacobian of C. In [13], Cantor pointed out that each element of the divisor class group can be represented uniquely by a so-called reduced divisor. Mumford [83] showed that a reduced divisor $D = \sum m_i P_i - (\sum m_i)P_\infty$ where $m_i \geq 0$, $\sum m_i \leq g$ and $P_i \neq -P_j$ when $i \neq j$, has a nice canonical representation by means of two polynomials $U(X)$ and $V(X)$ defined over $\mathbb{F}_q$, which satisfy the following conditions:

$$U(X) = \prod_i (X - x_i)^{m_i}, \;\; V(x_i) = y_i,$$

$$\deg V < \deg U \leq g, \;\; U \mid V^2 + hV - F.$$

In the remainder of this paper, we will use the notation $[U, V]$ for the divisor class represented by $U(X)$ and $V(X)$. For a genus 3 HEC, we have commonly $[U, V] = [X^3 + u_2X^2 + u_1X + u_0, v_2X^2 + v_1X + v_0]$.

B. Arithmetic Using Cantor's Algorithm

In this section, we give a brief description of the Cantor's algorithm for adding and doubling divisors on the divisor class group $\mathcal{J}_C(\mathbb{F}_q)$ of the HEC C. Here we deal with general HECs, i.e., curves of arbitrary genus. Using Cantor's algorithm to add the divisor classes is divided into two phases. The first phase is to find a semi-reduced divisor $D^{'} = [U^{'}, V^{'}]$, such that $D^{'} \sim D_1 + D_2 = [U_1, V_1] + [U_2, V_2]$ in the divisor class group $\mathcal{J}_C(\mathbb{F}_q)$, which is usually called *composition*. In the second phase, Cantor's algorithm reduces the semi-reduced divisor $D^{'} = [U^{'}, V^{'}]$ into an equivalent reduced divisor $D = [U, V]$. This step is called *reduction*.

Algorithm 1 Cantor's Algorithm for Group Addition

```
Input: $D_1 = [U_1, V_1], D_2 = [U_2, V_2], C : Y^2 + h(X)Y = F(X)$
Output: $D = [U_3, V_3]$ reduced with $D \equiv D_1 + D_2$
I: Composition
1. Compute $d_1 = \gcd\ (U_1, U_2) = e_1U_1 + e_2U_2$
2. Compute $d = \gcd\ (d_1, V_1 + V_2 + h) = c_1d_1 + c_2(V_1 + V_2 + h)$
3. Let $s_1 = c_1e_1, s_2 = c_1e_2, s_3 = c_2$
4. $U^{'} = \frac{U_1U_2}{d^2}$
5. $V^{'} = \frac{s_1U_1V_2 + s_2U_2V_1 + s_3(V_1V_2 + F)}{d} \bmod U^{'}$
II: Reduction
6. Let $U_3 = \frac{F - V^{'}h - V^2}{U^{'}}, V_3 = (-h - V^{'}) \bmod U_3$
7. If $\deg U > g$ put $U^{'} = U_3, V^{'} = V_3$ and goto step 6
8. make $U_3$ monic.
```

If we want to double a divisor class, we can simplify steps $1 \sim 5$ of Algorithm 1 as follows:

```
1. Compute $d = \gcd\ (U, 2V + h) = s_1U + s_3(2V + h)$
   (Note that $U = U_1 = U_2, V = V_1 = V_2$)
2. $U^{'} = \frac{U^2}{d^2}$
3. $V^{'} = \frac{s_1UV + s_3(V^2 + F)}{d} \bmod U^{'}$
```

Cantor's algorithm can be applied to any genus and any characteristics, and it only involves polynomial arithmetic over the finite field in which the divisor class group is defined. However, there are some redundant computations of the polynomial's coefficients in this classical algorithm. Therefore, it is necessary to simplify the Cantor's algorithm by making the steps explicit, which is the idea of Harley's algorithm. We will deal with the Harley's algorithm in the next subsection.

C. Arithmetic Using Harley's Algorithm

Gaudry and Harley in [39] proposed a fast addition algorithm of divisor classes on genus 2 hyperelliptic curves, so-called Harley's algorithm, which is an elegant generalization of the chord-tangent law for the addition of the points on elliptic curves. In order to remove the redundance in Cantor's algorithm, the authors executed a detailed classification for the input divisor classes according to their weights. The *weight* of a divisor is defined as the number of its points [80]. For each case, they derived the corresponding explicit formula. Furthermore, Harley's algorithm employs many modern polynomial computation techniques such as Chinese remainder theorem, Newton's iteration, and Karatsuba's multiplication.

The work of [39] was generalized by Kuroki *et al.* in [64] to genus 3 curves defined over prime fields. From now on we restrict our attentions on curves of genus 3. The authors pointed out that for genus 3 case, a detailed classification based on the weights of the input divisor classes will lead to 6 different cases. Further classification according to the common divisors will result in about 70 subcases. It will not be efficient to develop different procedures for all these cases. Therefore, it is important to consider optimizations and implementations for the most frequent cases (Note that the most frequent cases mean that for addition the inputs are two co-prime polynomials of degree 3, and for doubling the input is a square free polynomial of degree 3) which occur with overwhelming probability of $1 - O(1/q)$ for genus 3 curves over $\mathbb{F}_q$ [84]. For the remaining cases, one usually employs the Cantor's algorithm without affecting the overall performance of the algorithm. In the following, we give a review for the Harley's algorithm. For more details about the deviations of this algorithm, the reader is referred to [111].

1) Addition in Most Frequent Case: In this case we need to compute the addition $D_3 = D_1 + D_2 = (U_3, V_3)$ for reduced divisors $D_1 = (U_1, V_1)$ and $D_2 = (U_2, V_2)$ with $\deg U_1 = \deg U_2 = 3$ and $\gcd\ (U_1, U_2) = 1$.

Algorithm 2 describes all steps of the Harley's algorithm for adding two reduced divisor classes in the most frequent case for genus 3 HECs over finite fields of arbitrary characteristic.

Algorithm 2 Harley's Algorithm for Group Addition ($g = 3$)

```
Input: $D_1 = [U_1, V_1], D_2 = [U_2, V_2], C : Y^2 + h(X)Y = F(X)$
Output: $D_3 = [U_3, V_3]$ reduced with $D_3 \equiv D_1 + D_2$
1. $k = \frac{F - hV_1 - V_1^2}{U_1}$ (exact division)
2. $S \equiv \frac{V_2 - V_1}{U_1} \bmod U_2$
3. $Z = SU_1$
4. $U_t = \frac{k - S(Z + h + 2V_1)}{U_2}$ (exact division)
5. $U_t$ made monic
6. $V_t \equiv -(h + Z + V_1) \bmod U_t$
7. $U_3 = \frac{F - hV_t - V_t^2}{U_t}$ (exact division)
8. $V_3 \equiv -(h + V_t) \bmod U_3$
```

2) Doubling in Most Frequent Case: In this case we need to compute the doubling $D_2 = 2D_1 = (U_2, V_2)$ for reduced divisor $D_1 = (U_1, V_1)$ with $\deg U_1 = 3$ and $\gcd\ (U_1, h + 2V_1) = 1$.

Algorithm 3 describes all steps of the Harley's algorithm for doubling one reduced divisor class in the most frequent case for genus 3 HECs over finite fields of arbitrary characteristic.

Algorithm 3 Harley's Algorithm for Group Doubling ($g = 3$)

```
Input: $D_1 = [U_1, V_1], C : Y^2 + h(X)Y = F(X)$
Output: $D_2 = [U_2, V_2]$ reduced with $D_2 \equiv 2D_1$
1. $k = \frac{F - hV_1 - V_1^2}{U_1}$ (exact division)
2. $S \equiv \frac{k}{h + 2V_1} \bmod U_1$
3. $Z = SU_1$
4. $U_t = S^2 + \frac{S(h+2V_1)-k}{U_1}$ (exact division)
5. $U_t$ made monic
6. $V_t \equiv -(h + Z + V_1) \bmod U_t$
7. $U_3 = \frac{F - hV_t - V_t^2}{U_t}$ (exact division)
8. $V_3 \equiv -(h + V_t) \bmod U_3$
```

D. Security of Genus 3 HECC

The security of HECC is based on the difficulty of the discrete logarithm problem (DLP) on the divisor class group $\mathcal{J}_C(\mathbb{F}_q)$. The hyperelliptic curve discrete logarithm problem (HECDLP) on $\mathcal{J}_C(\mathbb{F}_q)$ can be stated as follows: given two divisors $D_1, D_2 \in \mathcal{J}_C(\mathbb{F}_q)$, determine the smallest integer m such that $D_2 = mD_1$, if such an m exists.

The best algorithm known for solving the DLP is Pollard's rho algorithm [93] and its parallelization by Van Oorschot and Wiener [87]. Pollard's rho algorithm has a purely exponential expected running time of $O(\sqrt{\frac{\pi n}{2}})$ group operations and negligible storage requirements. However, algorithms for solving HECDLP that are faster than Pollard's rho algorithm are found for some families of special HECs. In [34], Frey and Rück showed how to use the Tate pairing to efficiently reduce the DLP in the jacobian $\mathcal{J}_C(\mathbb{F}_q)$ to the DLP in the multiplicative group of an extension field $\mathbb{F}_{q^k}$, where the extension degree k is the smallest positive integer for which $\#\mathcal{J}_C(\mathbb{F}_q)$ (or the largest prime factor of $\#\mathcal{J}_C(\mathbb{F}_q)$) divides $q^k - 1$. For some special types of HECs (e.g. supersingular HECs), k is indeed small and hence the Tate pairing reduction will yield a subexponential-time algorithm for the DLP in $\mathcal{J}_C(\mathbb{F}_q)$. In [35], [98], the authors proved that there are no hyperelliptic supersingular curves of genus $2^n - 1$ over fields of characteristic 2 for any integer $n \geq 2$. Therefore, genus 3 HECs of the form $Y^2 + Y = F(X)$ over binary fields turn out to be the best option according to the complexity of the group operations.

The most powerful algorithm for attacking HECDLP is the index-calculus method which yields a subexponential-time algorithm for the DLP in the jacobian of a high genus hyperelliptic curve. The idea of using index-calculus to solve HECDLP was first proposed in [1], and then was improved and implemented in [27], [33], [38]. These results show that HECs with genus larger than 4 are insecure. In [104], Thériault optimized the algorithm to compute the discrete logarithm in the Jacobian of low genus HECs. Recently, Gaudry *et al.* [41] and Nagao [85] proposed the double large prime variations for small genus HECs index calculus, which is the fastest known attack for the moment. For genus 3 HECs defined over $\mathbb{F}_q$, this attack requires $O(q^{4/3})$ group operations. Therefore, we should choose a finite field about $\frac{3}{8}n$ bits for genus 3 HECC in order to achieve the similar security level as n-bits ECC. In this paper, we will take this recent attack into account when implementing genus 3 HECC.

In addition, one should also consider the Weil descent attack [40], [49], [79] when choosing the field extensions. These attacks show that using fields with composite extension degrees can have cryptographic weakness which can potentially lead to attacks. However, no known variation of the Weil decent attack exits for fields with prime extensions.

IV. Known Tricks to Improve the Explicit Formulae for Genus 3 HECs and Motivations

In practice, all kinds of tricks have been found to improve the efficiency of Harley's algorithm. In this section, we give a brief description of all kinds of tricks used to derive the explicit formulae which will be used in this paper. According to these tricks, we refine Algorithms 2 and 3, which are presented in Algorithms 4 and 5, respectively, and give the corresponding trick used in each step. For more details of these tricks, the reader is referred to the references mentioned in the discussions below.

1) Calculation of the Resultant and the Pseudo-Inverse Using Cramer's Rule: This trick is used in the steps 1 and 2 of Algorithms 4 and 5. In [89], Pelzl *et al.* applied Bézout's determinant to the resultant computation. This result was further improved by Guyot *et al.* in [45]. In other words, they found that it was more efficient to obtain firstly the partial results of the pseudo-inversion, and then compute the resultant using the results having been obtained than to calculate the resultant and the pseudo-inverse separately. They used Cramer's rule implicitly in their algorithm and saved two multiplications compared to Pelzl *et al.*'s algorithm in [89]. Due to the work of Avanzi *et al.* [3] (for the group addition of the even characteristic case) and Nyukai *et al.* [86] (for the odd characteristic case) in 2006, it is possible to save one more multiplication by computing the pseudo-inverse before calculating the resultant. These algorithms use the Cramer's rule explicitly and show an efficient procedure for the expansion of the determinant.

2) Karatsuba Multiplication: Karatsuba and Ofman introduced an algorithm to multiply two polynomials efficiently in [52]. Given $f(x) = ax+b$ and $g(x) = cx+d$, then the product $f(x)g(x) = d_1x^2 + (d_{01} - d_0 - d_1)x + d_0$, where $d_0 = bd$, $d_1 = ac$ and $d_{01} = (a + c)(b + d)$. This trick is used for obtaining the simplified steps 3, 5 and 6 in Algorithm 4 and steps 4 and 6 in Algorithm 5, respectively. Compared to the schoolbook method, Karatsuba multiplication algorithm saves one multiplication at the cost of extra three additions. For more details about Karatsuba multiplication algorithm and its generalizations, the reader is referred to [52], [59], [106].

3) Toom's Multiplication: This trick is only applicable to genus 3 HECs defined over prime fields instead of using Karatsuba multiplication. This algorithm is generally inefficient for low-degree polynomials. However, in the certain case of the group operations of genus 3 curves, each use of Toom's multiplication algorithm can save one multiplication compared

Algorithm 4 Explicit Formula for Group Addition ($g = 3$)

```
Input: $D_1 = [U_1, V_1], D_2 = [U_2, V_2], C : Y^2 + h(X)Y = F(X)$
Output: $D_3 = [U_3, V_3]$ reduced with $D_3 \equiv D_1 + D_2$
Compute:
1. the resultant $r$ of $U_1$ and $U_2$ (Cramer's Rule)
2. the pseudo-inverse $I \equiv \frac{r}{U_1} \bmod U_2 = i_2X^2 + i_1X + i_0$
3. $S^{'} = rS \equiv (V_2 - V_1)I \bmod U_2 = s^{'}_2X^2 + s^{'}_1X + s^{'}_0$ (Karatsuba, Toom)
4. $S = \frac{S^{'}}{r}$ and make $S$ monic: $S = X^2 + s_1X + s_0$ (Montgomery's Trick )
5. $Z = SU_1 = X^5 + z_4X^4 + z_3X^3 + z_2X^2 + z_1X + z_0$ (Karatsuba, Toom)
6. $U_t = [S(Z + \frac{r}{s^{'}_2}(h + 2V_1)) - (\frac{r}{s^{'}_2})^2\frac{F - hV_1 - V_1^2}{U_1}]/U_2$
   $= X^4 + u_{t3}X^3 + u_{t2}X^2 + u_{t1}X + u_{t0}$ (Karatsuba, Efficient Division)
7. $V_t \equiv -(h + s_2Z + V_1) \bmod U_t = v_{t3}X^3 + v_{t2}X^2 + v_{t1}X + v_{t0}$
8. $U_3 = \frac{F - hV_t - V_t^2}{U_t} = X^3 + u_{32}X^2 + u_{31}X + u_{30}$ (Efficient Division)
9. $V_3 \equiv -(h + V_t) \bmod U_3 = v_{32}X^2 + v_{31}X + v_{30}$
```

Algorithm 5 Explicit Formula for Group Doubling ($g = 3$)

```
Input: $D_1 = [U_1, V_1], C : Y^2 + h(X)Y = F(X)$
Output: $D_2 = [U_2, V_2]$ reduced with $D_2 \equiv 2D_1$
Compute:
1. the resultant $r$ of $U_1$ and $h + 2V_1$ (Cramer's Rule)
2. the pseudo-inverse $I \equiv \frac{r}{h+2V_1} \bmod U_1 = i_2X^2 + i_1X + i_0$
3. $Z = \frac{F - hV_1 - V_1^2}{U_1} \bmod U_1 = z_2X^2 + z_1X + z_0$ (Efficient Division)
4. $S^{'} = rS \equiv ZI \bmod U_1 = s^{'}_2X^2 + s^{'}_1X + s^{'}_0$ (Karatsuba, Toom)
5. $S = \frac{S^{'}}{r}$ and make $S$ monic: $S = X^2 + s_1X + s_0$ (Montgomery's Trick )
6. $G = SU_1 = X^5 + g_4X^4 + g_3X^3 + g_2X^2 + g_1X + g_0$ (Karatsuba, Toom)
7. $U_t = [(G + \frac{r}{s^{'}_2}V_1)^2 + \frac{r}{s^{'}_2}hG + (\frac{r}{s^{'}_2})^2(hV_1 - F)]/U_1^2 = X^4 + u_{t3}X^3 + u_{t2}X^2 + u_{t1}X + u_{t0}$
8. $V_t \equiv -(h + wG + V_1) \bmod U_t = v_{t3}X^3 + v_{t2}X^2 + v_{t1}X + v_{t0}$
9. $U_2 = \frac{F - hV_t - V_t^2}{U_t} = X^3 + u_{22}X^2 + u_{21}X + u_{20}$ (Efficient Division)
10. $V_2 \equiv -(h + V_t) \bmod U_2 = v_{22}X^2 + v_{21}X + v_{20}$
```

to that of Karatsuba's algorithm. Toom's multiplication can be applied twice to reduce the number of multiplications in both the group addition (steps 3 and 5 of in Algorithm 4) and the group doubling (steps 4 and 6 in Algorithm 5). For more details about Toom's multiplication algorithm and its application in genus 3 HECC, see [105], [42] and [43].

4) Montgomery's Trick of Simultaneous Inversions: Montgomery discovered the following trick to simultaneously calculate inversions of several elements in order to save inversions at the cost of some multiplications [18]. For given elements a and b in $\mathbb{F}$, a field, the computation of the inverse a^{-1} and b^{-1} can be done as follows: we first compute $c = (ab)^{-1}$, and then obtain $a^{-1} = bc$ and $b^{-1} = ac$. The cost of computing the inverse of two elements with Montgomery's trick is $1I + 3M$. This trick is used in the step 4 of Algorithm 4 and the step 5 of Algorithm 5, respectively, to combine two inversions in the Harley's algorithm (see steps 2 and 5 in Algorithms 2 and 3) at the cost of additional three multiplications.

5) Reordering of the Normalization Step: This trick was firstly proposed by Takahashi for genus 2 HECs in [103]. Takahashi noted that the step of making U_t monic (see the step 5 of Algorithms 2 and 3) is unnecessary if the polynomial S (see the step 2 of Algorithms 2 and 3) is already monic, and showed that applying this trick can obtain the required monic polynomial U_t and save some field operations at the same time. In [111], Thomas *et al.* generalized this method to the genus 3 case to simplify steps 6 and 7 in Algorithm 4 and steps 7 and 8 in Algorithm 5.

6) Efficient Division: This trick is based on the observation that the quotient of two polynomials g_1 and g_2 with degrees deg g_1 and deg g_2, with deg $g_1 >$ deg g_2, only depends on the deg $g_1 -$ deg $g_2 + 1$ highest coefficients of the dividend and the divisor [37]. Therefore, we do not have to consider all the coefficients when computing the quotient of two polynomials. This trick can be applied twice in both the group addition (steps 6 and 8 of Algorithm 4) and the group doubling (steps 3 and 9 of Algorithm 5).

7) Karatsuba Reduction: In [111], Thomas *et al.* used the idea of the Karatsuba multiplication algorithm [52] to compute the modulo reduction of polynomials of arbitrary degrees and applied this technique to improve HECC group operations. Their results show that the complexity of performing polynomial modulo reduction with Karatsuba's algorithm is $O(n^{1.58})$.

8) Virtual Polynomial Multiplication: In [42] and [43], Gonda *et al.* noted that there are many "multiply-and-add" operations in the Harley's algorithm of genus 3 HECs. Therefore, they applied Karatsuba's multiplication twice to reduce the number of field multiplications in the procedure of the group addition when an appropriate sequence of "multiply-and-add" operations appear.

9) Sequential Multiplication: This trick is only applicable to the group operations of genus 3 HECs over binary fields. In [3], Avanzi *et al.* found that there are several sets of multiplications with one of terms in common in the Harley's algorithm of genus two to four. Furthermore, they noted that the field multiplication algorithm over binary fields includes a precomputation part. Therefore, it is possible to improve the performance of the group operations of genus two to four HEC by repeatedly using the results of this precomputation part. Although we use the trick of the sequential multiplication to optimize the explicit formulae for genus 3 HECs defined over binary fields in this paper, we will not estimate the cost because the performance of this trick significantly depends on which kind of processors are used.

10) Use of Special Divisors: The group elements are points over an elliptic curve in ECC, whereas the elements in the Jacobian of a HEC are (reduced) divisors. For the case of genus 3 HECs, a divisor can have the weights 0, 1, 2 or 3. Harley's algorithm only deals with the most frequent case, in which the input divisors have weight 3. However, Katagi *et al.* found that using the special divisors whose weights are less than 3 can improve the performance of scalar multiplication for genus 3 HECC over binary fields [54], [55]. We will revisit this method and improve their explicit formulae in Section V.

11) Choice of HEC with Certain Properties: This trick has important influence on the performance of the group operations of genus 3 HECs over binary fields. Going into the details of the Harley's algorithm one can notice that the actual execution of the steps depends on the coefficients of the curves. Therefore, it is possible to use certain families of curves over which the execution of the group operations require less field operations than those of general curves. In this contribution, we find four families of genus 3 HECs over binary fields which have extremely fast group doubling. This will be presented in Section VI.

12) Use of Inversion-Free Arithmetic: In Harley' algorithm, each group operation requires one inversion. However, there exist application environments, for example smart cards, where inversions are extremely time or space critical. In this case, inversion-free group operations will be practical and advantageous. We will present this trick in Section VII.

V. Improved Explicit Formulae with Theta Divisors

In [54] and [55], Katagi *et al.* proposed the method of using Theta (or Degenerate) divisors to accelerate the scalar multiplication for genus 3 HECC. They discussed the case of binary fields and estimated the cost of the Theta divisor method. However, we find that there still exist some redundant operations in their explicit formulae. In this section, we further simplify their explicit formulae using all kinds of tricks summarized in Section IV, generalize their idea to the case of prime fields and estimate the cost of the two scalar multiplication algorithms in both cases.

For a genus 3 HEC defined over $\mathbb{F}_q$, a *Theta (or Degenerate) divisor* is a reduced divisor with the weight less than 3. Harley's algorithm only deals with the group operations in the most frequent case. For the rest cases, some of which are caused by the Theta divisors, one usually employs the Cantor's algorithm. Although Theta divisors occur with a low probability, the group operations using Theta divisors are much cheaper than those in the most frequent case. Therefore, if one can utilize the Theta divisors as the inputs of the scalar multiplication algorithms or determine under which conditions the Theta divisors will appear during the procedure of the scalar multiplication, the performance of genus 3 HECC will be improved. In addition, Theta divisors are also related closely to some pairing-based cryptographic protocols [36], [73]. Therefore, efficient explicit formulae with Theta divisors will be useful for both HECC and pairing-based cryptosystem.

Katagi *et al.* considered in [54] and [55] the following five cases for simplifying formulae related to Theta divisors over genus 3 HECs:

1) ADD$^{3+1\to 3}$: $D_3 = D_1 + D_2$ $(D_1 \neq D_2)$
deg $U_2 = 1$, deg $U_1 =$ deg $U_3 = 3$, gcd$(U_1, U_2) = 1$.

2) ADD$^{3+2\to 3}$: $D_3 = D_1 + D_2$ $(D_1 \neq D_2)$
deg $U_2 = 2$, deg $U_1 =$ deg $U_3 = 3$, gcd$(U_1, U_2) = 1$.

3) ADD$^{1+2\to 3}$: $D_3 = D_1 + D_2$ $(D_2 = 2D_1)$
deg $U_1 = 1$, deg $U_2 = 2$, deg $U_3 = 3$.

4) DBL$^{1\to 2}$: $D_2 = 2D_1$
deg $U_1 = 1$, deg $U_2 = 2$, gcd$(h + 2V_1, U_1) = 1$.

5) DBL$^{2\to 3}$: $D_2 = 2D_1$
deg $U_1 = 2$, deg $U_2 = 3$, gcd$(h + 2V_1, U_1) = 1$.

where ADD$^{i+j\to k}$ denotes the divisor class addition $D_3 = [U_3, V_3] = D_1 + D_2 = [U_1, V_1] + [U_2, V_2]$, and DBL$^{i\to j}$ denote the divisor class doubling $D_2 = [U_2, V_2] = 2D_1 = 2[U_1, V_1]$ (i, j and k are the degrees of U_1, U_2 and U_3, respectively).

For the genus 3 HECs defined over prime fields and binary fields, we derive a new set of explicit formulae for the above five cases, respectively. All explicit formulae are shown in Tables VII $\sim$ XVI in the appendix. We compare the computational cost of our explicit formulae with those derived by Katagi *et al.* in Table I. Applying all kinds of tricks presented in section IV we are able to save about $20\% \sim 50\%$ of the multiplications compared to Katagi *et al.*'s formulae in four cases for genus 3 HECs over binary fields.

We estimate the computational cost for the double-and-add-always method and SPA-resistant width-wNAF method using theta devisors based on our newly derived explicit formulae. We summarize the results and the comparisons with those in [55] in Table II, where a secrete scalar value is 160-bit and 'standard' denotes a divisor with weight 3 which corresponds to the most frequent case. The computational complexity for the group operations in the most frequent case can be found in Table IV. Using our explicit formulae, the performance of the two scalar multiplication algorithms above increases by about $14\% \sim 20\%$.

TABLE I
SPEEDING UP GENUS THREE HECC USING THETA DIVISORS

	Finite	Curve	Cost				
	Field	Properties	ADD$^{3+2\to3}$	ADD$^{3+1\to3}$	ADD$^{1+2\to3}$	DBL$^{1\to2}$	DBL$^{2\to3}$
Katagi *et al.* [55]	$\mathbb{F}_{2^n}$	deg $h=3, h_3=1$	$1I+52M$	$1I+20M$	$1I+28M$	$1I+21M$	$1I+44M$
Our work	$\mathbb{F}_p$	$h(X)=0, f_6=0$	$1I+44M$ Table VII	$1I+21M$ Table VIII	$1I+18M$ Table IX	$1I+11M$ Table X	$1I+28M$ Table XI
	$\mathbb{F}_{2^n}$	deg $h=3, h_3=1$	$1I+41M$ Table XII	$1I+20M$ Table XIII	$1I+19M$ Table XIV	$1I+12M$ Table XV	$1I+32M$ Table XVI

TABLE II
COST OF SCALAR MULTIPLICATION USING THETA AND STANDARD DIVISORS

Base	Weight of	Scalar Multiplication	Katagi *et al.* [55]	Our work	
Divisor	Divisor	Algorithm	$\mathbb{F}_{2^n}$	$\mathbb{F}_{2^n}$	$\mathbb{F}_p$
Theta	1	Double-and-Add	$318I+15989M$	$318I+12819M$	$318I+12501M$
Theta	2	Double-and-Add	$318I+21110M$	$318I+17453M$	$318I+17768M$
Theta	2	wNAF ($w=2$)	$237I+16869M$	$237I+14132M$	$237I+14204M$
Standard	3	Double-and-Add	$318I+25281M$	$318I+21783M$	$318I+21465M$
Standard	3	wNAF ($w=2$)	$237I+18960M$	$237I+16275M$	$237I+16038M$
Standard	3	wNAF ($w=3$)	$212I+17013M$	$212I+14575M$	$212I+14363M$
Standard	3	wNAF ($w=4$)	$195I+15678M$	$195I+13419M$	$195I+13224M$

VI. Efficient Doubling on Genus 3 Curves over Binary Fields

In this section, we generalize the method proposed by Lange and Stevens for genus 2 HECs in [74] to the genus 3 case which can significantly improve the results in [45]. We present four families of genus 3 HECs defined over binary fields for which we find efficient algorithms to calculate the divisor class doubling. Although some curves are very special, we are not aware of any security limitation of the curves that we used in this paper. Our results allow to choose curves from a large variety which have extremely fast doubling with requiring only one-third the time of an addition in the best case.

In order to simplify the equation of curves, we firstly review the isomorphic transformations among genus 3 HECs. For a genus 3 HEC $C/\mathbb{F}_q$ given by the equation (1), the following coordinate transformations

$$(X,Y)\mapsto(\alpha^2\tilde{X}+\beta,\alpha^7\tilde{Y}+a\tilde{X}^3+b\tilde{X}^2+c\tilde{X}+d),$$

where $\alpha,\beta,a,b,c,d\in\mathbb{F}_q$ with $\alpha\neq0$, are isomorphic transformations between the curve C and $C^{'}$: $\tilde{Y}^2+\tilde{h}(\tilde{X})=\tilde{F}(\tilde{X})$, where $\tilde{h},\tilde{F}\in\mathbb{F}_q[X]$ and can be expressed in terms of h,F,a,b,c,d,α and β [76]. These isomorphic transformations associate each point of C to a point of $C^{'}$.

Table III shows all curves suggested in this paper according to the degree of $h(X)$, the coordinate transformations and the corresponding isomorphic curves.

We now study the different cases of the equations depending on the degree of h because the actual execution of the Harley's algorithm depends on the coefficients of the curves. Especially, the coefficients of $h(X)$ have a significant influence on the computational complexity of the doubling algorithms. We will present the explicit formulae for the following four cases: deg $h=0$, deg $h=1$, deg $h=2$ and deg $h=3$. We firstly construct the isomorphic transformations to achieve as many zero coefficients as possible, and then make strong use of the defining equation of the curve to derive explicit doubling formulae. The major speedup is obtained by simplifying r (see step 1 of Algorithm 5) and canceling it in the following steps. For the cases of deg $h=2$ and deg $h=3$, we did not find a way to simply and cancel r for general curves. Therefore, we only discuss special kinds of curves which allow for a significant speedup. Using these special curves, we can obtain the explicit formulae with low complexity and better performance regarding the number of required field operations for the execution of the group operations.

A. Case deg $h=0$

In this subsection we assume deg $h=0$. One can obtain an isomorphic curve where $f_6=f_5=f_4=f_2=0$ and h_0 is divided by any α^7. To improve the efficiency of HECC, we hope that the coefficient h_0 is 'small'[2] in an isomorphic curve. Hence we will choose α^7 such that $\frac{h_0}{\alpha^7}$ is 'small' in the practical use. If we choose finite fields $\mathbb{F}_{2^n}$ with $n\equiv1$ mod 3 or $n\equiv2$ mod 3 there are no elements $\alpha\in\mathbb{F}_{2^n}$ such that $\alpha^7=1$ (the unit element of $\mathbb{F}_{2^n}$). Therefore, there is always an α such that $\alpha^7=h_0$. For $n\equiv0$ mod 3 this happens with probability $1/7$. Using the birational transformations of variables listed in Table III and dividing the equation by α^{14}, we obtain a curve of the form $Y^2+h_0Y=X^7+f_3X^3+f_1X+f_0$, usually with $h_0=1$. Adding a constant term to the substitution of $\widetilde{Y}$ one can achieve $f_0=0$ with probability $1/2$. Hence, there are only two parameters f_3,f_1 as opposed to five in the general case showing that the type is indeed special.

[2] In this section, we will call 'small' an element in $\mathbb{F}_{2^n}$ that is represented by a polynomial with almost all its coefficients equal to zero, so that multiplications by such an element can be performed via few additions and are almost for free

TABLE III

FOUR FAMILIES OF GENUS 3 HEC OVER $\mathbb{F}_{2^n}$ AND THEIR ISOMORPHIC CURVES

	Coordinate Transformation	Isomorphic	Zero	Zero Coefficient
$h(X)$	$(X,Y)\mapsto(\alpha^2\tilde{X}+\beta,\alpha^7\tilde{Y}+a\tilde{X}^3+b\tilde{X}^2+c\tilde{X}+d)$	Curve	Coefficients	with Probability $\frac{1}{2}$
h_0	$\beta=\sqrt{f_5}, a=\alpha^6\sqrt{f_6+\beta}$ $b=\alpha^4\sqrt{f_4+f_5\cdot\beta+f_6\cdot\beta^2+\beta^3}$ $c=\alpha^2\sqrt{f_2+f_3\cdot\beta+f_6\cdot\beta^4+\beta^5+h_0\cdot b}$	$Y^2+h_0Y=X^7+f_3X^3+f_1X+f_0$	f_6,f_5,f_4,f_2	f_0
h_1X+h_0	$\beta=\frac{h_0}{h_1}, a=\alpha^6\sqrt{f_6+\beta}$ $b=\alpha^4\sqrt{f_4+f_5\beta+f_6\beta^2+\beta^3+h_1\sqrt{f_6+\beta}}$ $d=\frac{f_1+f_3\beta^2+f_5\beta^4+\beta^6}{h_1}$	$Y^2+h_1XY=X^7+f_5X^5+f_3X^3+f_2X^2+f_0$	f_6,f_4,f_1	f_2
$h_2X^2+h_0$	$\beta=\sqrt{\frac{h_0}{h_2}}, a=\alpha^6\cdot\frac{f_5+\beta^2}{h_2}, c=\alpha^2\cdot\frac{f_3+\beta^4}{h_2}$ $d=\frac{h_2^2(f_2+f_3\beta+f_6\beta^4+\beta^5)+f_3^2+\beta^8}{h_2^3}$	$Y^2+h_2X^2Y=X^7+f_6X^6+f_4X^4+f_1X+f_0$	f_5,f_3,f_2	f_4
h_3X^3	$\alpha=h_3,\beta=0,a=0,b=h_3^3f_5$ $c=\frac{f_4h_3^2+f_5^2}{h_3}, d=\frac{f_3}{h_3}$	$Y^2+X^3Y=X^7+f_6X^6+f_2X^2+f_1X+f_0$	f_5,f_4,f_3	f_6

With the new curve coefficients the expression r and $S^{'}$ will simplify to:

$$r=h_0^3, s_2^{'}=h_0^2z_2, s_1^{'}=h_0^2z_1, s_0^{'}=h_0^2z_0.$$

We note that

$$\begin{aligned}
u_{t3}&=0,\\
u_{t2}&=s_1^2=(s_1^{'}/s_2^{'})^2=(z_1/z_2)^2,\\
u_{t1}&=(r/s_2^{'})^2=h_0^2(z_2^{-1})^2,\\
u_{t0}&=s_0^2=(s_0^{'}/s_2^{'})^2=(z_0/z_2)^2,
\end{aligned}$$

and

$$\begin{aligned}
v_{t3}&=(u_{t2}+g_3)(s_2^{'}/r)\\
&=h_0^{-1}(u_{t2}z_2+z_0+u_{12}z_1+u_{11}z_2),\\
v_{t2}&=(g_4u_{t2}+u_{t1}+g_2)(s_2^{'}/r)+v_{12}\\
&=h_0^{-1}[(u_{12}z_2+z_1)u_{t2}+u_{t1}z_2+u_{12}z_0\\
&\quad+u_{11}z_1+u_{10}z_2]+v_{12},\\
v_{t1}&=(g_4u_{t1}+u_{t0}+g_1)(s_2^{'}/r)+v_{11}\\
&=h_0^{-1}[(u_{12}z_2+z_1)u_{t1}+u_{t0}z_2+u_{11}z_0\\
&\quad+u_{10}z_1]+v_{11},\\
v_{t0}&=(g_4u_{t0}+g_0)(s_2^{'}/r)+h_0+v_{10}\\
&=h_0^{-1}[(u_{12}z_2+z_1)u_{t0}+u_{10}z_0]+h_0+v_{10}.
\end{aligned}$$

Since $F+hV_1+V_1^2=U_1Z+U_1^2X$ we also have

$$\begin{aligned}
f_0+h_0v_{10}+v_{10}^2&=u_{10}z_0,\\
f_1+h_0v_{11}&=u_{11}z_0+u_{10}z_1+u_{10}^2,\\
h_0v_{12}+v_{11}^2&=u_{12}z_0+u_{11}z_1+u_{10}z_2,\\
f_3&=z_0+u_{12}z_1+u_{11}z_2+u_{11}^2,\\
v_{12}^2&=z_1+u_{12}z_2,\\
0&=u_{12}^2+z_2.
\end{aligned}$$

Using the equations above, we can calculate cheaply u_{t2},u_{t0} and $v_{t3},v_{t2},v_{t1},v_{t0}$ as follows:

$$\begin{aligned}
u_{t2}&=(z_1/z_2)^2=[(v_{12}^2+u_{12}z_2)/z_2]^2\\
&=(v_{12}^2z_2^{-1})^2+u_{12}^2,\\
u_{t0}&=(z_0/z_2)^2=[(f_3+u_{11}^2+u_{12}z_1+u_{11}z_2)/z_2]^2\\
&=[(f_3+u_{11}^2)z_2^{-1}]^2+u_{11}^2+u_{12}^2u_{t2},\\
v_{t3}&=h_0^{-1}(u_{t2}z_2+f_3+u_{11}^2),\\
v_{t2}&=h_0^{-1}(v_{12}^2u_{t2}+v_{11}^2)+h_0z_2^{-1},\\
v_{t1}&=h_0^{-1}[(v_{12}^2+z_2)(u_{t1}+u_{t0})+v_{12}^2u_{t0}+f_1\\
&\quad+u_{10}^2]+h_0z_2^{-1},\\
v_{t0}&=h_0^{-1}(v_{12}^2u_{t0}+f_0+v_{10}^2)+h_0.
\end{aligned}$$

We give the doubling formula for this case in Table XXVI. The operations are counted for the case $h_0=1, h_0^{-1}$ is 'small', and arbitrary h_0. Both h_0^2 and h_0^{-1} are precomputed. The corresponding addition formula which requires $1I+57M+6S$ to add two reduced divisor classes can be found in Table XXV in the appendix. When $h_0=1$ we can save one more multiplication than the best known algorithm and obtain the fastest explicit doubling formula which only needs $1I+10M+11S$.

B. Case deg $h=1$

In this subsection we discuss the case of deg $h=1$. One can obtain an isomorphic curve where $f_6=f_4=f_1=h_0=0$ and h_1 is divided by any α^5. We will choose α^5 such that $\frac{h_1}{\alpha^5}$ is 'small' in the practical use. If we choose finite fields $\mathbb{F}_{2^n}$ with n not being divided by 4 there are no elements $\alpha\in\mathbb{F}_{2^n}$ such that $\alpha^5=1$. Therefore, there is always an α such that $\alpha^5=h_0$. For $n\equiv 0$ mod 4 this happens with probability $1/5$. Using the birational transformations of variables listed in Table III and dividing the equation by α^{14}, we obtain a curve of the form $Y^2+h_1XY=X^7+f_5X^5+f_3X^3+f_2X^2+f_0$, usually with $h_1=1$. Adding a linear factor to the substitution of $\widetilde{Y}$ one can achieve $f_2=0$ with probability $1/2$. Therefore, there are only three free parameters f_5,f_3,f_0.

With the new curve coefficients the expression r and $S^{'}$ will simplify to:

$$\begin{aligned} r &= u_{10}h_1^3, \\ s_2^{'} &= z_0 h_1^2, \\ s_1^{'} &= (u_{12}z_0 + u_{10}z_2)h_1^2, \\ s_0^{'} &= (u_{11}z_0 + u_{10}z_1)h_1^2, \\ rs_2^{'} &= u_{10}z_0 h_1^5, \\ s_2 &= s_2^{'}/r = z_0/(u_{10}h_1). \end{aligned}$$

In this case, we have

$$\begin{aligned} u_{t3} &= 0, \\ u_{t2} &= s_1^2 = (s_1^{'}/s_2^{'})^2 = (u_{12} + u_{10}\cdot\frac{z_2}{z_0})^2, \\ u_{t1} &= (r/s_2^{'})^2 = h_0^2 u_{10}^2 (z_0^{-1})^2, \\ u_{t0} &= s_0^2 = (s_0^{'}/s_2^{'})^2 = (u_{11} + u_{10}\cdot\frac{z_1}{z_0})^2, \end{aligned}$$

and

$$\begin{aligned} v_{t3} &= (u_{t2} + g_3)(s_2^{'}/r) \\ &= h_1^{-1}[z_2\cdot(u_{10}\cdot\frac{z_2}{z_0}) + z_1 + u_{12}z_2], \\ v_{t2} &= (g_4u_{t2} + u_{t1} + g_2)(s_2^{'}/r) + v_{12} \\ &= h_1^{-1}(z_2u_2^{'} + \frac{h_1}{s_2} + u_{12}z_1 + u_{11}z_2 + z_0) + v_{12}, \\ v_{t1} &= (g_4u_{t1} + u_{t0} + g_1)(s_2^{'}/r) + h_1 + v_{11} \\ &= h_1^{-1}[\frac{1}{h_1s_2}(\frac{z_2h_1}{s_2} + z_1^2) + u_{12}z_0 + u_{11}z_1 \\ &\quad +u_{10}z_2] + v_{11}, \\ v_{t0} &= (g_4u_{t0} + g_0)(s_2^{'}/r) + v_{10} \\ &= h_1^{-1}(z_2u_0^{'} + u_{11}z_0 + u_{10}z_1) + v_{10}. \end{aligned}$$

Since $F + hV_1 + V_1^2 = U_1Z + U_1^2x$ we also obtain that

$$\begin{aligned} f_0 + v_{10}^2 &= u_{10}z_0 \quad (= rs_2^{'}/h_1^5), \\ h_1v_{10} &= u_{11}z_0 + u_{10}z_1 + u_{10}^2, \\ f_2 + h_1v_{11} + v_{11}^2 &= u_{12}z_0 + u_{11}z_1 + u_{10}z_2, \\ f_3 + h_1v_{12} &= z_0 + u_{12}z_1 + u_{11}z_2 + u_{11}^2, \\ v_{12}^2 &= z_1 + u_{12}z_2, \\ f_5 &= u_{12}^2 + z_2. \end{aligned}$$

Using the equations above, we can calculate cheaply u_{t2}, u_{t0} and $v_{t3}, v_{t2}, v_{t1}, v_{t0}$ as follows:

$$\begin{aligned} u_{t2} &= (u_{12} + u_{10}\cdot\frac{z_2}{z_0})^2 = (u_{12} + \frac{z_2}{h_1s_2})^2, \\ u_{t0} &= (u_{11} + u_{10}\cdot\frac{z_1}{z_0})^2 = (u_{11} + \frac{z_1}{h_1s_2})^2, \\ v_{t3} &= h_1^{-1}(\frac{z_2^2}{h_1s_2} + v_{12}^2), \\ v_{t2} &= h_1^{-1}(z_2u_2^{'} + \frac{h_1}{s_2} + f_3 + u_{11}^2), \\ v_{t1} &= h_1^{-1}[\frac{1}{h_1s_2}(z_2\cdot\frac{h_1}{s_2} + z_1^2) + f_2 + v_{11}^2], \\ v_{t0} &= h_1^{-1}(z_2u_0^{'} + u_{10}^2). \end{aligned}$$

We note that $f_0 + v_{10}^2 = u_{10}z_0 = \frac{rs_2^{'}}{h_1^5}$, so it is very cheap to calculate $rs_2^{'}$ as the exact coefficients of Z are not necessary. In Table XXIV, we present the doubling formula for this case. The operations are counted for the case $h_1 = 1, h_1^{-1}$ is 'small' (multiplication with h_1^{-1} are not counted), and arbitrary h_1. Both h_1^2 and h_1^{-1} are precomputed. In the step 2 the inversion and multiplication with k_0 can also be replaced by a division as the inverse is not used later on. The corresponding addition formula which requires $1I + 57M + 6S$ (h_1 is 'small') or $1I + 58M + 6S$ (h_1 is arbitrary) to add two reduced divisor classes is showed in Table XXIII in the appendix. Compared with the explicit formula in [45], which costs $1I + 44M + 6S$ for doubling a divisor class, our formula requires only $1I + 13M + 13S$ and therefore can save $31M$ at the cost of extra $7S$ (note that a squaring is usually more efficient than a multiplication in binary fields).

C. *Case deg* $h = 2$

If h is of degree two then we cannot make any of its coefficients zero in general. In this subsection we will discuss special curves with $h_1 = 0$, that is, the curves having the form $Y^2 + (h_2X^2 + h_0)Y = X^7 + f_6X^6 + f_5X^5 + f_4X^4 + f_3X^3 + f_2X^2 + f_1X + f_0$, which allows for a significant speedup. By making a change of coordinates we can obtain $f_5 = f_3 = f_2 = h_0 = 0$ and h_2 is divided by any α^3. We will choose α^3 such that $\frac{h_2}{\alpha^3}$ is 'small' in the practical use. If, as usual, one choose finite $\mathbb{F}_{2^n}$ with n odd there are no non-trivial cubic roots of the unity. Hence, there is always an α such that $\alpha^3 = h_2$. For even n this happens with probability $1/3$. Using the birational transformations of variables listed in Table III an dividing the equation by α^{14}, we obtain the isomorphic curve of the form $Y^2 + h_2X^2Y = X^7 + f_6X^6 + f_4X^4 + f_1X + f_0$, usually with $h_2 = 1$. Adding a quadratic factor to the substitution of $\widetilde{Y}$ one can achieve $f_4 = 0$ with probability $1/2$. Accordingly, there are only three free parameters f_6, f_1, f_0.

Then the expressions for r, $S^{'}$ and S will simplify to:

$$\begin{aligned} r &= u_{10}^2h_2^3, \\ s_2^{'} &= (u_{11}z_0 + u_{10}z_1)h_2^2, \\ s_1^{'} &= [u_{12}(u_{11}z_0 + u_{10}z_1) + u_{10}z_0]h_2^2, \\ s_0^{'} &= [u_{11}(u_{11}z_0 + u_{10}z_1) + u_{10}(u_{12}z_0 + u_{10}z_2)]h_2^2, \\ s_1 &= s_1^{'}/s_2^{'} = u_{12} + k_1, \\ s_0 &= s_0^{'}/s_2^{'} = u_{11} + k_2. \end{aligned}$$

where $k_1 = \frac{u_{10}z_0}{u_{11}z_0+u_{10}z_1}$ and $k_2 = \frac{u_{10}(u_{12}z_0+u_{10}z_2)}{u_{11}z_0+u_{10}z_1}$. In this case, we have

$$\begin{aligned} u_{t3} &= 0, \\ u_{t2} &= s_1^2, \\ u_{t1} &= \frac{r}{s_2^{'}}(h_2 + \frac{r}{s_2^{'}}) = h_2^2w_1(1 + w_1), \\ u_{t0} &= \frac{r}{s_2^{'}}[h_2(u_2 + s_1) + \frac{rf_6}{s_2^{'}}] + s_0^2 \\ &= h_2^2w_1(k_1 + f_6w_1) + s_0^2, \end{aligned}$$

where $w_1 = \frac{u_{10}^2}{u_{11}z_0+u_{10}z_1}$, and

$$\begin{aligned}
v_{t3} &= (u_{t2}+g_3)(s_2^{'}/r) = h_2^{-1}[z_2 + \frac{(u_{10}z_0)^2}{u_{10}^2(u_{11}z_0+u_{10}z_1)}],\\
v_{t2} &= (g_4u_{t2}+u_{t1}+g_2)(s_2^{'}/r)+h_2+v_{12}\\
&= h_2^{-1}[z_1+u_{12}z_2+\frac{(u_{10}z_0)k_1^2}{u_{10}^2}]+h_2w_1+v_{12},\\
v_{t1} &= (g_4u_{t1}+u_{t0}+g_1)(s_2^{'}/r)+v_{11}\\
&= h_2^{-1}[z_0+u_{11}z_2+u_{12}z_1+\frac{(u_{12}z_0+u_{10}z_2)^2}{u_{11}z_0+u_{10}z_1}]\\
&\quad +(h_2w_1)(f_6+k_1)+v_{11},\\
v_{t0} &= (g_4u_{t0}+g_0)(s_2^{'}/r)+v_{10}\\
&= h_2^{-1}[u_{12}z_0+u_{11}z_1+u_{10}z_2+\frac{(u_{10}z_0)k_2^2}{u_{10}^2}]\\
&\quad +(h_2k_1)(k_1+f_6w_1)+v_{10}.
\end{aligned}$$

And since $F+hV_1+V_1^2 = U_1Z+U_1^2(X+f_6)$, we also have

$$\begin{aligned}
f_0+v_{10}^2 &= u_{10}z_0+f_6u_{10}^2,\\
f_1 &= u_{11}z_0+u_{10}z_1+u_{10}^2,\\
h_2v_{10} &= u_{12}z_0+u_{11}z_1+u_{10}z_2+f_6u_{11}^2,\\
h_2v_{11} &= z_0+u_{12}z_1+u_{11}z_2+u_{11}^2,\\
f_4+h_2v_{12}+v_{12}^2 &= z_1+u_{12}z_2+f_6u_{12}^2,\\
0 &= u_{12}^2+z_2.
\end{aligned}$$

We use the equations above to calculate k_1, k_2, w_1 and $v_{t3}, v_{t2}, v_{t1}, v_{t0}$ cheaper:

$$\begin{aligned}
k_1 &= \frac{f_0+v_{10}^2+f_6u_{10}^2}{f_1+u_{10}^2},\\
k_2 &= \frac{u_{10}(h_2v_{10}+u_{11}z_1+f_6u_{11}^2)}{f_1+u_{10}^2},\\
w_1 &= \frac{u_{10}^2}{f_1+u_{10}^2},\\
v_{t3} &= h_2^{-1}[z_2+\frac{(f_0+v_{10}^2+f_6u_{10}^2)^2}{u_{10}^2(f_1+u_{10}^2)}],\\
v_{t2} &= h_2^{-1}[f_4+v_{12}^2+f_6u_{12}^2+\frac{(f_0+v_{10}^2+f_6u_{10}^2)k_1^2}{u_{10}^2}]\\
&\quad +\frac{h_2u_{10}^2}{f_1+u_{10}^2},\\
v_{t1} &= h_2^{-1}[u_{11}^2+\frac{(u_{11}z_1+f_6u_{11}^2+h_2v_{10})^2}{f_1+u_{10}^2}]\\
&\quad +\frac{h_2u_{10}^2(f_6+k_1)}{f_1+u_{10}^2},\\
v_{t0} &= h_2^{-1}[f_6u_{11}^2+\frac{(f_0+v_{10}^2+f_6u_{10}^2)k_2^2}{u_{10}^2}]\\
&\quad +(h_2k_1)(k_1+\frac{u_{10}^2f_6}{f_1+u_{10}^2}).
\end{aligned}$$

We note that $rs_2^{'} = u_{10}^2\cdot(u_{11}z_0+u_{10}z_1)\cdot h_2^5 = u_{10}^2\cdot(f_1+u_0^2)\cdot h_2^5$, so it is very cheap to calculate $rs_2^{'}$ since we do not need to know the exact coefficients of Z. We describe the doubling formula for this case in Table XXII. The operations are counted for the case $h_2=1, h_2^{-1}$ is 'small' (multiplication with h_2^{-1} are not counted), and arbitrary h_2. Both h_2^2 and h_2^{-1} are precomputed. The corresponding addition formula which requires $1I+58M+6S$ (h_2 is 'small') or $1I+60M+5S$ (h_2 is arbitrary) to add two reduced divisor classes is showed in Table XXI in the appendix. For the general case with $h(x)=h_2x^2+h_1x+h_0$, Guyot *et al.* in [45] use a birational transformation to make the curve's coefficient f_6 zero. Their algorithm needs $1I+52M+8S$ to compute the divisor class doubling. Using special curves with $h(x)=h_2x^2+h_0$, our explicit formula requires only $1I+24M+12S$ when $h_2=1$. In the formulae presented in Table XXI there are four counted multiplications with f_6 which are cheaper when f_6 is 'small'.

D. *Case deg* $h=3$

When h is of degree three, we cannot also make any of its coefficients zero in general. We will show that special curves with $h_2=h_1=h_0=0$ can obtain excellent performance in this section. We can construct a change of coordinates to make $f_5=f_4=f_3=0$ and $h_3=1$. Using the birational transformations of variables listed in Table III an dividing the equation by h_3^{14}, we obtain a curve of the form $Y^2+X^3Y = X^7+f_6X^6+f_2X^2+f_1X+f_0$. Adding a cubic factor to the substitution of $\widetilde{Y}$ one can achieve $f_6=0$ with probability $1/2$. Thereby, there are only three free parameters f_2, f_1, f_0.

Then the expressions for r, $S^{'}$ and S will simplify to:

$$\begin{aligned}
r &= u_{10}^3,\\
s_2^{'} &= u_{10}(u_{12}z_0+u_{11}z_1+u_{10}z_2)+u_{11}^2z_0,\\
s_1^{'} &= u_{12}[u_{10}(u_{12}z_0+u_{11}z_1+u_{10}z_2)]\\
&\quad +u_{10}(u_{11}z_0+u_{10}z_1),\\
s_0^{'} &= u_{11}[u_{10}(u_{12}z_0+u_{11}z_1+u_{10}z_2)]\\
&\quad +u_{10}[u_2(u_{11}z_0+u_{10}z_1)+u_{10}z_0],\\
s_1 &= s_1^{'}/s_2^{'} = u_{12}+k_1,\\
s_0 &= s_0^{'}/s_2^{'} = u_{11}+k_2.
\end{aligned}$$

where $k_1 = \frac{u_{10}(u_{11}z_0+u_{10}z_1)}{u_{10}(u_{12}z_0+u_{11}z_1+u_{10}z_2)+u_{11}^2z_0}$ and $k_2 = \frac{u_{10}[u_{12}(u_{11}z_0+u_{10}z_1)+u_{10}z_0]}{u_{10}(u_{12}z_0+u_{11}z_1+u_{10}z_2)+u_{11}^2z_0}$. In this case, we have

$$\begin{aligned}
u_{t3} &= 0,\\
u_{t2} &= s_1^2+\frac{r}{s_2^{'}},\\
u_{t1} &= \frac{r}{s_2^{'}}(k_1+\frac{r}{s_2^{'}}),\\
u_{t0} &= \frac{r}{s_2^{'}}(k_2+u_{12}k_1+\frac{rf_6}{s_2^{'}})+s_0^2,\\
v_{t3} &= (u_{t2}+g_3)(s_2^{'}/r)+1\\
&= \frac{u_{10}z_0}{u_{10}^2}+\frac{(u_{11}z_0+u_{10}z_1)^2}{u_{10}^2(u_{12}z_0+u_{11}z_1+u_{10}z_2)+u_{11}^2(u_{10}z_0)},\\
v_{t2} &= (g_4u_{t2}+u_{t1}+g_2)(s_2^{'}/r)+v_{12}\\
&= \frac{(u_{11}z_0+u_{10}z_1)(u_2^{'}+u_{12}^2)}{u_{10}^2}+z_2+k_1+\frac{r}{s_2^{'}}+v_{12},\\
v_{t1} &= (g_4u_{t1}+u_{t0}+g_1)(s_2^{'}/r)+v_{11}\\
&= \frac{k_2[u_{12}(u_{11}z_0+u_{10}z_1)+u_{10}z_0]+u_{12}^2(u_{10}z_0)}{u_{10}^2}+k_1(\\
&\quad k_1+\frac{r}{s_2^{'}})+(k_2+u_{12}k_1+\frac{rf_6}{s_2^{'}})+(z_1+u_{12}z_2)+v_{11},
\end{aligned}$$

$$
\begin{aligned}
v_{t0} &= (g_4 u_{t0} + g_0)(s_2^{'}/r) + v_{10} \\
&= \frac{(u_{11}z_0 + u_{10}z_1)(u_0^{'} + u_{11}^2)}{u_{10}^2} + (z_0 + u_{12}z_1 \\
&\quad + u_{11}z_2) + v_{10}.
\end{aligned}
$$

And since $F + hV_1 + V_1^2 = U_1 Z + U_1^2(X + f_6)$, we also have

$$
\begin{aligned}
f_0 + v_{10}^2 &= u_{10}z_0 + f_6 u_{10}^2, \\
f_1 &= u_{11}z_0 + u_{10}z_1 + u_{10}^2, \\
f_2 + v_{11}^2 &= u_{12}z_0 + u_{11}z_1 + u_{10}z_2 + f_6 u_{11}^2, \\
v_{10} &= z_0 + u_{12}z_1 + u_{11}z_2 + u_{11}^2, \\
v_{11} + v_{12}^2 &= z_1 + u_{12}z_2 + f_6 u_{12}^2, \\
v_{12} &= u_{12}^2 + z_2.
\end{aligned}
$$

Using the equations above, we can calculate $k_1, k_2, \frac{r}{s_2^{'}}$ and $v_{t3}, v_{t2}, v_{t1}, v_{t0}$ cheaper:

$$
\begin{aligned}
k_1 &= \frac{u_{10}^2(f_1 + u_{10}^2)}{u_{10}^2(f_2 + v_{11}^2 + f_6 u_{11}^2) + u_{11}^2(f_0 + v_{10}^2 + f_6 u_{10}^2)}, \\
k_2 &= \frac{u_{10}^2[u_{12}(f_1 + u_{10}^2) + (f_0 + v_{10}^2 + f_6 u_{10}^2)]}{u_{10}^2(f_2 + v_{11}^2 + f_6 u_{11}^2) + u_{11}^2(f_0 + v_{10}^2 + f_6 u_{10}^2)}, \\
\frac{r}{s_2^{'}} &= \frac{u_{10}^4}{u_{10}^2(f_2 + v_{11}^2 + f_6 u_{11}^2) + u_{11}^2(f_0 + v_{10}^2 + f_6 u_{10}^2)}, \\
v_{t3} &= \frac{f_0 + v_{10}^2 + f_6 u_{10}^2}{u_{10}^2} \\
&\quad + \frac{(f_1 + u_{10}^2)^2}{u_{10}^2(f_2 + v_{11}^2 + f_6 u_{11}^2) + u_{11}^2(f_0 + v_{10}^2 + f_6 u_{10}^2)}, \\
v_{t2} &= \frac{(f_1 + u_{10}^2)(u_2^{'} + u_{12}^2)}{u_{10}^2} + k_1 + \frac{r}{s_2^{'}} + u_{12}^2, \\
v_{t1} &= \frac{1}{u_{10}^2}\{k_2[u_2(f_1 + u_{10}^2) + (f_0 + v_{10}^2 + f_6 u_{10}^2)] \\
&\quad + u_{12}^2(f_0 + v_{10}^2 + f_6 u_{10}^2)\} + k_1(k_1 + \frac{r}{s_2^{'}}) \\
&\quad + (k_2 + u_{12}k_1 + \frac{rf_6}{s_2^{'}}) + (v_{12}^2 + f_6 u_{12}^2), \\
v_{t0} &= \frac{(f_1 + u_{10}^2)(u_0^{'} + u_{11}^2)}{u_{10}^2} + u_{11}^2.
\end{aligned}
$$

We note that $r \cdot s_2^{'} = u_{10}^2 \cdot [u_{10}^2 \cdot (u_{12}z_0 + u_{11}z_1 + u_{10}z_2) + u_{11}^2 \cdot (u_{10}z_0)] = u_{10}^2 \cdot [u_{10}^2 \cdot (f_2 + v_{11}^2 + f_6 u_{11}^2) + u_{11}^2 \cdot (f_0 + v_{10}^2 + f_6 u_{10}^2)]$. Therefore, we can calculate $rs_2^{'}$ cheaply without knowing the exact coefficients of Z. We present the explicit formula for this case in Table XX. The corresponding addition formula which requires $1I + 60M + 5S$ to add two reduced divisor classes is showed in Table XXI in the appendix. In [45], Guyot *et al.* discuss two types of curves with $h_2 = 0$ and $f_6 = 0$, respectively. Their doubling formulae cost $1I + 63M + 9S$ and $1I + 64M + 5S$ for these two different cases. We note that using special curves with $h(x) = h_3x^3$ can lead to the fast computation of a divisor class doubling. We derive the new explicit doubling formula which needs only $1I + 30M + 11S$. In addition, there are four counted multiplications with f_6 which can be computed cheaply when f_6 is 'small' in the formulae.

E. Summary

Depending on the degree of h, we have derived the corresponding explicit formulae which can compute the divisor class doubling fast in subsections above. For h of degree 0 and 1 the case f_6 not small does not apply since we make it zero by isomorphic transformations. We also find the fast doubling formulae for the special curves when the degree of h is 2 and 3. Table IV presents a summary of our work, as well as the previous work that has been done on improving the Harley's algorithm for genus 3 HECC. It can be seen clearly that we obtain the fastest explicit doubling formula which requires only $1I + 10M + 11S$ for the case of $h(X) = 1$, and improve the recent results in [45] significantly.

VII. Inversion-Free Arithmetic on Genus 3 HEC

In this section we discuss the inversion-free coordinate system and restrict our attentions only on the most frequent case. Our findings are based on the fastest explicit formulae showed in Table XVII and Table XVIII (for odd characteristic case), and Table XXV and Table XXVI (for even characteristic case), respectively. We generalize the idea proposed in [68], [82] to genus 3 curves. In addition, we also consider a group addition with mixed coordinates: one of the input divisor classes is with the affine representation, the other projective representation and the output divisor class is represented by the projective coordinates. The idea of using mixed coordinates for the addition algorithm was first proposed for elliptic curves in [20] and then generalized to genus 2 curves in [68].

For genus 3 HECs, the divisor class is denoted by $[U, V] = [X^3 + u_2X^2 + u_1X + u_0, v_2X^2 + v_1X + v_0]$ with $U \mid (V^2 + hV - F)$. When computing the divisor class addition or doubling in the affine coordinate system, one inversion is required. In order to avoid this inversion, we introduce an additional coordinate Z to collect the common denominator of the usual six coordinates and let the septuple $[U_2, U_1, U_0, V_2, V_1, V_0, Z]$ stand for $[X^3 + (U_2/Z)X^2 + (U_1/Z)X + (U_0/Z), (V_2/Z)X^2 + (V_1/Z)X + (V_0/Z)]$. If the output of a scalar multiplication is with the affine representation we require one inversion and six multiplications to execute the coordinate transformations for the output divisor class at the end of the computation.

We now proceed to investigate the inversion-free arithmetic for the most frequent case. In the practical applications, inversion-free group operations are useful not only for improving the performance of genus 3 HECC in the embedded processors where the field inversion is much slower than the field multiplication but also for accelerating the hardware implementation of genus 3 HECC. The simplification for the group operations in the projective coordinate system can be achieved by applying the methods described in Section IV.

A. Inversion-Free Addition Formulae

In this subsection, we give explicit formula for adding two reduced divisor classes in the projective coordinate system. Our formula can also be used for affine inputs if we regard $[U_1, V_1]$ as the septuple $[u_{12}, u_{11}, u_{10}, v_{12}, v_{11}, v_{10}, 1]$. Table XXVII and Table XXVIII list the number of field operations required to finish each step for a group addition on genus 3 HECs defined over prime fields and binary fields, respectively.

TABLE IV
SPEEDING UP GROUP OPERATION ON A HEC OF GENUS THREE USING HARLEY'S ALGORITHM

Reference	Finite	Curve	Cost	
	Field	Properties	Addition	Doubling
Kuroki *et al.* [64]	$\mathbb{F}_p$	$h(X) = 0, f_6 = 0$	$1I + 81M/S$	$1I + 74M/S$
Gonda *et al.* [43]	$\mathbb{F}_p$	$h(X) = 0, f_6 = 0$	$1I + 70M/S$	$1I + 71M/S$
Thomas *et al.* [111]	general	$h_i \in \mathbb{F}_2, f_6 = 0$	$1I + 70M + 6S$	$1I + 62M + 10S$
	$\mathbb{F}_{2^n}$	$h_i \in \mathbb{F}_2, f_6 = 0$	$1I + 65M + 6S$	$1I + 53M + 10S$
	$\mathbb{F}_{2^n}$	$h(X) = 1, f_6 = 0$	$1I + 65M + 6S$	$1I + 14M + 11S$
Guyot *et al.* [45]	$\mathbb{F}_p$	$h(X) = 0, f_6 = 0$	$1I + 64M + 6S$	$1I + 61M + 9S$
	$\mathbb{F}_{2^n}$	deg $h = 3, h_2 = 0$	$1I + 62M + 5S$	$1I + 63M + 9S$
	$\mathbb{F}_{2^n}$	deg $h = 3, f_6 = 0$	$1I + 64M + 4S$	$1I + 64M + 5S$
	$\mathbb{F}_{2^n}$	deg $h = 2, f_6 = 0$	$1I + 60M + 6S$	$1I + 52M + 8S$
	$\mathbb{F}_{2^n}$	deg $h = 1, h_0 = 0$	$1I + 58M + 6S$	$1I + 44M + 6S$
	$\mathbb{F}_{2^n}$	$h(X) = 1, f_6 = 0$	$1I + 58M + 6S$	$1I + 11M + 11S$
Avanzi *et al.* [3]	$\mathbb{F}_{2^n}$	$h(X) = 1$	$1I + 57M + 6S$ (classical)	$1I + 11M + 11S$ (classical)
			$1I + 47.7M + 6S$ (effective)	$1I + 9.3M + 11S$ (effective)
Nyukai *et al.* [86]	$\mathbb{F}_p$	$h(X) = 0, f_6 = 0$	$1I + 67M/S$	$1I + 68M/S$
	$\mathbb{F}_{2^n}$	deg $h = 3, f_6 = 0$	$1I + 63M + 4S$	$1I + 63M + 5S$
Our work	$\mathbb{F}_{2^n}$	$h(X) = X^3$	$1I + 60M + 5S$	**1I + 26M + 11S** (f_6 sma.)
		$f_5 = f_4 = f_3 = 0$	Table XIX	$1I + 30M + 11S$ (f_6 arb.)
				Table XX
	$\mathbb{F}_{2^n}$	$h(X) = h_2X^2$	$1I + 58M + 6S$ (h_2 sma.)	**1I + 20M + 12S** ($h_2 = 1$, f_6 sma.)
		$f_5 = f_3 = f_2 = 0$	$1I + 60M + 5S$ (h_2 arb.)	$1I + 24M + 12S$ ($h_2 = 1$, f_6 arb.)
			Table XXI	$1I + 28M + 10S$ (h_2^{-1} sma., f_6 sma.)
				$1I + 32M + 10S$ (h_2^{-1} sma., f_6 arb.)
				$1I + 32M + 10S$ (h_2 arb., f_6 sma.)
				$1I + 36M + 10S$ (h_2 arb., f_6 arb.)
				Table XXII
	$\mathbb{F}_{2^n}$	$h(X) = h_1X$	$1I + 57M + 6S$ (h_1 sma.)	**1I + 13M + 13S** ($h_1 = 1$)
		$f_6 = f_4 = f_1 = 0$	$1I + 58M + 6S$ (h_1 arb.)	$1I + 16M + 12S$ (h_1^{-1} sma.)
			Table XXIII	$1I + 20M + 12S$ (h_1 arb.)
				Table XXIV
	$\mathbb{F}_{2^n}$	$h(X) = h_0$	$1I + 57M + 6S$	**1I + 10M + 11S** ($h_0 = 1$)
		$f_6 = f_5 = f_4 = f_2 = 0$	Table XXV	$1I + 11M + 11S$ (h_0^{-1} sma.)
				$1I + 15M + 11S$ (h_0 arb.)
				Table XXVI

B. Inversion-Free Mixed Addition Formulae

In this subsection, we present inversion-free mixed addition formulae which take a reduced affine divisor class and a reduced projective divisor class as the inputs and a reduced projective divisor class as the output. This kind of formula has been widely used in many scalar multiplication algorithms such as (signed) double-and-add, NAF and so on. When using these scalar multiplication algorithms, one of the inputs is the base divisor class in affine representation and the intermediate result in projective representation. We can see clearly that this kind of addition formula can do better than the algorithm in section VII-A. Table XXIX and Table XXX list the number of field operations required to perform the respective steps for a group mixed addition on genus 3 HECs defined over prime fields and binary fields, respectively. For the case that genus 3 HECs are defined over a binary field, Toom's multiplication can not be used again because some steps of Toom algorithm need the computations of dividing by 2 [43]. we use Karatsuba's multiplication instead of Toom's one and derive the corresponding explicit formula.

Using the formula in Table XXIX, one can save $19M + 1S$ compared to the general addition formula for genus 3 HECs over prime fields. Therefore, it is more efficient to use mixed addition formula to compute the scalar multiplication. For special genus 3 curves defined over binary fields with $h(X) = 1$, our formula can save $23M$ at the cost of only $2S$.

C. Inversion-Free Doubling Formulae

In this subsection, the inversion-free doubling formulae are given. The input for the doubling algorithm is in projective representation for most of cases. Table XXXI and XXXIII list the number of field operations for a group doubling on genus 3 HECs defined over prime fields and binary fields, respectively. We also optimize the inversion-free doubling formulae using the affine input, see Table XXXII and Table XXXIV. These formulae can be useful to get another small speedup when used in applications where area or code size is not an issue.

TABLE VI
TIMINGS OF THE FIELD LIBRARY AND THE CORRESPONDING *MI*-RATIOS

Field	Multiplication	Squaring	Inversion	*MI*-ratio
$\mathbb{F}_{2^{59}}$	$0.33\mu s$	$0.30\mu s$	$1.7\mu s$	5.15
$\mathbb{F}_{2^{61}}$	$0.34\mu s$	$0.31\mu s$	$2.0\mu s$	5.88
$\mathbb{F}_{2^{63}}$	$0.36\mu s$	$0.32\mu s$	$2.1\mu s$	5.83

D. Summary

This is the first contribution that presents the inversion-free group operations for genus 3 HECs defined over both prime fields and binary fields. In order to minimize the number of operations, we do not keep the additional coordinate Z_3 (for the addition and mixed addition), Z_2 (for the doubling) and Z (for the affine doubling) minimal. We respectively take $Z_3 = r^3 s_2^{'5} Z^8$, $Z_3 = r^3 s_2^{'5} Z_1^8$, $Z_2 = 8r^3 s_2^{'5} Z^8$, and $Z = 8r^3 s_2^{'5}$ for the addition, mixed addition, doubling and affine doubling formula when genus 3 HECs are defined over $\mathbb{F}_p$. For genus 3 HECs over $\mathbb{F}_{2^n}$ with $h(X) = 1$, we take $Z_3 = r^3 s_2^{'5}(Z_1 Z_2)^8$, $Z_3 = r^3 s_2^{'5}(Z_1 Z_2)^5$, $Z_2 = u_2^8 Z_1^{14}$ and $Z = u_2^5$ for the addition, mixed addition, doubling and affine doubling formula, respectively. Besides the output results have to be adjusted to have the same denominator. We summarize the computational complexity of the inversion-free group operations in Table V.

VIII. Implementation Results

This section introduces our implementations of the efficient doubling algorithms. In order to test the performance of the proposed explicit formulae, we chose three binary fields to implement genus 3 HECC. Due to the attack proposed by Thériault [104], we should select at least 56-bit finite fields in order to obtain the same security as a 160-bit elliptic curve cryptosystem. We used the binary fields $\mathbb{F}_{2^{59}}$, $\mathbb{F}_{2^{61}}$ and $\mathbb{F}_{2^{63}}$. For $\mathbb{F}_{2^{59}}$ and $\mathbb{F}_{2^{61}}$, we used the minimal weight irreducible pentanomial $x^{59}+x^7+x^4+x^2+1$ and $x^{61}+x^5+x^2+x+1$ to construct finite fields, respectively. However, for $\mathbb{F}_{2^{63}}$, we utilized the minimal weight irreducible trinomial $x^{63}+x+1$ as the field extension. Efficient algorithms summarized in [88] were used to perform the arithmetics over binary fields. Although we used the composite field $\mathbb{F}_{2^{63}}$, the implementation methods *do not use* the composite filed structure. All the algorithms are implemented on a Pentium-4 @2.8GHz processor and with C programming language. Table VI shows the timings of the finite field library and the corresponding *MI*-ratio (the ratio of the timing of one inversion to one multiplication).

We noted that the *MI*-ratio is small on the Pentium-4 processor. In our newly derived inversion-free group operations, at least 30 additional multiplications are needed to save the one remaining inversion. Therefore, using the inversion-free arithmetic cannot promote the performance of genus 3 HECC on this processor. However, for some embedded processors where the inversions are extremely time and space critical, the inversion-free explicit formulae will be much useful. Based on the analysis of the performance of the field library, we decided not to implement the inversion-free group operations on the Pentium-4 processor.

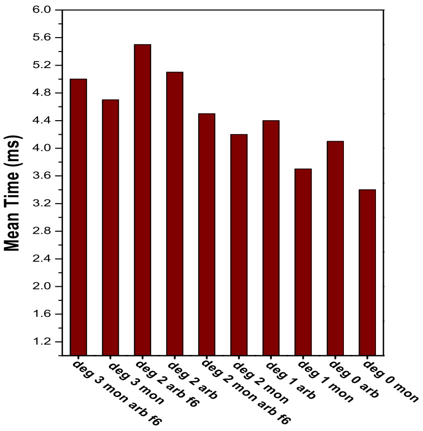

Fig. 1. Timings on $\mathbb{F}_{2^n}$, $n = 59$

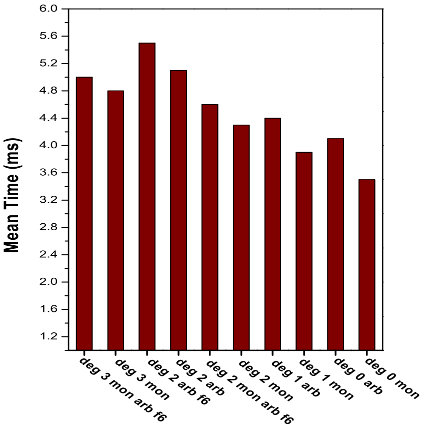

Fig. 2. Timings on $\mathbb{F}_{2^n}$, $n = 61$

We implemented genus 3 HECC over three binary fields based on our efficient doubling algorithms. We utilized NAF method [10] for the scalar multiplication and 160-bit random integers as the scalars. Each timing shows the average of every 1,000,000 operations on a genus 3 HEC generated randomly with $F(X)$ to be irreducible. The experimental results were depicted in three bar graphs Fig. 1 $\sim$ Fig. 3.

In the graphs Fig. 1 $\sim$ Fig. 3, we include the following ten cases respectively:

1) deg 3 mon arb f_6: The case where deg $h = 3$, $h_2 = h_1 = h_0 = 0$, $f_6 \neq 0$ and $h_3 = 1$;
2) deg 3 mon: The case where deg $h = 3$, $h_2 = h_1 = h_0 = 0$, $f_6 = 0$ and $h_3 = 1$;
3) deg 2 arb f_6: The case where deg $h = 2$, $h_1 = h_0 = 0$, $f_6 \neq 0$;
4) deg 2 arb: The case where deg $h = 2$, $h_1 = h_0 = 0$, $f_6 = 0$;
5) deg 2 mon arb f_6: The case where deg $h = 2$, $h_1 = h_0 = 0$, $f_6 \neq 0$ and $h_2 = 1$;

TABLE V
INVERSION-FREE GROUP OPERATIONS FOR GENUS 3 HEC

Finite Field	Curve Properties	Addition	Doubling	Mixed Addition	Affine Doubling
$\mathbb{F}_p$	$h(X) = 0, f_6 = 0$	$123M + 7S$ Table XXVII	$107M + 10S$ Table XXXI	$104M + 6S$ Table XXIX	$86M + 6S$ Table XXXII
$\mathbb{F}_{2^n}$	$h(X) = 1, f_6 = f_5 = f_4 = f_2 = 0$	$116M + 8S$ Table XXVIII	$37M + 16S$ Table XXXIII	$93M + 10S$ Table XXX	$23M + 11S$ Table XXXIV

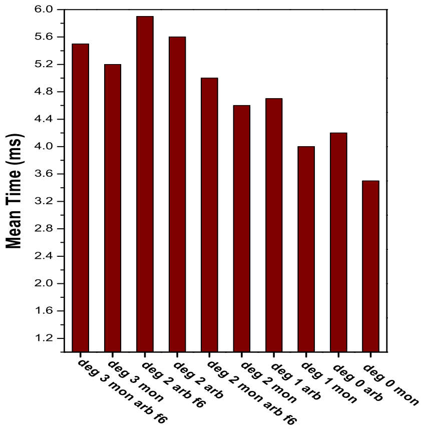

Fig. 3. Timings on $\mathbb{F}_{2^n}$, $n = 63$

6) deg 2 mon: The case where deg $h = 2$, $h_1 = h_0 = 0$, $f_6 = 0$ and $h_2 = 1$;
7) deg 1 arb: The case where deg $h = 1, h_0 = 0$;
8) deg 1 mon: The case where deg $h = 1, h_0 = 0$ and $h_1 = 1$;
9) deg 0 arb: The case where deg $h = 0$;
10) deg 0 mon: The case where deg $h = 0$ and $h_0 = 1$.

IX. Conclusion

Our contribution is another step towards the efficient implementation of the group operations for genus 3 HECs. We showed how to improve and optimize the Harley's algorithm from three aspects: using the Theta divisors, optimizing explicit doubling formulae for genus 3 curves over binary fields and employing inversion-free arithmetic.

Our work starts in Section IV with a thorough summarization of all kinds of tricks used to derive the explicit formulae from Cantor's algorithm. In Section V, we further simplify Katagi *et al.*'s explicit formulae with theta divisors for four cases by saving about $20\% \sim 50\%$ of the multiplications for genus 3 HECs over binary fields. In addition, we generalize their idea to the prime fields case. In Section VI, we move to the issues of finding efficient doubling algorithms for genus 3 HECs over binary fields. By constructing birational transformations of variables, we obtain four families of curves over which the divisor class doubling is extremely efficient. Especially, for the case of $h(X) = 1$, we obtain the fastest explicit doubling formula which needs only $1I + 10M + 11S$. While for the case of deg $h = 1$ our explicit formula improves the recent result in [45] significantly by saving $31M$ at the cost of extra $7S$. Furthermore, the implementations of our new derived explicit formulae show the excellent performance on a Pentium-4 processor. In Section VII, we switch our attentions to the projective coordinate system. We present the inversion-free addition, mixed addition, doubling and affine doubling explicit formulae, respectively, for genus 3 HECs defined over both prime fields and binary fields. These formulae will be suited for applications on embedded processors and in constrained environments.

The improvement of our algorithms by using generalized weighted projective coordinates is the next logical step to promote the performance of genus 3 HECC. Furthermore, the practical performance of our inversion-free explicit formulae in embedded systems such as ARMs, DSPs and smart cards needs further studying.

Acknowledgment

Xinxin Fan would like to thank Prof. Yumin Wang from Xidian University, China, for everything he did when the first author is a master candidate in the State Key Lab of Integrated Service Networks.

References

[1] L. Adlemann, J. DeMarrais, and M.-D. Huang, "A Subexponential Algorithm for Discrete Logarithm over the Rational Subgroup of the Jacobians of Large Genus Hyperelliptic Curves over Finite Fields," *Algorithmic Number Theory, First International Symposium, ANTS-I*, ser. LNCS 877, L. Adlemann and M.-D. Huang, Eds. Berlin, Germany: Springer-Verlag, pp. 28-40, 1994.

[2] R. M. Avanzi, "Aspects of Hyperelliptic Curves over Large Prime Fields in Software Implementations," *Workshop on Cryptographic Hardware and Embedded Systems - CHES 2004*, ser. LNCS 3156, M. Joye and J.-J. Quisquater, Eds. Berlin, Germany: Springer-Verlag, pp. 148-162, 2004.

[3] R. M. Avanzi, N. Thériault, and Z. Wang, "Rethinking Low Genus Hyperelliptic Jacobian Arithmetic over Binary Fields: Interplay of Field Arithmetic and Explicit Formulae," *Centre for Applied Cryptographic Research (CACR) Technical Reports*, CACR 2006-07, available at http://www.cacr.math.uwaterloo.ca/.

[4] S. Baktir, J. Pelzl, T. Wollinger, B. Sunar, and C. Paar, "Optimal Tower Fields for Hyperelliptic Curve Cryptosystems," *38th Asilomar Conference on Signals, Systems and Computers*, November 7-10, 2004, Pacific Grove, USA.

[5] P. S. L. M. Barreto, S. Galbraith, C. O. hEigeartaigh, and M. Scott, "Efficient Pairing Computation on Supersingular Abelian Varities", Cryptology ePrint Archive, Report 2004/375, 2004, http://eprint.iacr.org.

[6] E. Barteska, J. Pelzl, C. Paar, V. Wittelsberger and T. Wollinger, "Case Study: Compiler Comparison for an Embedded Cryptographical Application", *The 2004 International Conference on Embedded Systems and Applications - ESA*, CSREA Press, pp. 589-595, 2004.

[7] L. Batina, D. Hwang, A. Hodjat, B. Preneel, and I. Verbauwhede, "Hardware/Software Co-design for Hyperelliptic Curve Cryptography (HECC) on the 8051 microprocessor," *Cryptographic Hardware and Embedded Systems - CHES 2005*, ser. LNCS 3659, J. R. Rao, B. Sunar, Eds. Berlin, Germany: Springer-Verlag, pp. 106-118, 2005.

[8] G. Bertoni, L. Breveglieri, T. Wollinger, and C. Paar, “Hyperelliptic Curve Cryptosystem: What is the Best Parallel Hardware Architecture?,” chapter in *Embedded Cryptographic Hardware: Design and Security*, Nadia Nedjah Ed. Nova Science Publishers, NY, USA, 2004.

[9] G. Bertoni, L. Breveglieri, T. Wollinger and C. Paar, “Finding Optimum Parallel Coprocessor Design for Genus 2 Hyperelliptic Curve Cryptosystems”, *International Conference on Information Technology: Coding and Computing - ITCC 2004*, IEEE Computer Society, pp. 538-544, 2004.

[10] I. Blake, G. Seroussi, and N. Smart, *Elliptic Curves in Cryptography*, London Mathematical Society Lecture Note Series, No.265. The Edingburgh Building, Cambridge CB2 2RU, UK: Cambridge University Press, 1999.

[11] N. Boston, T. Clancy, Y. Liow, and J. Webster, “Genus Two Hyperelliptic Curve Coprocessor,” *Cryptographic Hardware and Embedded Systems - CHES 2002*, ser. LNCS 2523, B. S. Kaliski, Ç. K. Koç, and C. Paar, Eds. Berlin, Germany: Springer-Verlag, pp. 529-539, 2002. Updated version available at http://www.cs. umd.edu/ clancy/docs/hec-ches2002.pdf.

[12] B. Byramjee, and S. Duquesne, “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.

[13] D. Cantor, “Computing in Jacobian of a Hyperelliptic Curve,” *Mathematics of Computation*, vol. 48 (177), pp. 95-101, January 1987.

[14] Y. Choie, J. Kim, and E. Lee, “Efficient Computation of the Tate Pairing on Hyperelliptic Curves for Cryptosystems,” Cryptology ePrint Archive, Report 2005/167, 2005, http://eprint.iacr.org.

[15] Y. Choie, and E. Lee, “Implementation of Tate Pairing on Hyperelliptic Curves of Genus 2,” *The 6th International Conference on Information Security and Cryptology - ICISC 2003*, ser. LNCS 2971, J. I. Lim, D. H. Lee, Eds. Berlin, Germany: Springer-Verlag, pp. 97-111, 2003.

[16] Y. Choie, and E. Lee, “Tate pairing computation on the divisors of hyperelliptic curves for cryptosystems,” Cryptology ePrint Archive, Report 2005/166, 2005, http://eprint.iacr.org.

[17] T. Clancy, “Analysis of FPGA-based Hyperelliptic Curve Cryptosystems,” Master’s thesis, University of Illinois Urbana-Champaign, December 2002.

[18] H. Cohen, *A Course in Computational Algebraic Number Theory*, ser. Graduate Texts in Math. 138. Berlin, Germany: Springer-Verlag, 1993, fourth corrected printing, 2000.

[19] H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen and F. Vercauteren, *Handbook of Elliptic and Hyperelliptic Curve Cryptography*, Boca Raton, Florida, USA: Chapman & Hall/CRC, 2006.

[20] H. Cohen, A. Miyaji, and T. Ono, “Efficient Elliptic Curve Exponentiation Using Mixed Coordinates,” *Advance in Cryptology - ASIACRYPT 1998*, ser. LNCS 1514, K. Ohta, and D. Pei, Eds. Berlin, Germany: Springer-Verlag, pp. 51-65, 1998.

[21] W. Diffie and M. E. Hellman, “New Directions in Cryptography,” *IEEE Transactions on Information Theory*, vol. IT-22, pp. 644-654, 1976.

[22] S. Duquesne, “Montgomery Scalar Multiplication for Genus 2 Curves,” *Algorithm Number Theory Symposium - ANTS VI*, ser. LNCS 3076, D. Buell, Ed. Berlin, Germany: Springer-Verlag, pp. 153-168, 2004.

[23] I. Duursma, and H. S. Lee, “Tate Pairing Implementation for Hyperelliptic Curves $y^2 = x^p - x + d$,” *Advances in Cryptology - ASIACRYPT 2003*, ser. LNCS 2894, C. S. Laih, Ed. Berlin, Germany: Springer-Verlag, pp. 111-123, 2003.

[24] K. Eisenträger, K. Lauter, and P. L. Montgomery, “Improved Weil and Tate Pairings for Elliptic and Hyperelliptic Curves”, *Advances in Cryptology - ASIACRYPT 2003*, ser. LNCS 3076, D. Buell, Ed. Berlin, Germany: Springer-Verlag, pp. 169-183, 2004.

[25] T. ElGamal, “A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms,” *IEEE Transactions on Information Theory*, vol. IT-31, pp. 469-472, 1985.

[26] G. Elias, A. Miri, and T. H. Yeap. “High-Performance, FPGA-Based Hyperelliptic Curve Cryptosystems,” *The Proceeding of the 22nd Biennial Symposium on Communications*, May 2004, Queen’s University, Kingston, Ontario, Canada.

[27] A. Enge, and G. Gaudry, “A General Framework for Subexponential Discrete Logarithm Algorithms,” *Acta Arith*, vol. 102, pp. 83-103, 2002.

[28] X. Fan, “Research on Fast Algorithms for Hyperelliptic Curve Cryptosystem”, Master’s thesis, State Key Lab of Integrated Service Networks, Xidian University, Xi’an, Shannxi, P.R. China, 2005. In Chinese.

[29] X. Fan, and Y. Wang, “Simultaneous Divisor Class Addition-Subtraction Algorithm and Its Applications to Hyperelliptic Curve Cryptosystem”, *The IEEE 19th International Conference on Advanced Information Networking and Applications - AINA 2005*, IEEE Computer Society, pp. 978-983, 2005.

[30] X. Fan, T. Wollinger, and Y. Wang, “Inversion-Free Arithmetic on Genus 3 Hyperelliptic Curves and Its Implementations”, *International Conference on Information Technology: Coding and Computing - ITCC 2005*, IEEE Computer Society, pp. 642-647, 2005.

[31] X. Fan, T. Wollinger, and Y. Wang, “Efficient Doubling on Genus 3 Curves over Binary Fields,” *The Cryptographers’ Track at the RSA Conference - CT-RSA 2006*, ser. LNCS 3860, D. Pointcheval, Ed. Berlin, Germany: Springer-Verlag, pp. 64-81, 2006.

[32] FIPS 186, “Digital Signature Standard,” *Federal Information Processing Standards Publication 186*, U.S. Department of Commerce/ N.I.S.T. National Technical Infromation Service, 1994.

[33] R. Flassenberg, and S. Paulus, “Sieving in Fuction Fileds,” *Experimental Mathematics*, iss. 4, no. 8, pp. 339-349, 1999.

[34] 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*, vol. 62, no. 206, pp. 865-874, 1994.

[35] S. Galbraith, “Supersingluar Curves in Cryptography,” *Advance in Cryptology - ASIACRYPT ’01*, ser. LNCS 2248, C. Boyd, Ed. Berlin, Germany: Springer-Verlag, pp. 495-517, 2001.

[36] S. Galbraith, “Pairings on Hyperelliptic Curves,” Talk at the 9th workshop on Elliptic Curve Cryptography (ECC 2005), 2005, slide available at http://www.cacr.math.uwaterloo.ca/.

[37] J. von zur Gathen, and J. Gerhard, *Mordern Computer Algebra*, ser. Graduate Texts in Math. 138. The Edingburgh Building, Cambridge CB2 2RU, UK: Cambridge University Press, 1999.

[38] P. Gaudry, “An Algorithm for Solving the Discrete Log Problem on Hyperelliptic Curves,” *Advance in Cryptology - EUROCRYPT 2000*, ser. LNCS 1807, B. Preneel, Ed. Berlin, Germany: Springer-Verlag, pp. 19-34, 2000.

[39] P. Gaudry and R. Harley, “Counting Points on Hyperelliptic Curves over Finite Fields,” *Algorithm Number Theory Symposium - ANTS IV*, ser. LNCS 1838, W. Bosma, Ed. Berlin, Germany: Springer-Verlag, pp. 297-312, 2000.

[40] P. Gaudry, F. Hess, and N. P. Smart, “Constructive and Destructive Facets of Weil Descent on Elliptic Curves,” *Journal of Cryptology*, vol. 15, no. 1, pp. 19-46, 2002.

[41] P. Gaudry, E. Thomé, N. Thériault, and C. Diem, “A Double Large Prime Variation for Small Genus Hyperelliptic Index Calculus,” Cryptology ePrint Archive, Report 2004/153, 2004, http://eprint.iacr.org.

[42] M. Gonda, K. Matsuo, K. Aoki, J. Chao, and S. Tsujii, “Improvements of Addition Algorithm on Genus 3 Hyperelliptic Curves and Their Implementations,” *The 2004 Symposium on Cryptography and Information Security - SCIS 2004*, IEICE Janpan, January 2004.

[43] M. Gonda, K. Matsuo, K. Aoki, J. Chao, and S. Tsujii, “Improvements of Addition Algorithm on Genus 3 Hyperelliptic Curves and Their Implementation,” *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Science*, vol. E88-A NO.1, pp. 89-96, January 2005.

[44] G. Günter, T. Lange, and A. Stein, “Speeding Up the Arithmetic on Koblitz Curves of Genus Two,” *Seventh Annual Workshop on Selected Areas in Cryptography - SAC 2000*, ser. LNCS 2012, D. R. Stinson, S. Tavares, Eds. Berlin, Germany: Springer-Verlag, pp. 106-117, 2000.

[45] C. Guyot, K. Kaveh, and V. M. Patankar, “Explicit Algorithm for the Arithmetic on the Hyperelliptic Jacobians of Genus 3,” *Journal of the Ramanujan Mathematical Society*, 19, no. 2, pp. 75-115, 2004.

[46] R. Harasawa, Y. Sueyoshi, and Aichi Kudo, “Tate pairing for $y^2 = x^5 - \alpha x$ in Characteristic Five,” Cryptology ePrint Archive, Report 2006/114, 2006, http://eprint.iacr.org.

[47] R. Harley, “Fast Arithmetic on Genus Two Curves,” http://cristal. inria. fr/harley/hyper/, adding.txt and doubling.c, 2000.

[48] C. O. hEigeartaigh, and M. Scott, “Pairing Calculation on Supersingular Genus 2 Curves”, Cryptology ePrint Archive, Report 2006/005, 2006, http://eprint.iacr.org.

[49] F. Hess, “The GHS Attack Revisited,” *Advance in Cryptology - EUROCRYPT 2003*, ser. LNCS 2656, E. Biham, Ed. Berlin, Germany: Springer-Verlag, pp. 374-387, 2003.

[50] A. Hodjat, L. Batina, D. Hwang, and I. Verbauwhede, “A Hyperelliptic Curve Cryto Coprocessor for an 8051 Microcontroller,” *IEEE Workshop on Signal Processing Systems (SIPS 2005)*, IEEE Computer Society, pp. 93-98, November 2005.

[51] A. Hodjat, L. Batina, D. Hwang, and I. Verbauwhede, “HW/SW Co-Design of a Hyperelliptic Curve Cryptosystem using a Microcode Instruction Set Coprocessor,” to appear in *Elsevier Integration, the VLSI Journal, special issue on Embedded Cryptographic Hardware*, 2006.

[52] A. Karatsuba, and Y. Ofman, “Multiplication of Multidigit Numbers on Automata,” *Soviet Physics Doklady. (English Translation)*, vol. 7, no. 7, pp. 595-596, 1963.

[53] M. Katagi, I. Kitamura, T. Akishita, and T. Takagi, "Novel Efficient Implementations of Hyperelliptic Curve Cryptosystems using Degenerate Divisors", *Workshop on Information Security Applications - WISA 2004*, ser. LNCS 3325, C. H. Lim, M. Yung, Eds. Berlin, Germany: Springer-Verlag, pp. 345-359, 2005.

[54] M. Katagi, T. Akishita, I. Kitamura, and T. Takagi, "Some Improved Algorithms for Hyperelliptic Curve Cryptosystems using Degenerate Divisors", *Seventh Annual International Conference on Information Security and Cryptology - ICISC 2004*, ser. LNCS 3506, C. Park, S. Chee, Eds. Berlin, Germany: Springer-Verlag, pp. 296-312, 2005.

[55] M. Katagi, T. Akishita, I. Kitamura, and T. Takagi, "Efficient Hyperelliptic Curve Cryptosystems Using Theta Divisors", *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Science*, vol. E89-A NO.1, pp. 151-160, January 2006.

[56] H. Kim, T. Wollinger, Y. Choi, K. Chung, and C. Paar, "Hyperelliptic Curve Coprocessors on a FPGA," *Workshop on Information Security Applications - WISA 2004*, ser. LNCS 3325, C. H. Lim, M. Yung, Eds. Berlin, Germany: Springer-Verlag, pp. 360-374, 2005.

[57] I. Kitamura and M. Katagi, "Efficient Implementation of Genus Three Hyperelliptic Curve Cryptography over $GF(2^n)$," Cryptology ePrint Archive, Report 2003/248, 2003, http://eprint.iacr.org.

[58] I. Kitamura, M. Katagi, and T. Takagi, "A Complete Divisor Class Halving Algorithm for Hyperelliptic Curve Cryptosystems of Genus Two", *Tenth Australasian Conference on Information Security and Privacy - ACISP 2005*, ser. LNCS 3574, C. Boyd, J. M. G. Nieto, Eds. Berlin, Germany: Springer-Verlag, pp. 146-157, 2005.

[59] D. E. Knuth, *The Art of Computer Programming. Volume 2: Seminumerical Algorithm, 3rd ed.*, Reading, Massachusetts, USA: Addison-Wesley, 1997.

[60] N. Koblitz, "Elliptic Curve Cryptosystem," *Mathematics of Computation*, vol. 48, pp. 203-209, 1987.

[61] N. Koblitz, "A Family of Jacobian Suitable for Discrete Log Cryptosystems," *Advance in Cryptology - CRYPTO'88*, ser. LNCS 403, Shafi Goldwasser, Ed. Berlin, Germany: Springer-Verlag, pp. 94-99, 1988.

[62] N. Koblitz, "Hyperelliptic Cryptosystems," *Journal of Cryptology*, vol. 1, no. 3, pp. 129-150, 1989.

[63] U. Krieger, "signature.c," Master's thesis, Mathematik und Informatik, Universität Gesamthochschule Essen, 1997.

[64] J. Kuroki, M. Gonda, K. Matsuo, J. Chao and S. Tsujii, "Fast Genus Three Hyperelliptic Curve Cryptosystems," *The 2002 Symposium on Cryptography and Information Security - SCIS 2002*, IEICE Japan, 2002.

[65] T. Lange, "Efficient Arithmetic on Hyperelliptic Curves," Ph.D. dissertation, Institute for Experimental Mathematics, University of Essen, Essen, Germany, 2001.

[66] T. Lange, "Efficient Arithmetic on Hyperelliptic Koblitz Curves," Technical Report 2-2001, University of Essen, Essen, Germany, 2001.

[67] 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.

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

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

[70] T. Lange, "Montgomery Addition for Genus Two Curves," *Algorithm Number Theory Symposium - ANTS VI*, ser. LNCS 3076, D. Buell, Ed. Berlin, Germany: Springer-Verlag, pp. 309-317, 2004.

[71] T. Lange, "Formulae for Arithmetic on Genus 2 Hyperelliptic Curves," *Applicable Algebra in Engineering, Communication and Computing*, vol.15, No.5, pp. 295-328, 2005.

[72] T. Lange, "Koblitz Curve Cryptosystems," *Finite Fields and Their Applications*, vol.11, No.2, pp. 220-229, 2005.

[73] T. Lange, "Pairings on ordinary hyperelliptic curves," Talk at Pairing in Cryptography Workshop 2005, 2005, slide available at http://www2.mat.dtu.dk/people/T.Lange/.

[74] T. Lange, and M. Stevens, "Efficient Doubling for Genus Two Curves over Binary Fields", *Eleventh Annual Workshop on Selected Areas in Cryptography - SAC 2004*, ser. LNCS 3357, H. Handschuh, M. A. Hasan, Eds. Berlin, Germany: Springer-Verlag, pp. 170-181, 2005.

[75] E. Lee, H. S. Lee, and Yoonjin Lee, "Fast computation of Tate pairing on general divisors of genus 3 hyperelliptic curves," Cryptology ePrint Archive, Report 2006/125, 2006, http://eprint.iacr.org.

[76] P. Lockhart, "On the Discriminant of a Hyperelliptic Curve," *Tran. Amer. Math. Soc.*, 342, 2, pp. 729-752, 1994.

[77] K. Matsuo, J. Chao and S. Tsujii, "Fast Genus Two Hyperelliptic Curve Cryptosystem," *Technical Report ISEC 2001-31*, IEICE Japan, 2001.

[78] A. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, Boca Raton, Florida, USA: Chapman & Hall/CRC, 1997.

[79] A. Menezes, and E. Teske, "Cryptographyic Implications of Hess's Generalized GHS Attack," Cryptology ePrint Archive, Report 2004/235, 2004, http://eprint.iacr.org.

[80] A. Menezes, Y. Wu and R. Zuccherato, "An Elementary Introduction to Hyperelliptic Curve," *Centre for Applied Cryptographic Research (CACR) Technical Reports*, CORR 1996-19, available at http://www.cacr.math.uwaterloo.ca/.

[81] V. Miller, "Use of elliptic curves in cryptography," in *Advance in Cryptology - CRYPTO'85*, ser. LNCS 218, H. C. Williams, Ed. Berlin, Germany: Springer-Verlag, pp. 417-426, 1986.

[82] Y. Miyamoto, H.Doi, K. Matsuo, J. Chao and S. Tsujii, "A Fast Addition Algorithm of Genus Two Hyperelliptic Curve," *The 2002 Symposium on Cryptography and Information Security - SCIS 2002*, IEICE Japan, pp. 497-502, 2002, in Japanese.

[83] D. Mumford, "Tata Lectures on Theta II," *Prog. Math.*, vol. 43. Birkhäuser, 1984.

[84] K. Nagao, "Improving Group Law Algorithms for Jacobians of Hyperelliptic Curves," *ANTS IV*, ser. LNCS 1838, W. Bosma, Eds. Berlin, Germany: Springer-Verlag, pp. 439-448, 2000.

[85] K. Nagao, "Improvement of Thériault Algorithm of Index Calculus for Jacobian of Hyperelliptic Curves of Small Genus," Cryptology ePrint Archive, Report 2004/161, 2004, http://eprint.iacr.org.

[86] J. Nyukai, K. Matsuo, J. Chao, and S. Tsujii, "On the resultant computation in the addition Harley algorithms on hyperelliptic cureves," *Technical Report ISEC2006-5*, IEICE Japan, May 2006. in Japanese.

[87] P. van Oorschot, and M. Wiener, "Parallel Collision Search with Cryptanalytic Applications," *Journal of Cryptology*, vol. 12, pp. 1-28, 1999.

[88] J. Pelzl, "Hyperelliptic Cryptosystems on Embedded Microprocessor," Master's thesis, Department of Electrical Engineering and Information Sciences, Ruhr-Universitäet Bochum, Bochum, Germany, 2002.

[89] J. Pelzl, T. Wollinger, J. Guajardo, and C. Paar, "Hyperelliptic Curve Cryptosystems: Closing the Performance Gap to Elliptic Curves," *Workshop on Cryptographic Hardware and Embedded Systems - CHES 2003*, ser. LNCS 2779, C. D. Walter, Ç. K. Koç, and C. Paar, Eds. Berlin, Germany: Springer-Verlag, pp. 349-365, 2003.

[90] J. Pelzl, T. Wollinger, and C. Paar, "Low Cost Security: Explicit Formulae for Genus-4 Hyperelliptic Curves," *Tenth Annual Workshop on Selected Areas in Cryptography - SAC 2003*, ser. LNCS 3006, M. Matsui and R. Zuccherato, Eds. Berlin, Germany: Springer-Verlag, pp. 1-16, 2003.

[91] J. Pelzl, T. Wollinger, and C. Paar, "High Performance Arithmetic for Special Hyperelliptic Curve Cryptosystems of Genus Two," *International Conference on Information Technology: Coding and Computing - ITCC 2004*, IEEE Computer Society, pp. 513-517, 2004.

[92] J. Pelzl, T. Wollinger, and C. Paar, "Special Hyperelliptic Curve Cryptosystems of Genus Two: Efficient Arithmetic and Fast Implementation," chapter in *Embedded Cryptographic Hardware: Design and Security*, Nadia Nedjah Ed. Nova Science Publishers, NY, USA, 2004.

[93] J. M. Pollard, "Monte Carlo Methods for Index Computation mod p," *Mathmatics of Computation*, vol. 32, no. 143, pp. 918-924, 1978.

[94] R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystem," *Communications of the ACM*, vol. 21, no.2, pp. 120-126, February 1978.

[95] Y. Sakai and K. Sakurai, "Design of Hyperelliptic Cryptosystem in Small Characteristic and a Software Implementation over $\mathbb{F}_2^n$," *Advance in Cryptology - ASIACRYPT'98*, ser. LNCS 1514, K. Ohta and D. Pei, Eds. Berlin, Germany: Springer-Verlag, pp. 80-94, 1998.

[96] Y. Sakai and K. Sakurai, "On the Performance of Hyperelliptic Curve Cryptosystem in Software Implementation," *IEICE Transctions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E83-A NO.4, pp. 692-703, April 2000.

[97] Y. Sakai, K. Sakurai, and H. Ishizuka, "Secure Hyperelliptic Curve Cryptosyatems and Performance," *Public Key Cryptography: First International Workshop on Practice and Theory in Public Key Cryptograph - PKC'98*, ser. LNCS 1431, H. Imai and Y. Zheng, Eds. Berlin, Germany: Springer-Verlag, pp. 164-181, 1998.

[98] S. Scholten, and H. J. Zhu, "Hyperelliptic Curves in Charateristic 2," *International Mathematics Research Notices*, vol. 17, pp. 905-917, 2002.

[99] N. Smart, "On the Performance of Hyperelliptic Cryptosystems," *Advance in Cryptology - EUROCRYPT'99*, ser. LNCS 1592, J. Stern, Ed. Berlin, Germany: Springer-Verlag, pp. 165-175, 1999.

[100] A. M. Spallek, "Kurven vom Geschlecht 2 und ihre Anwendung in Pub-Key-Krytosystemen," PhD. Thesis, Mathematik und Informatik, Universität Gesamthochschule Essen, 1994.

[101] H. Sugizaki, K. Matsuo, J. Chao and S. Tsujii, “An Extension of Harley Addition Algorithm for Hyperelliptic Curves over Finite Fields of Characteristic Two,” *Technical Report ISEC 2002-9*, IEICE Japan, 2002.

[102] H. Sugizaki, K. Matsuo, J. Chao, and S. Tsujii, “A generalized Harley algorithm for genus two hyperelliptic curves,” *Journal of the National Institute of Information and Communications Technology* 52, nos. 1/2, pp. 111-117, 2005.

[103] M. Takahashi, “Improving Harley Algorithm for Jacobian of Genus 2 Hyperelliptic Curves,” *The 2002 Symposium on Cryptography and Information Security - SCIS 2002*, IEICE Japan, 2002, in Japanese.

[104] N. Thériault, “Index Calculus Attack for Hyperelliptic Curves of Small Genus,” *Advance in Cryptology - ASIACRYPT ’03*, ser. LNCS 2894, G. Goos, J. Hartmanis, and J. van Leeuwen, Eds. Berlin, Germany: Springer-Verlag, pp. 79-92, 2003.

[105] A. L. Toom, “The Complexity of a Scheme of Functional Elements Realizing the Multiplication of Integers,” *Soviet Mathematics Doklady*, vol. 3, pp. 714-716, 1963.

[106] A. Weimerskirch, and C. Paar, “Generalizations of the Karatsuba Algorithm for Efficient Implementations,” Cryptology ePrint Archive, Report 2006/224, 2006, http://eprint.iacr.org.

[107] T. Wollinger, “Computer Architectures for Cryptosystems Based on Hyperelliptic Curves,” Master’s thesis, ECE Department, Worcester Polytechnic Institute, Worcester, Massachusetts, USA, May 2001.

[108] T. Wollinger, “Software and Harware Implementation of Hyperelliptic Curve Cryptosystems,” PhD. thesis, Department of Electrical Engineering and Information Sciences, Ruhr-Universitäet Bochum, Bochum, Germany, 2004.

[109] T. Wollinger, G. Bertoni, L. Breveglieri, and C. Paar, “Performance of HECC Coprocessors Using Inversionfree Formulae,” *International Workshop on Information Security & Hiding (ISH ’05) part of the International Conference on Computational Science and its Applications (ICCSA 2005)*, ser. LNCS 3982, M. Gavrilova, O. Gervasi, V. Kumar, C. J. K. Tan, D. Taniar, A. Laganà, Y. Mun, H. Choo, Eds. Berlin, Germany: Springer-Verlag, pp. 1004-1012, 2006.

[110] T. Wollinger, and C. Paar, “Hardware Architectures for Cryptosystems Based on Hyperelliptic Curves,” *Proceedings of the 9th IEEE International Conference on Electronics, Circuits and Systems - ICECS 2002*, vol. III, pp. 1159-1163, 2002.

[111] T. Wollinger, J. Pelzl, and C. Paar, “Cantor versus Harley: Optimization and Analysis of Explicit Formulae for Hyperelliptic Curve Cryptosystems,” *IEEE Transactions on Computers*, vol. 54, no. 7, pp. 861-872, 2005.

[112] T. Wollinger, J. Pelzl, V. Wittelsberger, C. Paar, G. Saldamli, and Ç. K. Koç, “Elliptic & Hyperelliptic Curves on Embedded μP,” *ACM Transactions in Embedded Computing Systems (TECS)*, Special Issue on Embedded Systems and Security, 2004.

TABLE VII

EXPLICIT FORMULA FOR ADDITION ON A HEC OF GENUS THREE OVER $\mathbb{F}_p$: ADD$^{3+2\rightarrow 3}$

Input	Genus 3 HEC $C : Y^2 = F(X), F = X^7 + f_5X^5 + f_4X^4 + f_3X^3 + f_2X^2 + f_1X + f_0$; Reduced Divisors $D_1 = (U_1, V_1)$ and $D_2 = (U_2, V_2)$, $U_1 = X^3 + u_{12}X^2 + u_{11}X + u_{10}$, $V_1 = v_{12}X^2 + v_{11}X + v_{10}$, $U_2 = X^2 + u_{21}X + u_{20}$, $V_2 = v_{21}X + v_{20}$;	
Output	Reduced Divisor $D_3 = (U_3, V_3) = D_1 + D_2$, $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30}$, $V_3 = v_{32}X^2 + v_{31}X + v_{30}$;	

Step	Expression	Cost
1	**Compute the resultant r of U_1 and U_2:** $t_1 = u_{21} - u_{12} + 1, t_2 = u_{21} - u_{11} + u_{20}, t_3 = u_{20} - u_{10}, t_4 = t_2 - u_{21}t_1;$ $t_5 = t_3 - u_{20}t_1, t_6 = t_4u_{21} - t_5, t_7 = t_4^2, t_8 = t_7u_{20}, t_9 = t_5t_6, r = t_8 - t_9;$	$6M$
2	**If $r = 0$ then call the Cantor algorithm**	–
3	**Compute the pseudo-inverse $I = i_1X + i_0 \equiv r/U_1$ mod U_2:** $i_1 = t_4, i_0 = t_6;$	–
4	**Compute $S^{'} = s_1^{'}X + s_0^{'} = rS \equiv (V_2 - V_1)I$ mod U_2:** $c_1 = v_{21} - v_{11} + v_{12}u_{21}, c_0 = v_{20} - v_{10} + v_{12}u_{20}, t_1 = i_1c_1, t_2 = i_0c_0;$ $s_0^{'} = t_1u_{20} + t_2, s_1^{'} = t_1u_{21} + (i_0 + i_1)(c_0 + c_1) - t_1 - t_2;$	$7M$
5	**If $s_1^{'} = 0$ then call the Cantor algorithm**	–
6	**Compute $S = (S^{'}/r) = s_1X + s_0$:** $t_1 = (rs_1^{'})^{-1}, t_2 = rt_1, t_3 = t_1s_1^{'}, w = rt_2, s_0 = t_3s_0^{'}, s_1 = t_3s_1^{'};$	$1I + 6M$
7	**Compute $V = s_1X^4 + k_3X^3 + k_2X^2 + k_1X + k_0 = SU_1 + V_1$:** $t_0 = s_0u_{12}, t_1 = s_0u_{10}, t_2 = s_1u_{11}, k_3 = s_1u_{12} + s_0, k_2 = t_0 + t_2 + v_{12};$ $k_1 = (s_0 + s_1)(u_{10} + u_{11}) - t_1 - t_2 + v_{11}, k_0 = t_1 + v_{10};$	$5M$
8	**Compute $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30} = s_1^{-2}(V^2 - F)/U_1U_2$:** $t_1 = u_{20} + u_{21}, t_2 = u_{10} + u_{12}, t_3 = t_1(t_2 + u_{11}), t_4 = (t_2 - u_{11})(u_{20} - u_{21});$ $t_5 = u_{12}u_{21}, z_0 = u_{10}u_{20}, z_1 = (t_3 + t_4)/2 - z_0 + u_{10}, z_2 = u_{11} + u_{20} + t_5;$ $z_3 = u_{12} + u_{21}, u_{32} = w(2k_3 - w) - z_3, u_{31} = w[2(k_2 - z_3k_3) + w(k_3^2 + z_3)] + z_3^2 - z_2;$ $u_{30} = w[2(k_1 - z_2k_3) + w(2k_2k_3 + z_2 - f_5)] + z_3(z_2 - u_{31}) - z_1;$	$15M$
9	**Compute $V_3 = v_{32}X^2 + v_{31}X + v_{30} \equiv -V$ mod U_3:** $t_1 = u_{32}s_1, t_2 = t_1 - k_3, t_3 = u_{31}t_2, v_{32} = (u_{31} + u_{32})(s_1 + t_2) - t_1 - t_3 - k_2;$ $v_{31} = u_{30}s_1 - k_1 - t_3, v_{30} = -(k_0 + u_{30}t_1);$	$5M$
Sum		$1I + 44M$

TABLE VIII

EXPLICIT FORMULA FOR ADDITION ON A HEC OF GENUS THREE OVER $\mathbb{F}_p$: ADD$^{3+1\to 3}$

Input	Genus 3 HEC $C : Y^2 = F(X), F = X^7 + f_5X^5 + f_4X^4 + f_3X^3 + f_2X^2 + f_1X + f_0$; Reduced Divisors $D_1 = (U_1, V_1)$ and $D_2 = (U_2, V_2)$, $U_1 = X^3 + u_{12}X^2 + u_{11}X + u_{10}$, $V_1 = v_{12}X^2 + v_{11}X + v_{10}$, $U_2 = X + u_{20}$, $V_2 = v_{20}$;	
Output	Reduced Divisor $D_3 = (U_3, V_3) = D_1 + D_2$, $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30}$, $V_3 = v_{32}X^2 + v_{31}X + v_{30}$;	

Step	Expression	Cost
1	**Compute the resultant r of U_1 and U_2:** $w_0 = u_{20}^2, w_1 = w_0(u_{20} - u_{12}), w_2 = u_{11}u_{20}, r = w_1 + w_2 - u_{10}$;	$3M$
2	**If $r = 0$ then call the Cantor algorithm**	–
3	**Compute the inverse of U_1 mod U_2:** $i = r^{-1}$;	$1I$
4	**Compute $s_0 \equiv i(V_2 - V_1)$ mod U_2:** $z_0 = u_{20}v_{12}, s_0 = i(v_{20} - u_{20}(z_0 - v_{11}) - v_{10})$;	$3M$
5	**Compute $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30} = (V^2 - F)/U_1U_2$, $V = s_0U_1 + V_1$:** $t_0 = s_0^2, t_1 = u_{20} + u_{12}, u_{32} = t_0 - t_1, t_2 = u_{12}u_{20}, t_3 = t_2 + u_{11}, t_4 = s_0u_{12}$; $t_5 = 2s_0(t_4 + v_{12}) - f_5, u_{31} = t_5 - t_3 - t_1u_{32}, t_6 = w_2 + u_{10}, t_7 = s_0u_{11}$; $t_8 = u_{12}v_{12}, t_9 = t_4^2 + 2s_0(t_7 + t_8 + v_{11}) + v_{12}^2 - f_4, u_{30} = t_9 - t_6 - t_3u_{32} - t_1u_{31}$;	$12M$
6	**Compute $V_3 = v_{32}X^2 + v_{31}X + v_{30} \equiv -V$ mod U_3:** $v_{32} = s_0(u_{32} - u_{12}) - v_{12}, v_{31} = s_0(u_{31} - u_{11}) - v_{11}, v_{30} = s_0(u_{30} - u_{10}) - v_{10}$;	$3M$
Sum		$1I + 21M$

TABLE IX

EXPLICIT FORMULA FOR ADDITION ON A HEC OF GENUS THREE OVER $\mathbb{F}_p$: ADD$^{1+2\to 3}$

Input	Genus 3 HEC $C : Y^2 = F(X), F = X^7 + f_5X^5 + f_4X^4 + f_3X^3 + f_2X^2 + f_1X + f_0$; Reduced Divisors $D_1 = (U_1, V_1)$ and $D_2 = (U_2, V_2) = 2D_1$, $U_1 = X + u_{10}$, $V_1 = v_{10}$, $U_2 = (X + u_{10})^2$, $V_2 = v_{21}X + v_{20}$,	
Output	Reduced Divisor $D_3 = (U_3, V_3) = D_1 + D_2 = 3D_1$, $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30}$, $V_3 = v_{32}X^2 + v_{31}X + v_{30}$;	

Step	Expression	Cost
1	**Compute $d_1 =$ gcd $(U_1, U_2) = X + u_{10} = e_1(X + u_{10}) + e_2(X + u_{10})^2$:** $e_1 = 1, e_2 = 0$;	–
2	**Compute $d =$ gcd $(d_1, V_1 + V_2) = 1 = c_1(X + u_{10}) + c_2(v_{21}X + v_{20} + v_{10})$:** $s_1 = c_1e_1 = c_1, s_2 = c_2e_2 = 0, t_0 = v_{10} - u_{10}v_{21}, s_3 = c_2 = (t_0 + v_{20})^{-1}$;	$1I + 1M$
3	**Compute $U_3 = U_1^3d^{-2} = X^3 + u_{32}X^2 + u_{31}X + u_{30}$:** $u_{10}^{'} = u_{10}^2, u_{32} = 3u_{10}, u_{31} = 3u_{10}^{'}, u_{30} = u_{10}^{'}u_{10}$;	$2M$
4	**Compute $V_3 = v_{32}X^2 + v_{31}X + v_{30} \equiv [s_1U_1V_2 + s_3(V_1V_2 + F)]d^{-1}$ mod U_3:** $t_1 = t_0 - v_{20}, f_0^{'} = f_0 + v_{20}t_0, f_1^{'} = f_1 + v_{21}t_1, f_2^{'} = f_2 - v_{21}^2, f_5^{'} = f_5 + 2u_{10}^{'}$; $t_2 = 2f_5^{'}, t_3 = t_2 + u_{10}^{'}, t_4 = 4t_2 + t_3, t_5 = f_5^{'} + t_3, t_6 = u_{31}t_3, t_7 = u_{30}t_4$; $t_8 = u_{10}t_5, t_9 = f_2^{'} - u_{32}f_3 + 2u_{31}f_4 - t_7, t_{10} = f_1^{'} - u_{31}f_3 + u_{30}(8f_4 - 5t_8)$; $t_{11} = f_0^{'} - u_{30}(f_3 - u_{32}f_4 + t_6), v_{32} = c_2t_9, v_{31} = c_2t_{10}, v_{30} = c_2t_{11}$;	$15M$
Sum		$1I + 18M$

TABLE X

EXPLICIT FORMULA FOR DOUBLING ON A HEC OF GENUS THREE OVER $\mathbb{F}_p$: DBL$^{1\rightarrow 2}$

Input	Genus 3 HEC $C : Y^2 = F(X), F = X^7 + f_5X^5 + f_4X^4 + f_3X^3 + f_2X^2 + f_1X + f_0$; Reduced Divisors $D_1 = (U_1, V_1)$, $U_1 = X + u_{10}$, $V_1 = v_{10}$;	
Output	Reduced Divisor $D_2 = (U_2, V_2) = 2D_1$, $U_2 = X^2 + u_{21}X + u_{20}$, $V_2 = v_{21}X + v_{20}$;	

Step	Expression	Cost
1	**Compute** $d =$ **gcd** $(U_1, 2V_1) = 1 = s_1(X + u_{10}) + s_3(2v_{10})$**:** $s_1 = 0, s_3 = (2v_{10})^{-1}$;	$1I$
2	**Compute** $U_2 = U_1^2 d^{-2} = X^2 + u_{21}X + u_{20}$**:** $u_{21} = 2u_{10}, u_{20} = u_{10}^2$;	$1M$
3	**Compute** $V_2 = v_{21}X + v_{20} \equiv [s_1U_1V_1 + s_3(V_1^2 + F)]d^{-1}$ **mod** U_2**:** $t_1 = 2(f_5 + u_{20}), t_2 = 2t_1 - f_5 + u_{20}, t_3 = t_1 + t_2, t_4 = u_{10}t_2, t_5 = u_{10}t_3, t_6 = 2f_4$; $t_7 = t_6 - t_4, t_8 = 2t_6 - t_5, t_9 = u_{10}t_7, t_{10} = u_{10}t_8, t_{11} = f_3 - t_9, t_{12} = 3f_3 - t_{10}$; $t_{13} = u_{20}t_{11}, t_{14} = u_{10}t_{12}, t_{15} = f_1 - t_{13}, t_{16} = 2f_2 - t_{14}, t_{17} = u_{10}t_{15}, t_{18} = u_{10}t_{16}$; $t_{19} = 2f_0 - t_{17}, t_{20} = f_1 - t_{18}, v_{21} = s_3t_{20}, v_{20} = s_3t_{19}$;	$10M$
Sum		$1I + 11M$

TABLE XI

EXPLICIT FORMULA FOR DOUBLING ON A HEC OF GENUS THREE OVER $\mathbb{F}_p$: DBL$^{2\rightarrow 3}$

Input	Genus 3 HEC $C : Y^2 = F(X), F = X^7 + f_5X^5 + f_4X^4 + f_3X^3 + f_2X^2 + f_1X + f_0$; Reduced Divisors $D_1 = (U_1, V_1)$, $U_1 = X^2 + u_{11}X + u_{10}$, $V_1 = v_{11}X + v_{10}$;	
Output	Reduced Divisor $D_2 = (U_2, V_2) = 2D_1$, $U_2 = X^3 + u_{22}X^2 + u_{21}X + u_{20}$, $V_2 = v_{22}X^2 + v_{21}X + v_{20}$;	

Step	Expression	Cost
1	**Compute the resultant** r **of** U_1 **and** V_1**:** $t_1 = u_{11}v_{11}, t_2 = v_{10} - t_1, t_3 = v_{10}t_2, v_{11}^{'} = v_{11}^2, t_4 = u_{10}v_{11}^{'}, r = t_3 + t_4$;	$4M$
2	**If** $r = 0$ **then call the Cantor algorithm**	–
3	**Compute the pseudo-inverse** $I = i_1X + i_0 \equiv r/V_1$ **mod** U_1**:** $i_1 = -v_{11}, i_0 = t_2$;	–
4	**Compute** $Z = z_1X + z_0 \equiv (F - V_1^2)/U_1$ **mod** U_1**:** $u_{11}^{'} = u_{11}^2, t_1 = f_5 - u_{10} + u_{11}^{'}, u^{'} = u_{10}u_{11}, t_2 = u_{11}t_1, t_3 = u^{'} - t_2, t_4 = f_4 + t_3$; $t_5 = (u_{10} + u_{11})(t_1 + t_4), t_6 = u_{10}t_4, t_7 = f_3 - t_5 + t_6 + t_2, t_8 = u_{11}t_7$; $t_9 = f_2 - v_{11}^{'} - t_6 - t_8, t_{10} = t_1 - u_{10} + 2u_{11}^{'}, t_{11} = u_{11}t_{10}, t_{12} = t_4 + 2u^{'} - t_{11}$; $t_{13} = (u_{10} + u_{11})(t_{10} + t_{12}), t_{14} = u_{10}t_{12}, z_1 = t_7 - t_{13} + t_{14} + t_{11}, z_0 = t_9 - t_{14}$;	$9M$
5	**Compute** $S^{'} = s_1^{'}X + s_0^{'} = 2rS \equiv ZI$ **mod** U_1**:** $t_1 = (i_0 + i_1)(z_0 + z_1), t_2 = i_0z_0, t_3 = i_1z_1, s_1^{'} = t_1 - t_2 - t_3(1 + u_{11}), s_0^{'} = t_2 - t_3u_{10}$;	$5M$
6	**If** $s_1^{'} = 0$ **then call the Cantor algorithm**	–
7	**Compute** $S = (S^{'}/2r) = s_1X + s_0$**:** $t_1 = (2r)^{-1}, s_0 = t_1s_0^{'}, s_1 = t_1s_1^{'}$;	$1I + 2M$
8	**Compute** $U_2 = X^3 + u_{22}X^2 + u_{21}X + u_{20} =$ **Monic** $(S^2 + [\frac{2SV_1}{U_1}] + [\frac{V_1^2 - F}{U_1^2}])$**:** $t_1 = s_1^2, t_2 = s_0s_1, u_{22} = -(t_1 + 2u_{11}), u_{21} = f_5 - 2(u_{10} + t_2) + 3u_{11}^{'}$; $u_{20} = f_4 - 2u_{11}(2u_{11}^{'} - 3u^{'} + f_5)$;	$3M$
9	**Compute** $V_2 = v_{22}X^2 + v_{21}X + v_{20} \equiv (-V_1 - SU_1)$ **mod** U_2**:** $v_{22} = (u_{22} - u_{11})s_1 - s_0, v_{21} = (u_{21} - u_{10})s_1 - u_{11}s_0 - v_{11}, v_{20} = u_{20}s_1 - u_{10}s_0 - v_{10}$;	$5M$
Sum		$1I + 28M$

TABLE XII

EXPLICIT FORMULA FOR ADDITION ON A HEC OF GENUS THREE OVER $\mathbb{F}_{2^n}$: ADD$^{3+2\rightarrow 3}$

Input	Genus 3 HEC $C : Y^2 + h(X)Y = F(X), h = X^3 + h_2X^2 + h_1X + h_0$; $F = X^7 + f_5X^5 + f_4X^4 + f_3X^3 + f_2X^2 + f_1X + f_0$; Reduced Divisors $D_1 = (U_1, V_1)$ and $D_2 = (U_2, V_2)$, $U_1 = X^3 + u_{12}X^2 + u_{11}X + u_{10}$, $V_1 = v_{12}X^2 + v_{11}X + v_{10}$, $U_2 = X^2 + u_{21}X + u_{20}$, $V_2 = v_{21}X + v_{20}$;	
Output	Reduced Divisor $D_3 = (U_3, V_3) = D_1 + D_2$, $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30}$, $V_3 = v_{32}X^2 + v_{31}X + v_{30}$;	

Step	Expression	Cost
1	**Compute the resultant r of U_1 and U_2:** $t_1 = u_{21} + u_{12} + 1, t_2 = u_{21} + u_{11} + u_{20}, t_3 = u_{20} + u_{10}, (e_0, e_1) = t_1 \cdot (u_{20}, u_{21})$; $t_4 = t_2 + e_1, t_5 = t_3 + e_0, t_6 = t_4u_{21} + t_5, t_7 = t_4^2, t_8 = t_7u_{20}, t_9 = t_5t_6, r = t_8 + t_9$;	$6M$
2	**If $r = 0$ then call the Cantor algorithm**	–
3	**Compute the pseudo-inverse $I = i_1X + i_0 \equiv r/U_1$ mod U_2:** $i_1 = t_4, i_0 = t_6$;	–
4	**Compute $S^{'} = s_1^{'}X + s_0^{'} = rS \equiv (V_2 + V_1)I$ mod U_2:** $(e_0, e_1) = v_{12} \cdot (u_{20}, u_{21}), c_1 = v_{11} + v_{21} + e_1, c_0 = v_{10} + v_{20} + e_0, t_1 = i_1c_1, t_2 = i_0c_0$; $(e_0, e_1) = t_1 \cdot (u_{20}, u_{21}), s_0^{'} = e_0 + t_2, s_1^{'} = e_1 + (i_0 + i_1)(c_0 + c_1) + t_1 + t_2$;	$7M$
5	**If $s_1^{'} = 0$ then call the Cantor algorithm**	–
6	**Compute $S = (S^{'}/r) = s_1X + s_0$:** $t_1 = (rs_1^{'})^{-1}, t_2 = rt_1, t_3 = t_1s_1^{'}, w = rt_2, (s_0, s_1) = t_3 \cdot (s_0^{'}, s_1^{'})$;	$1I + 6M$
7	**Compute $V = s_1X^4 + k_3X^3 + k_2X^2 + k_1X + k_0 = SU_1 + V_1$:** $(t_0, t_1) = s_0 \cdot (u_{12}, u_{10}), (t_2, e_0) = s_1 \cdot (u_{11}, u_{12}), k_3 = e_0 + s_0, k_2 = t_0 + t_2 + v_{12}$; $k_1 = (s_0 + s_1)(u_{10} + u_{11}) + t_1 + t_2 + v_{11}, k_0 = t_1 + v_{10}$;	$5M$
8	**Compute $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30} = s_1^{-2}(V^2 + hV + F)/(U_1U_2)$:** $t_1 = u_{12} + u_{21}, u_{32} = w^2 + w + t_1, t_2 = u_{11} + u_{20} + u_{12}u_{21}, t_3 = t_1^2 + t_2$; $w_1 = t_1 + k_3 + k_3^2, w_2 = f_5 + k_2 + h_2k_3 + t_2, (e_0, e_1) = w \cdot (w_1, w_2)$; $w_3 = e_0 + t_1 + h_2, w_4 = e_1 + h_1 + t_2, (t_4, e_0) = w \cdot (w_3, w_4), u_{31} = t_3 + t_4$; $t_5 = u_{10} + u_{12}(u_{20} + t_2) + u_{21}(u_{11} + t_2), t_6 = e_0 + t_1u_{31}, u_{30} = t_5 + t_6$;	$12M$
9	**Compute $V_3 = v_{32}X^2 + v_{31}X + v_{30} \equiv (h + V_1 + SU_1)$ mod U_3:** $t_1 = u_{32}s_1, t_2 = t_1 + k_3 + 1, t_3 = u_{31}t_2, v_{32} = (u_{31} + u_{32})(s_1 + t_1) + t_1 + t_3 + k_2 + h_2$; $(e_0, e_1) = u_{30} \cdot (s_1, t_1), v_{31} = e_0 + t_3 + k_1 + h_1, v_{30} = e_1 + k_0 + h_0$;	$5M$
Sum		$1I + 41M$

TABLE XIII

EXPLICIT FORMULA FOR ADDITION ON A HEC OF GENUS THREE OVER $\mathbb{F}_{2^n}$: ADD$^{3+1\to 3}$

Input	Genus 3 HEC $C : Y^2 + h(X)Y = F(X), h = X^3 + h_2X^2 + h_1X + h_0$; $F = X^7 + f_5X^5 + f_4X^4 + f_3X^3 + f_2X^2 + f_1X + f_0$; Reduced Divisors $D_1 = (U_1, V_1)$ and $D_2 = (U_2, V_2)$, $U_1 = X^3 + u_{12}X^2 + u_{11}X + u_{10}$, $V_1 = v_{12}X^2 + v_{11}X + v_{10}$, $U_2 = X + u_{20}$, $V_2 = v_{20}$;	
Output	Reduced Divisor $D_3 = (U_3, V_3) = D_1 + D_2$, $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30}$, $V_3 = v_{32}X^2 + v_{31}X + v_{30}$;	

Step	Expression	Cost
1	**Compute the resultant r of U_1 and U_2:** $w_0 = u_{20}^2, w_1 = w_0(u_{12} + u_{20}), w_2 = u_{11}u_{20}, r = w_1 + w_2 + u_{10}$;	$3M$
2	**If $r = 0$ then call the Cantor algorithm**	–
3	**Compute the inverse of U_1 mod U_2:** $i = r^{-1}$;	$1I$
4	**Compute $s_0 \equiv i(V_1 + V_2)$ mod U_2:** $(e_0, e_1) = u_{20} \cdot (v_{12}, v_{11}), z_0 = u_{20}e_0, s_0 = i(v_{10} + v_{20} + e_1 + z_0)$;	$3M$
5	**Compute $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30} = (V^2 + hV + F)/(U_1U_2)$, $V = s_0U_1 + V_1$:** $t_0 = s_0^2, u_{32} = t_0 + s_0 + u_{12} + u_{20}, t_1 = t_0 + u_{12}, t_2 = u_{12}t_1, t_3 = h_2s_0$; $w = t_2 + v_{12} + f_5 + t_3 + u_{11}, (t_4, t_5) = u_{20}(u_{32}, w)$; $u_{31} = f_5 + t_2 + t_3 + t_4 + u_{11} + v_{12}, t_6 = v_{12}(v_{12} + u_{12} + h_2)$; $t_7 = u_{12}(u_{12}^2 + f_5), u_{30} = w_0u_{32} + t_5 + t_6 + u_{11}t_0 + h_1s_0 + t_7 + u_{10} + f_4 + v_{11}$;	$11M$
6	**Compute $V_3 = v_{32}X^2 + v_{31}X + v_{30} \equiv (h + V_1 + s_0U_1)$ mod U_3:** $w_1 = u_{12} + u_{32}, w_2 = u_{11} + u_{31}, w_3 = u_{10} + u_{30}, (e_0, e_1, e_2) = s_0 \cdot (w_1, w_2, w_3)$; $v_{32} = v_{12} + h_2 + e_0 + u_{32}, v_{31} = v_{11} + h_1 + e_1 + u_{31}, v_{30} = v_{10} + h_0 + e_2 + u_{30}$;	$3M$
Sum		$1I + 20M$

TABLE XIV

EXPLICIT FORMULA FOR ADDITION ON A HEC OF GENUS THREE OVER $\mathbb{F}_{2^n}$: ADD$^{1+2\to 3}$

Input	Genus 3 HEC $C : Y^2 + h(X)Y = F(X), h = X^3 + h_2X^2 + h_1X + h_0$; $F = X^7 + f_5X^5 + f_4X^4 + f_3X^3 + f_2X^2 + f_1X + f_0$; $U_1 = X + u_{10}$, $V_1 = v_{10}$, $U_2 = X^2 + u_{10}^2$, $V_2 = v_{21}X + v_{20}$ and $2D_1 \neq \mathcal{O}$;	
Output	Reduced Divisor $D_3 = (U_3, V_3) = D_1 + D_2 = 3D_1$, $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30}$, $V_3 = v_{32}X^2 + v_{31}X + v_{30}$;	

Step	Expression	Cost
1	**Compute $d_1 =$ gcd $(U_1, U_2) = X + u_{10} = e_1(X + u_{10}) + e_2(X^2 + u_{10}^2)$:** $e_1 = 1, e_2 = 0$;	–
2	**Compute $d =$ gcd $(d_1, V_1 + V_2 + h) = 1 = (c_{12}X^2 + c_{11}X + c_{10})(X + u_{10}) +$ $c_2(X^3 + h_2X^2 + (v_{21} + h_1)X + v_{20} + v_{10} + h_0)$:** $s_{11} = u_{10} + h_2, (e, w) = u_{10} \cdot (s_{11}, s_{10}), s_{10} = e + v_{21} + h_1$; $c_{12} = (w + v_{10} + v_{20} + h_0)^{-1}, s_1 = c_1e_1 = c_{12}X^2 + c_{11}X + c_{10}$; $s_2 = c_2e_2 = 0, s_3 = c_2 = c_{12}$;	$1I + 2M$
3	**Compute $U_3 = U_1^3d^{-2} = X^3 + u_{32}X^2 + u_{31}X + u_{30}$:** $u_{10}^{'} = u_{10}^2, u_{32} = u_{10}, u_{31} = u_{10}^{'}, u_{30} = u_{10}u_{10}^{'}$;	$2M$
4	**Compute $V_3 = v_{32}X^2 + v_{31}X + v_{30} \equiv [s_1U_1V_2 + s_3(V_1V_2 + F)]d^{-1}$ mod U_3:** $f_4^{'} = f_4 + v_{21}, t_1 = h_2v_{21}, f_3^{'} = f_3 + v_{20} + t_1, t_2 = h_1v_{20}$; $f_2^{'} = f_2 + (h_1 + h_2)(v_{20} + v_{21}) + t_1 + t_2 + v_{21}^2, t_3 = w + v_{10}$; $f_1^{'} = f_1 + t_2 + v_{21}(t_3 + v_{20}), f_0^{'} = f_0 + v_{20}t_3, t_4 = u_{10}^{'2}, t_5 = f_3^{'} + t_4$; $(e, w) = t_5 \cdot (u_{10}, u_{10}^{'}), t_6 = f_2^{'} + e, t_7 = f_1^{'} + f_5t_4 + w$; $t_8 = f_0^{'} + u_{30}(t_5 + u_{10}f_4^{'}), (v_{32}, v_{31}, v_{30}) = c_{12} \cdot (t_6, t_7, t_8)$;	$15M$
Sum		$1I + 19M$

TABLE XV

EXPLICIT FORMULA FOR DOUBLING ON A HEC OF GENUS THREE OVER $\mathbb{F}_{2^n}$: DBL$^{1\rightarrow 2}$

Input	Genus 3 HEC $C : Y^2 + h(X)Y = F(X), h = X^3 + h_2X^2 + h_1X + h_0;$ $F = X^7 + f_5X^5 + f_4X^4 + f_3X^3 + f_2X^2 + f_1X + f_0;$ Reduced Divisors $D_1 = (U_1, V_1)$, $U_1 = X + u_{10}$, $V_1 = v_{10}$ and $2D_1 \neq \mathcal{O}$;	
Output	Reduced Divisor $D_2 = (U_2, V_2) = 2D_1$, $U_2 = X^2 + u_{21}X + u_{20}$, $V_2 = v_{21}X + v_{20}$;	

Step	Expression	Cost
1	**Compute** $d =$ **gcd** $(U_1, h) = 1 = s_1(X + u_{10}) + s_3(X^3 + h_2X^2 + h_1X + h_0)$**:** $s_{11} = u_{10} + h_2, (e_0, t) = u_{10} \cdot (s_{11}, s_{10}), s_{10} = e_0 + h_1, s_{12} = t + h_0, s = s_{12}^{-1};$	$1I + 2M$
2	**Compute** $U_2 = U_1^2 d^{-2} = X^2 + u_{21}X + u_{20}$**:** $u_{21} = 0, u_{20} = u_{10}^2;$	$1M$
3	**Compute** $V_2 = v_{21}X + v_{20} \equiv [s_1U_1V_1 + s_3(V_1^2 + F)]d^{-1}$ **mod** U_2**:** $w_1 = f_5 + u_{20}, (e_0, e_1) = u_{20} \cdot (w_1, f_4), w_2 = t + v_{10};$ $(e_2, e_3, e_4) = v_{10} \cdot (h_1, h_2, w_2), t_1 = f_3 + v_{10} + e_0, t_2 = f_2 + e_3 + e_1;$ $(e_0, e_1) = u_{20} \cdot (t_1, t_2), t_3 = f_1 + e_2 + e_0, t_4 = f_0 + e_4 + e_1, (v_{21}, v_{20}) = s \cdot (t_3, t_4);$	$9M$
Sum		$1I + 12M$

TABLE XVI

EXPLICIT FORMULA FOR DOUBLING ON A HEC OF GENUS THREE OVER $\mathbb{F}_{2^n}$: DBL$^{2\rightarrow 3}$

Input	Genus 3 HEC $C : Y^2 + h(X)Y = F(X), h = X^3 + h_2X^2 + h_1X + h_0;$ $F = X^7 + f_5X^5 + f_4X^4 + f_3X^3 + f_2X^2 + f_1X + f_0;$ Reduced Divisors $D_1 = (U_1, V_1)$, $U_1 = X + u_{10}$, $V_1 = v_{10}$ and $2D_1 \neq \mathcal{O}$;	
Output	Reduced Divisor $D_2 = (U_2, V_2) = 2D_1$, $U_2 = X^3 + u_{22}X^2 + u_{21}X + u_{20}$, $V_2 = v_{22}X^2 + v_{21}X + v_{20}$;	

Step	Expression	Cost
1	**Compute the resultant** r **of** U_1 **and** h**:** $h_2^{'} = h_2 + u_{11}, h_1^{'} = h_1 + u_{10}, (e_0, e_1) = u_{10} \cdot (h_1^{'}, h_2^{'}), (e_2, e_3) = u_{11} \cdot (h_0, h_2^{'});$ $t_1 = e_3 + h_1^{'}, t_2 = e_1 + h_0, t_3 = e_0 + e_2, (e_0, e_1) = t_1 \cdot (u_{11}, t_3), t_4 = e_0 + t_2, r = e_1 + t_2^2;$	$7M$
2	**If** $r = 0$ **then call the Cantor algorithm**	–
3	**Compute the pseudo-inverse** $I = i_1X + i_0 \equiv r/h$ **mod** U_1**:** $i_1 = t_1, i_0 = t_4;$	–
4	**Compute** $Z = z_1X + z_0 \equiv (F + hV_1 + V_1^2)/U_1$ **mod** U_1**:** $(e_0, e_1) = h_2 \cdot (v_{11}, v_{10}), t_1 = f_3 + v_{10} + e_0 + u_{10}^2, t_2 = u_{11}^2, f_5^{'} = f_5 + t_2;$ $f_4 = f_4^{'} + v_{11}, (t_3, e_2) = t_2 \cdot (f_5^{'}, f_4^{'}), z_1 = t_1 + t_3, t_4 = f_2 + e_1 + v_{11}(h_1 + v_{11});$ $t_5 = u_{11}z_1 + e_2, z_0 = t_4 + t_5;$	$8M$
5	**Compute** $S^{'} = s_1^{'}X + s_0^{'} = rS \equiv ZI$ **mod** U_1**:** $t_1 = (i_0 + i_1)(z_0 + z_1), t_2 = i_0z_0, t_3 = i_1z_1, s_1^{'} = t_1 + t_2 + t_3(1 + u_{11}), s_0^{'} = t_2 + t_3u_{10};$	$5M$
6	**If** $s_1^{'} = 0$ **then call the Cantor algorithm**	–
7	**Compute** $S = (S^{'}/r) = s_1X + s_0$**:** $t_1 = r^{-1}, (s_0, s_1) = t_1 \cdot (s_0^{'}, s_1^{'});$	$1I + 2M$
8	**Compute** $U_2 = X^3 + u_{22}X^2 + u_{21}X + u_{20} =$ **Monic** $(S^2 + [\frac{Sh}{U_1}] + [\frac{V_1^2+V_1h+F}{U_1^2}])$**:** $w = 1 + s_1, (u_{22}, e_0, e_1) = s_1 \cdot (w, h_2^{'}, h_1^{'});$ $t_1 = e_0 + s_0, u_{21} = t_1 + f_5^{'}, t_2 = e_1 + s_0h_2 + u_{11}t_1, u_{20} = t_2 + f_4^{'};$	$5M$
9	**Compute** $V_2 = v_{22}X^2 + v_{21}X + v_{20} \equiv (h + V_1 + SU_1)$ **mod** U_2**:** $w_1 = u_{11} + u_{22}, w_2 = u_{10} + u_{21}, (e_0, e_1, e_2) = s_1 \cdot (w_1, w_2, u_{20});$ $(e_3, e_4) = s_0 \cdot (u_{11}, u_{10}), v_{22} = e_0 + s_0 + h_2 + u_{22};$ $v_{21} = e_1 + e_3 + v_{11} + h_1 + u_{21}, v_{20} = e_2 + e_4 + v_{10} + h_0 + u_{20};$	$5M$
Sum		$1I + 32M$

TABLE XVII
EXPLICIT FORMULA FOR ADDITION ON A HEC OF GENUS THREE OVER $\mathbb{F}_p$ [86]

Input	Genus 3 HEC $C : Y^2 = F(X), F = X^7 + f_5X^5 + f_4X^4 + f_3X^3 + f_2X^2 + f_1X + f_0$; Reduced Divisors $D_1 = (U_1, V_1)$ and $D_2 = (U_2, V_2)$, $U_1 = X^3 + u_{12}X^2 + u_{11}X + u_{10}$, $V_1 = v_{12}X^2 + v_{11}X + v_{10}$, $U_2 = X^3 + u_{22}X^2 + u_{21}X + u_{20}$, $V_2 = v_{22}X^2 + v_{21}X + v_{20}$;	
Output	Reduced Divisor $D_3 = (U_3, V_3) = D_1 + D_2$, $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30}$, $V_3 = v_{32}X^2 + v_{31}X + v_{30}$;	

Step	Expression	Cost
1	**Compute the resultant r of U_1 and U_2:** $t_0 = u_{10} - u_{20}, t_1 = u_{11} - u_{21}, t_2 = u_{12} - u_{22}, t_3 = t_1 - u_{22}t_2, t_4 = t_0 - u_{21}t_2, t_5 = t_4 - u_{22}t_3$; $t_6 = u_{20}t_2 + u_{21}t_3, t_7 = -(t_4t_5 + t_3t_6), t_8 = t_2t_6 + t_1t_5, t_9 = t_2t_4 - t_1t_3, r = t_0t_7 - u_{20}(t_3t_9 + t_2t_8)$;	$15M$
2	**If $r = 0$ then call the Cantor algorithm**	–
3	**Compute the pseudo-inverse $I = i_2X^2 + i_1X + i_0 \equiv r/U_1$ mod U_2:** $i_2 = t_9, i_1 = t_8, i_0 = t_7$;	–
4	**Compute $S^{'} = s^{'}_2X^2 + s^{'}_1X + s^{'}_0 = rS \equiv (V_2 - V_1)I$ mod U_2:** $t_1 = v_{10} - v_{20}, t_2 = v_{11} - v_{21}, t_3 = v_{12} - v_{22}, t_4 = t_2i_1, t_5 = t_1i_0, t_6 = t_3i_2, t_7 = u_{22}t_6$; $t_8 = t_4 + t_6 + t_7 - (t_2 + t_3)(i_1 + i_2), t_9 = u_{20} + u_{22}, t_{10} = (t_9 + u_{21})(t_8 - t_6), t_9 = (t_9 - u_{21})(t_8 + t_6)$; $s^{'}_0 = -(u_{20}t_8 + t_5), s^{'}_1 = t_4 + t_5 + (t_9 - t_{10})/2 - (t_7 + (t_1 + t_2)(i_0 + i_1))$; $s^{'}_2 = t_6 - (s^{'}_0 + t_4 + (t_1 + t_3)(i_0 + i_2) + (t_9 + t_{10})/2)$;	$10M$
5	**If $s^{'}_2 = 0$ then call the Cantor algorithm**	–
6	**Compute $S = (S^{'}/r)$ and make S monic:** $t_1 = (rs^{'}_2)^{-1}, t_2 = rt_1, w = t_1s^{'2}_2, w_i = rt_2, s_0 = t_2s^{'}_0, s_1 = t_2s^{'}_1$;	$1I + 7M$
7	**Compute $Z = X^5 + z_4X^4 + z_3X^3 + z_2X^2 + z_1X + z_0 = SU_1$:** $t_1 = u_{10} + u_{12}, t_2 = (s_0 + s_1)(t_1 + u_{11}), t_3 = (s_0 - s_1)(t_1 - u_{11}), t_4 = u_{12}s_1$; $z_0 = u_{10}s_0, z_1 = (t_2 - t_3)/2 - t_4, z_2 = (t_2 + t_3)/2 - z_0 + u_{10}, z_3 = u_{11} + s_0 + t_4, z_4 = u_{12} + s_1$;	$4M$
8	**Compute $U_t = X^4 + u_{t3}X^3 + u_{t2}X^2 + u_{t1}X + u_{t0} = (S(Z + 2w_iV_1) - w_i^2((F - V_1^2)/U_1))/U_2)$:** $u_{t3} = z_4 + s_1 - u_{22}, t_1 = s_1z_4 - u_{22}u_{t3}, u_{t2} = z_3 + s_0 + t_1 - u_{21}, t_2 = (u_{22} + u_{21})(u_{t3} + u_{t2})$; $t_3 = s_0z_3 - u_{21}u_{t2}, u_{t1} = z_2 + (s_0 + s_1)(z_4 + z_3) + w_i(2v_{12} - w_i) - (t_1 + t_2 + t_3 + u_{20})$; $u_{t0} = z_1 + t_3 + s_1z_2 + w_i(2(v_{11} + s_1v_{12}) + w_iu_{12}) - (u_{22}u_{t1} + u_{20}u_{t3})$;	$13M$
9	**Compute $V_t = v_{t3}X^3 + v_{t2}X^2 + v_{t1}X + v_{t0} \equiv wZ + V_1$ mod U_t:** $t_1 = u_{t3} - z_4, v_{t0} = w(t_1u_{t0} + z_0) + v_{10}, v_{t1} = w(t_1u_{t1} + z_1 - u_{t0}) + v_{11}$; $v_{t2} = w(t_1u_{t2} + z_2 - u_{t1}) + v_{12}, v_{t3} = w(t_1u_{t3} + z_3 - u_{t2})$;	$8M$
10	**Compute $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30} = (F - V_t^2)/U_t$:** $t_1 = 2v_{t3}, u_{32} = -(u_{t3} + v_{t3}^2), u_{31} = f_5 - (u_{t2} + u_{32}u_{t3} + t_1v_{t2})$; $u_{30} = f_4 - (u_{t1} + v_{t2}^2 + u_{32}u_{t2} + u_{31}u_{t3} + t_1v_{t1})$;	$7M$
11	**Compute $V_3 = v_{32}X^2 + v_{31}X + v_{30} \equiv V_t$ mod U_3:** $v_{32} = v_{t2} - u_{32}v_{t3}, v_{31} = v_{t1} - u_{31}v_{t3}, v_{30} = v_{t0} - u_{30}v_{t3}$;	$3M$
Sum		$1I + 67M$

TABLE XVIII

EXPLICIT FORMULA FOR DOUBLING ON A HEC OF GENUS THREE OVER $\mathbb{F}_p$ [86]

Input	Genus 3 HEC $C : Y^2 = F(X), F = X^7 + f_5X^5 + f_4X^4 + f_3X^3 + f_2X^2 + f_1X + f_0$; Reduced Divisors $D_1 = (U_1, V_1)$, $U_1 = X^3 + u_{12}X^2 + u_{11}X + u_{10}$, $V_1 = v_{12}X^2 + v_{11}X + v_{10}$;	
Output	Reduced Divisor $D_2 = (U_2, V_2) = 2D_1$, $U_2 = X^3 + u_{22}X^2 + u_{21}X + u_{20}$, $V_2 = v_{22}X^2 + v_{21}X + v_{20}$;	

Step	Expression	Cost
1	**Compute the resultant r of U_1 and V_1:** $t_1 = v_{11} - u_{12}v_{12}, t_2 = v_{10} - u_{11}v_{12}, t_3 = t_2 - u_{12}t_1, t_4 = u_{10}v_{12} + u_{11}t_1, t_5 = t_2t_3 + t_1t_4;$ $t_6 = -(v_{11}t_3 + v_{12}t_4), t_7 = v_{11}t_1 - v_{12}t_2, r = v_{10}t_5 - u_{10}(t_1t_7 + v_{12}t_6);$	$15M$
2	**If $r = 0$ then call the Cantor algorithm**	–
3	**Compute the pseudo-inverse $I = i_2X^2 + i_1X + i_0 \equiv r/V_1$ mod U_1:** $i_2 = t_7, i_1 = t_6, i_0 = t_5;$	–
4	**Compute $Z = z_2X^2 + z_1X + z_0 \equiv (F - V_1^2)/U_1$ mod U_1:** $t_1 = 2u_{10}, t_2 = 2u_{11}, t_3 = u_{12}^2, t_4 = f_4 - (t_1 + v_{12}^2), t_5 = f_5 + t_3 - t_2, t_{10} = 2v_{12}, z_2 = t_5 + 2t_3;$ $z_1 = u_{12}(t_2 - t_5) + t_4, z_0 = f_3 + t_3(t_5 - u_{11}) + u_{12}(t_1 - t_4) + u_{11}(u_{11} - f_5) - t_{10}v_{11};$	$7M$
5	**Compute $S^{'} = s_2^{'}X^2 + s_1^{'}X + s_0^{'} = 2rS \equiv ZI$ mod U_1:** $t_1 = i_1z_1, t_2 = i_0z_0, t_3 = i_2z_2, t_4 = u_{12}t_3, t_5 = (i_1 + i_2)(z_1 + z_2) - (t_1 + t_3 + t_4), t_6 = u_{10}t_5;$ $t_7 = u_{10} + u_{12}, t_8 = t_7 + u_{11}, t_9 = t_7 - u_{11}, t_7 = t_8(t_3 + t_5), t_{11} = t_9(t_5 - t_3);$ $s_0^{'} = t_2 - t_6, s_1^{'} = t_4 + (i_0 + i_1)(z_0 + z_1) + (t_{11} - t_7)/2 - (t_1 + t_2);$ $s_2^{'} = t_1 + t_6 + (i_0 + i_2)(z_0 + z_2) - (t_2 + t_3 + (t_7 + t_{11})/2);$	$10M$
6	**If $s_2^{'} = 0$ then call the Cantor algorithm**	–
7	**Compute $S = (S^{'}/2r)$ and make S monic:** $t_1 = 2r, t_2 = (t_1s_2^{'})^{-1}, t_3 = t_1t_2, w = t_2s_2^{'2}, w_i = t_1t_3, s_0 = t_3s_0^{'}, s_1 = t_3s_1^{'};$	$1I + 7M$
8	**Compute $G = X^5 + g_4X^4 + g_3X^3 + g_2X^2 + g_1X + g_0 = SU_1$:** $t_1 = t_8(s_0 + s_1), t_2 = t_9(s_0 - s_1), t_3 = u_{12}s_1, g_0 = u_{10}s_0, g_1 = (t_1 - t_2)/2 - t_3;$ $g_2 = (t_1 + t_2)/2 - g_0 + u_{10}, g_3 = u_{11} + s_0 + t_3, z_4 = u_{12} + s_1;$	$4M$
9	**Compute $U_t = X^4 + u_{t3}X^3 + u_{t2}X^2 + u_{t1}X + u_{t0} = ((G + w_iV_1)^2 - w_i^2F)/U_1^2$:** $u_{t3} = 2s_1, u_{t2} = s_1^2 + 2s_0, u_{t1} = u_{t3}s_0 + w_i(t_{10} - w_i), u_{t0} = s_0^2 + 2w_i((s_1 - u_{12})v_{12} + v_{11} + w_iu_{12});$	$7M$
10	**Compute $V_t = v_{t3}X^3 + v_{t2}X^2 + v_{t1}X + v_{t0} \equiv wG + V_1$ mod U_t:** $t_1 = u_{t3} - g_4, v_{t0} = w(t_1u_{t0} + g_0) + v_{10}, v_{t1} = w(t_1u_{t1} + g_1 - u_{t0}) + v_{11};$ $v_{t2} = w(t_1u_{t2} + g_2 - u_{t1}) + v_{12}, v_{t3} = w(t_1u_{t3} + g_3 - u_{t2});$	$8M$
11	**Compute $U_2 = X^3 + u_{22}X^2 + u_{21}X + u_{20} = (F - V_t^2)/U_t$:** $t_1 = 2v_{t3}, u_{22} = -(u_{t3} + v_{t3}^2), u_{21} = f_5 - (u_{t2} + u_{22}u_{t3} + t_1v_{t2});$ $u_{20} = f_4 - (u_{t1} + v_{t2}^2 + u_{22}u_{t2} + u_{21}u_{t3} + t_1v_{t1});$	$7M$
12	**Compute $V_2 = v_{22}X^2 + v_{21}X + v_{20} \equiv V_t$ mod U_2:** $v_{22} = v_{t2} - u_{22}v_{t3}, v_{21} = v_{t1} - u_{21}v_{t3}, v_{20} = v_{t0} - u_{20}v_{t3};$	$3M$
Sum		$1I + 68M$

TABLE XIX

EXPLICIT FORMULA FOR ADDITION ON A HEC OF GENUS THREE OVER $\mathbb{F}_{2^n}$ WITH $h(X) = X^3$

Input	Genus 3 HEC $C : Y^2 + X^3Y = F(X), F = X^7 + f_6X^6 + f_2X^2 + f_1X + f_0$; Reduced Divisors $D_1 = (U_1, V_1)$ and $D_2 = (U_2, V_2)$, $U_1 = X^3 + u_{12}X^2 + u_{11}X + u_{10}$, $V_1 = v_{12}X^2 + v_{11}X + v_{10}$, $U_2 = X^3 + u_{22}X^2 + u_{21}X + u_{20}$, $V_2 = v_{22}X^2 + v_{21}X + v_{20}$;	
Output	Reduced Divisor $D_3 = (U_3, V_3) = D_1 + D_2$, $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30}$, $V_3 = v_{32}X^2 + v_{31}X + v_{30}$;	

Step	Expression	Cost
1	**Compute the resultant r of U_1 and U_2:** $w_0 = u_{10} + u_{20}, w_1 = u_{11} + u_{21}, w_2 = u_{12} + u_{22}, (d_0, d_1, d_2) = w_2 \cdot (u_{22}, u_{21}, u_{20})$; $t_1 = w_1 + d_0, t_2 = w_0 + d_1, (e_0, e_1) = t_1 \cdot (u_{22}, u_{21}), t_3 = t_2 + e_0, t_4 = d_2 + e_1$; $(e_0, e_1) = t_3 \cdot (t_2, w_1), (e_2, e_3) = t_4 \cdot (t_1, w_2), t_5 = e_0 + e_2, t_6 = e_1 + e_3$; $(e_0, e_1) = w_2 \cdot (t_2, t_6), t_7 = w_1t_1 + e_0, r = w_0t_5 + u_{20}(t_1t_7 + e_1)$;	$15M$
2	**If $r = 0$ then call the Cantor algorithm**	–
3	**Compute the pseudo-inverse $I = i_2X^2 + i_1X + i_0 \equiv r/U_1$ mod U_2:** $i_2 = t_7, i_1 = t_6, i_0 = t_5$;	–
4	**Compute $S^{'} = s_2^{'}X^2 + s_1^{'}X + s_0^{'} = rS \equiv (V_2 + V_1)I$ mod U_2:** $t_0 = v_{20} + v_{10}, t_1 = v_{21} + v_{11}, t_2 = v_{22} + v_{12}, t_3 = t_0i_0, t_4 = t_1i_1, t_5 = t_2i_2$; $t_6 = (i_1 + i_2)(t_1 + t_2), t_7 = (i_0 + i_2)(t_0 + t_2), t_8 = (i_0 + i_1)(t_0 + t_1)$; $(t_9, t_{13}) = t_5 \cdot (u_{22}, u_{21}), t_{10} = t_4 + t_5 + t_6 + t_9, (t_{11}, t_{12}) = t_{10} \cdot (u_{20}, u_{22})$; $t_{14} = (u_{20} + u_{21})(t_5 + t_{10}), s_0^{'} = t_3 + t_{11}, s_1^{'} = t_3 + t_4 + t_8 + t_{11} + t_{13} + t_{14}$; $s_2^{'} = t_3 + t_4 + t_5 + t_7 + t_{12} + t_{13}$;	$11M$
5	**If $s_2^{'} = 0$ then call the Cantor algorithm**	–
6	**Compute $S = (S^{'}/r)$ and make S monic:** $t_1 = (rs_2^{'})^{-1}, t_2 = rt_1, w = t_1s_2^{'2}, w_i = rt_2, (s_0, s_1) = t_2 \cdot (s_0^{'}, s_1^{'})$;	$1I + 6M + 1S$
7	**Compute $Z = X^5 + z_4X^4 + z_3X^3 + z_2X^2 + z_1X + z_0 = SU_1$:** $(z_0, e_0) = s_0 \cdot (u_{10}, u_{12}), (t_1, e_1) = s_1 \cdot (u_{11}, u_{12})$; $z_1 = (u_{10} + u_{11})(s_0 + s_1) + z_0 + t_1, z_2 = e_0 + t_1 + u_{10}, z_3 = e_1 + s_0 + u_{11}$;	$5M$
8	**Compute $U_t = X^4 + u_{t3}X^3 + u_{t2}X^2 + u_{t1}X + u_{t0} =$ $(S(Z + w_iX^3) + w_i^2((F + V_1X^3 + V_1^2)/U_1))/U_2)$:** $u_{t3} = w_2, t_0 = s_1^2, t_1 = w_1 + d_0 + t_0, u_{t2} = t_1 + w_i, (e_0, e_1) = t_1 \cdot (u_{22}, u_{21})$; $(e_2, e_3) = t_0 \cdot (u_{12}, u_{11}), t_2 = w_0 + d_1 + e_2 + e_0, t_8 = s_1 + w_i + u_{22}, t_9 = f_6 + w_2$; $(t_3, t_6) = w_i \cdot (t_8, t_9), u_{t1} = t_2 + t_3, t_{10} = s_1 + u_{22}, (e_0, e_2) = u_{22} \cdot (t_2, t_{10})$; $t_4 = e_3 + s_0^2 + d_2 + e_0 + e_1, t_5 = s_0 + u_{21} + e_2, t_7 = w_i(t_5 + t_6), u_{t0} = t_4 + t_7$;	$9M + 2S$
9	**Compute $V_t = v_{t3}X^3 + v_{t2}X^2 + v_{t1}X + v_{t0} \equiv wZ + X^3 + V_1$ mod U_t:** $t_1 = u_{22} + s_1, (e_0, e_1, e_2, e_3) = t_1 \cdot (u_{t0}, u_{t1}, u_{t2}, u_{t3}), t_2 = e_0 + z_0$; $t_3 = e_1 + z_1 + u_{t0}, t_4 = e_2 + z_2 + u_{t1}, t_5 = e_3 + z_3 + u_{t2}$; $(e_0, e_1, e_2, e_3) = w \cdot (t_2, t_3, t_4, t_5), v_{t0} = e_0 + v_{10}$; $v_{t1} = e_1 + v_{11}, v_{t2} = e_2 + v_{12}, v_{t3} = e_3 + 1$;	$8M$
10	**Compute $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30} = (F + V_tX^3 + V_t^2)/U_t$:** $u_{32} = f_6 + u_{t3} + v_{t3} + v_{t3}^2, (e_0, e_1) = u_{32} \cdot (u_{t3}, u_{t2})$; $u_{31} = u_{t2} + v_{t2} + e_0, u_{30} = u_{t1} + v_{t2}^2 + e_1 + u_{31}u_{t3} + v_{t1}$;	$3M + 2S$
11	**Compute $V_3 = v_{32}X^2 + v_{31}X + v_{30} \equiv V_t + X^3$ mod U_3:** $t_1 = v_{t3} + 1, (e_0, e_1, e_2) = t_1 \cdot (u_{30}, u_{31}, u_{32})$; $v_{32} = v_{t2} + e_2, v_{31} = v_{t1} + e_1, v_{30} = v_{t0} + e_0$;	$3M$
Sum		$1I + 60M + 5S$

TABLE XX

EXPLICIT FORMULA FOR DOUBLING ON A HEC OF GENUS THREE OVER $\mathbb{F}_{2^n}$ WITH $h(X) = X^3$

Input	Genus 3 HEC $C : Y^2 + X^3Y = F(X), F = X^7 + f_6X^6 + f_2X^2 + f_1X + f_0;$ Reduced Divisor $D_1 = (U_1, V_1),$ $U_1 = X^3 + u_{12}X^2 + u_{11}X + u_{10}$, $V_1 = v_{12}X^2 + v_{11}X + v_{10};$	
Output	Reduced Divisor $D_2 = (U_2, V_2) = 2D_1,$ $U_2 = X^3 + u_{22}X^2 + u_{21}X + u_{20}$, $V_2 = v_{22}X^2 + v_{21}X + v_{20};$	

Step	Expression	Cost
1	**Precomputation:** $u_2 = u_{12}^2, u_1 = u_{11}^2, u_0 = u_{10}^2, v_2 = v_{12}^2, v_1 = v_{11}^2, v_0 = v_{10}^2;$ $(e_0, e_1) = f_6 \cdot (u_0, u_1), t_1 = f_0 + v_0 + e_0, t_2 = f_2 + v_1 + e_1;$ $t_3 = u_0t_2 + u_1t_1, t_4 = f_1 + u_0, t_5 = u_{12}t_4 + t_1;$ **If** $t_3 = 0$ **then call the Cantor algorithm**	$5M + 6S$
2	**Compute** s_1, s_0**:** $t_6 = (t_3u_0)^{-1}, (t_7, t_8) = t_6 \cdot (t_3, u_0), t_9 = u_0t_8;$ $(k_1, k_2) = t_9 \cdot (t_4, t_5), s_1 = u_{12} + k_1, s_0 = u_{11} + k_2;$	$1I + 6M$
3	**Compute** $U_t = X^4 + u_{t3}X^3 + u_{t2}X^2 + u_{t1}X + u_{t0}$**:** $w_4 = u_0t_9, u_{t2} = s_1^2 + w_4, t_{10} = k_1 + w_4, (u_{t1}, e_0) = w_4(t_{10}, f_6);$ $t_{11} = k_2 + u_{12}k_1 + e_0, u_{t0} = s_0^2 + w_4t_{11};$	$5M + 2S$
4	**Compute** $V_t = v_{t3}X^3 + v_{t2}X^2 + v_{t1}X + v_{t0}$**:** $t_{12} = k_2t_5 + u_2t_1, (e_0, t_{13}, e_1) = t_7 \cdot (t_1, t_4, t_{12});$ $t_{14} = u_{t2} + u_2, t_{15} = u_{t0} + u_1, (e_2, e_3) = t_{13} \cdot (t_{14}, t_{15});$ $v_{t3} = e_0 + t_4^2t_8, v_{t2} = e_2 + k_1 + w_4 + u_2;$ $v_{t1} = k_1t_{10} + t_{11} + e_1 + v_2 + f_6u_2, v_{t0} = e_3 + u_1;$	$10M + 1S$
5	**Reduce** U_t**, i.e.** $U_2 = X^3 + u_{22}X^2 + u_{21}X + u_{20}$**:** $u_{22} = f_6 + v_{t3} + v_{t3}^2, u_{21} = u_{t2} + v_{t2}, u_{20} = u_{22}u_{t2} + u_{t1} + v_{t2}^2 + v_{t1};$	$1M + 2S$
6	**Compute** $V_2 = v_{22}X^2 + v_{21}X + v_{20}$**:** $t_1 = v_{t3} + 1, (e_0, e_1, e_2) = t_1 \cdot (u_{20}, u_{21}, u_{22});$ $v_{22} = v_{t2} + e_2, v_{21} = v_{t1} + e_1, v_{20} = v_{t0} + e_0;$	$3M$
Sum		$1I + 30M + 11S$

TABLE XXI

EXPLICIT FORMULA FOR ADDITION ON A HEC OF GENUS THREE OVER $\mathbb{F}_{2^n}$ WITH $h(X) = h_2X^2$

Input	Genus 3 HEC $C : Y^2 + h_2X^2Y = F(X), F = X^7 + f_6X^6 + f_4X^4 + f_1X + f_0$; Reduced Divisors $D_1 = (U_1, V_1)$ and $D_2 = (U_2, V_2)$, $U_1 = X^3 + u_{12}X^2 + u_{11}X + u_{10}$, $V_1 = v_{12}X^2 + v_{11}X + v_{10}$, $U_2 = X^3 + u_{22}X^2 + u_{21}X + u_{20}$, $V_2 = v_{22}X^2 + v_{21}X + v_{20}$;	
Output	Reduced Divisor $D_3 = (U_3, V_3) = D_1 + D_2$, $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30}$, $V_3 = v_{32}X^2 + v_{31}X + v_{30}$;	

Step	Expression	Cost
1	**Compute the resultant r of U_1 and U_2:** $w_0 = u_{10} + u_{20}, w_1 = u_{11} + u_{21}, w_2 = u_{12} + u_{22}, (d_0, d_1, d_2) = w_2 \cdot (u_{22}, u_{21}, u_{20})$; $t_1 = w_1 + d_0, t_2 = w_0 + d_1, (e_0, e_1) = t_1 \cdot (u_{22}, u_{21}), t_3 = t_2 + e_0, t_4 = d_2 + e_1$; $(e_0, e_1) = t_3 \cdot (t_2, w_1), (e_2, e_3) = t_4 \cdot (t_1, w_2), t_5 = e_0 + e_2, t_6 = e_1 + e_3$; $(e_0, e_1) = w_2 \cdot (t_2, t_6), t_7 = w_1t_1 + e_0, r = w_0t_5 + u_{20}(t_1t_7 + e_1)$;	$15M$
2	**If $r = 0$ then call the Cantor algorithm**	–
3	**Compute the pseudo-inverse $I = i_2X^2 + i_1X + i_0 \equiv r/U_1$ mod U_2:** $i_2 = t_7, i_1 = t_6, i_0 = t_5$;	–
4	**Compute $S^{'} = s^{'}_2X^2 + s^{'}_1X + s^{'}_0 = rS \equiv (V_2 + V_1)I$ mod U_2:** $t_0 = v_{20} + v_{10}, t_1 = v_{21} + v_{11}, t_2 = v_{22} + v_{12}, t_3 = t_0i_0, t_4 = t_1i_1, t_5 = t_2i_2$; $t_6 = (i_1 + i_2)(t_1 + t_2), t_7 = (i_0 + i_2)(t_0 + t_2), t_8 = (i_0 + i_1)(t_0 + t_1)$; $(t_9, t_{13}) = t_5 \cdot (u_{22}, u_{21}), t_{10} = t_4 + t_5 + t_6 + t_9, (t_{11}, t_{12}) = t_{10} \cdot (u_{20}, u_{22})$; $t_{14} = (u_{20} + u_{21})(t_5 + t_{10}), s^{'}_0 = t_3 + t_{11}, s^{'}_1 = t_3 + t_4 + t_8 + t_{11} + t_{13} + t_{14}$; $s^{'}_2 = t_3 + t_4 + t_5 + t_7 + t_{12} + t_{13}$;	$11M$
5	**If $s^{'}_2 = 0$ then call the Cantor algorithm**	–
6	**Compute $S = (S^{'}/r)$ and make S monic:** $t_1 = (rs^{'}_2)^{-1}, t_2 = rt_1, w = t_1s^{'2}_2, w_i = rt_2, (s_0, s_1) = t_2 \cdot (s^{'}_0, s^{'}_1)$;	$1I + 6M + 1S$
7	**Compute $Z = X^5 + z_4X^4 + z_3X^3 + z_2X^2 + z_1X + z_0 = SU_1$:** $(z_0, e_0) = s_0 \cdot (u_{10}, u_{12}), (t_1, e_1) = s_1 \cdot (u_{11}, u_{12})$; $z_1 = (u_{10} + u_{11})(s_0 + s_1) + z_0 + t_1, z_2 = e_0 + t_1 + u_{10}, z_3 = e_1 + s_0 + u_{11}$;	$5M$
8	**Compute $U_t = X^4 + u_{t3}X^3 + u_{t2}X^2 + u_{t1}X + u_{t0} =$ $(S(Z + h_2w_iX^2) + w_i^2((F + h_2V_1X^2 + V_1^2)/U_1))/U_2)$:** $u_{t3} = w_2, t_0 = s_1^2, u_{t2} = w_1 + d_0 + t_0, (e_0, e_1) = u_{t2} \cdot (u_{22}, u_{21})$; $(e_2, e_3) = t_0 \cdot (u_{12}, u_{11}), t_1 = w_0 + d_1 + e_2 + e_0, u_{t1} = t_1 + w_i + w_i^2$; $t_2 = e_3 + s_0^2 + d_2 + t_1u_{22} + e_1, t_3 = w_i(f_6 + w_2) + s_1 + u_{22}$; $t_7 = w_it_3, u_{t0} = t_2 + t_7$;	$7M + 3S$
9	**Compute $V_t = v_{t3}X^3 + v_{t2}X^2 + v_{t1}X + v_{t0} \equiv wZ + h_2X^2 + V_1$ mod U_t:** $t_1 = u_{22} + s_1, (e_0, e_1, e_2, e_3) = t_1 \cdot (u_{t0}, u_{t1}, u_{t2}, u_{t3}), t_2 = e_0 + z_0$; $t_3 = e_1 + z_1 + u_{t0}, t_4 = e_2 + z_2 + u_{t1}, t_5 = e_3 + z_3 + u_{t2}$; $(e_0, e_1, e_2, v_{t3}) = w \cdot (t_2, t_3, t_4, t_5), v_{t0} = e_0 + v_{10}$; $v_{t1} = e_1 + v_{11}, v_{t2} = e_2 + v_{12} + h_2$;	$8M$
10	**Compute $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30} = (F + h_2V_tX^2 + V_t^2)/U_t$:** $u_{32} = f_6 + u_{t3} + v_{t3}^2, (e_0, e_1) = u_{32} \cdot (u_{t3}, u_{t2}), (e_2, e_3) = h_2 \cdot (v_{t3}, v_{t2})$; $u_{31} = u_{t2} + e_0 + e_2, u_{30} = f_4 + u_{t1} + v_{t2}^2 + e_1 + e_3 + u_{31}u_{t3}$;	$5M + 1S$ (h_2 arb.) $3M + 2S$ (h_2 sma.)
11	**Compute $V_3 = v_{32}X^2 + v_{31}X + v_{30} \equiv V_t + h_2X^2$ mod U_3:** $(e_0, e_1, e_2) = v_{t3} \cdot (u_{32}, u_{31}, u_{30}), v_{32} = v_{t2} + e_0 + h_2, v_{31} = v_{t1} + e_1, v_{30} = v_{t0} + e_2$;	$3M$
Sum	h_2 is arbitrary	$1I + 60M + 5S$
	h_2 is small	$1I + 58M + 6S$

TABLE XXII

EXPLICIT FORMULA FOR DOUBLING ON A HEC OF GENUS THREE OVER $\mathbb{F}_{2^n}$ WITH $h(X) = h_2X^2$

Input	Genus 3 HEC $C : Y^2 + h_2X^2Y = F(X)$, $F = X^7 + f_6X^6 + f_4X^4 + f_1X + f_0$, h_2^2, h_2^{-1}; Reduced Divisor $D_1 = (U_1, V_1)$, $U_1 = X^3 + u_{12}X^2 + u_{11}X + u_{10}$, $V_1 = v_{12}X^2 + v_{11}X + v_{10}$;
Output	Reduced Divisor $D_2 = (U_2, V_2) = 2D_1$, $U_2 = X^3 + u_{22}X^2 + u_{21}X + u_{20}$, $V_2 = v_{22}X^2 + v_{21}X + v_{20}$;

Step	Expression	$h_2 = 1$	h_2^{-1} small	h_2 arbitrary
1	**Precomputation:** $u_2 = u_{12}^2, u_1 = u_{11}^2, u_0 = u_{10}^2, (e_0, e_1) = h_2 \cdot (v_{12}, v_{10})$; $(e_2, t_2, e_3) = f_6 \cdot (u_0, u_1, u_2), z_2 = f_4 + v_{12}^2 + e_3$; $z_1 = z_2 + e_0 + u_2u_{12}, t_1 = f_0 + v_{10}^2 + e_2$; $t_3 = t_2 + e_1 + u_{11}z_1, t_4 = f_1 + u_0$; **If $t_4 = 0$ then call the Cantor algorithm**	$5M + 5S$	$7M + 5S$	$7M + 5S$
2	**Compute s_1, s_0:** $t_5 = (t_4u_0)^{-1}, t_6 = t_4t_5, t_7 = u_0t_5, (t_8, k_1) = t_1(t_6, t_7)$; $k_2^{'} = t_3t_7, k_2 = u_0k_2^{'}, s_1 = u_2 + k_1, s_0 = u_1 + k_2$;	$1I + 7M$	$1I + 7M$	$1I + 7M$
3	**Compute $U_t = X^4 + u_{t3}X^3 + u_{t2}X^2 + u_{t1}X + u_{t0}$:** $w_1 = u_0t_7, w_2 = h_2^2w_1, u_2^{'} = s_1^2, u_1^{'} = w_2(1 + w_1)$;	$1M + 2S$	$3M + 1S$	$3M + 1S$
4	**Compute $V_t = v_{t3}X^3 + v_{t2}X^2 + v_{t1}X + v_{t0}$:** $(e_0, e_1) = h_2 \cdot (w_1, k_1), (e_2, e_3) = t_8 \cdot (k_1^2, k_2^2)$; $t_9 = u_2 + t_1^2t_5, t_{10} = z_2 + e_2, t_{11} = u_1 + k_2^{'}t_3$; $t_{12} = t_2 + e_3, (v_{t3}, e_2, e_3, e_4) = h_2^{-1} \cdot (t_9, t_{10}, t_{11}, t_{12})$; $v_{t2} = e_0 + e_2, v_{t1} = e_3 + e_0(f_6 + k_1)$; $v_{t0} = e_4 + e_1(k_1 + f_6w_1)$;	$7M + 3S$	$9M + 3S$	$13M + 3S$
5	**Reduce U_t, i.e. $U_2 = X^3 + u_{22}X^2 + u_{21}X + u_{20}$:** $(e_0, e_1) = h_2 \cdot (v_{t3}, v_{t2}), u_{22} = f_6 + v_{t3}^2, u_{21} = u_{t2} + e_0$; $u_{20} = f_4 + u_{22}u_{t2} + u_{t1} + v_{t2}^2 + e_1$;	$1M + 2S$	$3M + 1S$	$3M + 1S$
6	**Compute $V_2 = v_{22}X^2 + v_{21}X + v_{20}$:** $(e_0, e_1, e_2) = v_{t3} \cdot (u_{22}, u_{21}, u_{20})$; $v_{22} = v_{t2} + e_0 + h_2, v_{21} = v_{t1} + e_1, v_{20} = v_{t0} + e_2$;	$3M$	$3M$	$3M$
Sum		$1I + 24M + 12S$	$1I + 32M + 10S$	$1I + 36M + 10S$

TABLE XXIII

EXPLICIT FORMULA FOR ADDITION ON A HEC OF GENUS THREE OVER $\mathbb{F}_{2^n}$ WITH $h(X) = h_1X$

Input	Genus 3 HEC $C : Y^2 + h_1XY = F(X), F = X^7 + f_5X^5 + f_3X^3 + f_2X^2 + f_0$; Reduced Divisors $D_1 = (U_1, V_1)$ and $D_2 = (U_2, V_2)$, $U_1 = X^3 + u_{12}X^2 + u_{11}X + u_{10}$, $V_1 = v_{12}X^2 + v_{11}X + v_{10}$, $U_2 = X^3 + u_{22}X^2 + u_{21}X + u_{20}$, $V_2 = v_{22}X^2 + v_{21}X + v_{20}$;	
Output	Reduced Divisor $D_3 = (U_3, V_3) = D_1 + D_2$, $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30}$, $V_3 = v_{32}X^2 + v_{31}X + v_{30}$;	

Step	Expression	Cost
1	**Compute the resultant r of U_1 and U_2:** $w_0 = u_{10} + u_{20}, w_1 = u_{11} + u_{21}, w_2 = u_{12} + u_{22}, (d_0, d_1, d_2) = w_2 \cdot (u_{22}, u_{21}, u_{20})$; $t_1 = w_1 + d_0, t_2 = w_0 + d_1, (e_0, e_1) = t_1 \cdot (u_{22}, u_{21}), t_3 = t_2 + e_0, t_4 = d_2 + e_1$; $(e_0, e_1) = t_3 \cdot (t_2, w_1), (e_2, e_3) = t_4 \cdot (t_1, w_2), t_5 = e_0 + e_2, t_6 = e_1 + e_3$; $(e_0, e_1) = w_2 \cdot (t_2, t_6), t_7 = w_1t_1 + e_0, r = w_0t_5 + u_{20}(t_1t_7 + e_1)$;	$15M$
2	**If $r = 0$ then call the Cantor algorithm**	–
3	**Compute the pseudo-inverse $I = i_2X^2 + i_1X + i_0 \equiv r/U_1$ mod U_2:** $i_2 = t_7, i_1 = t_6, i_0 = t_5$;	–
4	**Compute $S^{'} = s^{'}_2X^2 + s^{'}_1X + s^{'}_0 = rS \equiv (V_2 + V_1)I$ mod U_2:** $t_0 = v_{20} + v_{10}, t_1 = v_{21} + v_{11}, t_2 = v_{22} + v_{12}, t_3 = t_0i_0, t_4 = t_1i_1, t_5 = t_2i_2$; $t_6 = (i_1 + i_2)(t_1 + t_2), t_7 = (i_0 + i_2)(t_0 + t_2), t_8 = (i_0 + i_1)(t_0 + t_1)$; $(t_9, t_{13}) = t_5 \cdot (u_{22}, u_{21}), t_{10} = t_4 + t_5 + t_6 + t_9, (t_{11}, t_{12}) = t_{10} \cdot (u_{20}, u_{22})$; $t_{14} = (u_{20} + u_{21})(t_5 + t_{10}), s^{'}_0 = t_3 + t_{11}, s^{'}_1 = t_3 + t_4 + t_8 + t_{11} + t_{13} + t_{14}$; $s^{'}_2 = t_3 + t_4 + t_5 + t_7 + t_{12} + t_{13}$;	$11M$
5	**If $s^{'}_2 = 0$ then call the Cantor algorithm**	–
6	**Compute $S = (S^{'}/r)$ and make S monic:** $t_1 = (rs^{'}_2)^{-1}, t_2 = rt_1, w = t_1s^{'2}_2, w_i = rt_2, w^{'} = w_i^2, (s_0, s_1) = t_2 \cdot (s^{'}_0, s^{'}_1)$;	$1I + 6M + 2S$
7	**Compute $Z = X^5 + z_4X^4 + z_3X^3 + z_2X^2 + z_1X + z_0 = SU_1$:** $(z_0, e_0) = s_0 \cdot (u_{10}, u_{12}), (t_1, e_1) = s_1 \cdot (u_{11}, u_{12})$; $z_1 = (u_{10} + u_{11})(s_0 + s_1) + z_0 + t_1, z_2 = e_0 + t_1 + u_{10}, z_3 = e_1 + s_0 + u_{11}$;	$5M$
8	**Compute $U_t = X^4 + u_{t3}X^3 + u_{t2}X^2 + u_{t1}X + u_{t0} =$** **$(S(Z + h_1w_iX) + w_i^2((F + h_1V_1X + V_1^2)/U_1))/U_2)$:** $u_{t3} = w_2, t_0 = s_1^2, u_{t2} = w_1 + d_0 + t_0, (e_0, e_1) = u_{t2} \cdot (u_{22}, u_{21})$; $(e_2, e_3) = t_0 \cdot (u_{12}, u_{11}), t_1 = w_0 + d_1 + e_2 + e_0, u_{t1} = t_1 + w^{'}$; $t_2 = e_3 + s_0^2 + d_2 + t_1u_{22} + e_1, t_3 = w^{'}w_2, u_{t0} = t_2 + t_3$;	$6M + 2S$
9	**Compute $V_t = v_{t3}X^3 + v_{t2}X^2 + v_{t1}X + v_{t0} \equiv wZ + h_1X + V_1$ mod U_t:** $t_1 = u_{22} + s_1, (e_0, e_1, e_2, e_3) = t_1 \cdot (u_{t0}, u_{t1}, u_{t2}, u_{t3}), t_2 = e_0 + z_0$; $t_3 = e_1 + z_1 + u_{t0}, t_4 = e_2 + z_2 + u_{t1}, t_5 = e_3 + z_3 + u_{t2}$; $(e_0, e_1, e_2, v_{t3}) = w \cdot (t_2, t_3, t_4, t_5), v_{t0} = e_0 + v_{10}$; $v_{t1} = e_1 + v_{11} + h_1, v_{t2} = e_2 + v_{12}$;	$8M$
10	**Compute $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30} = (F + h_1V_tX + V_t^2)/U_t$:** $u_{32} = u_{t3} + v_{t3}^2, (e_0, e_1) = u_{32} \cdot (u_{t3}, u_{t2}), u_{31} = f_5 + u_{t2} + e_0$; $u_{30} = u_{t1} + v_{t2}^2 + h_1v_{t3} + e_1 + u_{31}u_{t3}$;	$4M + 2S$ (h_1 arb.) $3M + 2S$ (h_1 sma.)
11	**Compute $V_3 = v_{32}X^2 + v_{31}X + v_{30} \equiv V_t + h_1X$ mod U_3:** $(e_0, e_1, e_2) = v_{t3} \cdot (u_{32}, u_{31}, u_{30}), v_{32} = v_{t2} + e_0, v_{31} = v_{t1} + e_1 + h_1, v_{30} = v_{t0} + e_2$;	$3M$
Sum	h_1 is arbitrary	$1I + 58M + 6S$
	h_1 is small	$1I + 57M + 6S$

TABLE XXIV

EXPLICIT FORMULA FOR DOUBLING ON A HEC OF GENUS THREE OVER $\mathbb{F}_{2^n}$ WITH $h(X) = h_1X$

Input	Genus 3 HEC $C : Y^2 + h_1XY = F(X)$, $F = X^7 + f_5X^5 + f_3X^3 + f_2X^2 + f_0$, h_1^2, h_1^{-1}; Reduced Divisor $D_1 = (U_1, V_1)$, $U_1 = X^3 + u_{12}X^2 + u_{11}X + u_{10}$, $V_1 = v_{12}X^2 + v_{11}X + v_{10}$;
Output	Reduced Divisor $D_2 = (U_2, V_2) = 2D_1$, $U_2 = X^3 + u_{22}X^2 + u_{21}X + u_{20}$, $V_2 = v_{22}X^2 + v_{21}X + v_{20}$;

Step	Expression	$h_1 = 1$	h_1^{-1} small	h_1 arbitrary
1	**Compute** $rs_2^{'}$**:** $k_0 = u_{10}^2, z_2 = f_5 + u_{12}^2, t_1 = v_{12}^2;$ $z_1 = t_1 + u_{12}z_2, w_0 = f_0 + v_{10}^2 \ (= rs_2^{'}/h_1^5);$ **If** $w_0 = 0$ **then call the Cantor algorithm**	$1M, 4S$	$1M, 4S$	$1M, 4S$
2	**Compute** $1/h_1s_2$ **and** s_1, s_0**:** $w_1 = (1/w_0) \cdot k_0 \ (= 1/h_1s_2), (k_1, k_2) = w_1 \cdot (z_2, z_1);$ $s_1 = u_{12} + k_1, s_0 = u_{11} + k_2;$	$1I, 3M$	$1I, 3M$	$1I, 3M$
3	**Compute** $U_t = X^4 + u_{t3}X^3 + u_{t2}X^2 + u_{t1}X + u_{t0}$**:** $w_2 = h_1^2w_1 \ (= h_1/s_2), u_{t3} = 0;$ $u_{t2} = s_1^2, u_{t1} = w_2w_1, u_{t0} = w_2 + s_0^2;$	$3S$	$2M, 2S$	$2M, 2S$
4	**Compute** $V_t = v_{t3}X^3 + v_{t2}X^2 + v_{t1}X + v_{t0}$**:** $(e_0, e_1, e_2, e_3) = z_2 \cdot (k_1, u_{t2}, w_2, u_{t0}), t_2 = e_0 + t_1;$ $t_3 = e_1 + w_2 + f_3 + u_1^2, t_4 = w_1(e_2 + z_1^2) + f_2 + v_1^2;$ $t_5 = e_3 + u_0^2, (v_{t3}, v_{t2}, v_{t1}, v_{t0}) = h_1^{-1} \cdot (t_2, t_3, t_4, t_5);$	$5M, 4S$	$5M, 4S$	$9M, 4S$
5	**Reduce** U_t**, i.e.** $U_2 = X^3 + u_{22}X^2 + u_{21}X + u_{20}$**:** $u_{22} = v_{t3}^2, u_{21} = f_5 + u_{t2};$ $u_{20} = f_4 + u_{22}u_{t2} + v_{t2}^2 + u_{t1} + h_1v_{t3};$	$1M, 2S$	$2M, 2S$	$2M, 2S$
6	**Compute** $V_2 = v_{22}X^2 + v_{21}X + v_{20}$**:** $(e_0, e_1, e_2) = v_{t3} \cdot (u_{22}, u_{21}, u_{20});$ $v_{22} = v_{t2} + e_0, v_{21} = v_{t1} + e_1 + h_1, v_{20} = v_{t0} + e_2;$	$3M$	$3M$	$3M$
Sum		$1I, 13M, 13S$	$1I, 16M, 12S$	$1I, 20M, 12S$

TABLE XXV

EXPLICIT FORMULA FOR ADDITION ON A HEC OF GENUS THREE OVER $\mathbb{F}_{2^n}$ WITH $h(X) = h_0$

Input	Genus 3 HEC $C : Y^2 + h_0Y = F(X), F = X^7 + f_3X^3 + f_1X + f_0$; Reduced Divisors $D_1 = (U_1, V_1)$ and $D_2 = (U_2, V_2)$, $U_1 = X^3 + u_{12}X^2 + u_{11}X + u_{10}$, $V_1 = v_{12}X^2 + v_{11}X + v_{10}$, $U_2 = X^3 + u_{22}X^2 + u_{21}X + u_{20}$, $V_2 = v_{22}X^2 + v_{21}X + v_{20}$;	
Output	Reduced Divisor $D_3 = (U_3, V_3) = D_1 + D_2$, $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30}$, $V_3 = v_{32}X^2 + v_{31}X + v_{30}$;	

Step	Expression	Cost
1	**Compute the resultant r of U_1 and U_2:** $w_0 = u_{10} + u_{20}, w_1 = u_{11} + u_{21}, w_2 = u_{12} + u_{22}, (d_0, d_1, d_2) = w_2 \cdot (u_{22}, u_{21}, u_{20})$; $w_3 = w_1 + d_0, w_4 = w_0 + d_1, (e_0, e_1) = w_3 \cdot (u_{22}, u_{21}), t_1 = w_4 + e_0, t_2 = d_2 + e_1$; $(e_0, e_1) = t_1 \cdot (w_4, w_1), (e_2, e_3) = t_2 \cdot (w_3, w_2), t_3 = e_0 + e_2, t_4 = e_1 + e_3$; $(e_0, e_1) = w_2 \cdot (w_4, t_4), t_5 = w_1w_3 + e_0, r = w_0t_3 + u_{20}(w_3t_5 + e_1)$;	$15M$
2	**If $r = 0$ then call the Cantor algorithm**	–
3	**Compute the pseudo-inverse $I = i_2X^2 + i_1X + i_0 \equiv r/U_1$ mod U_2:** $i_2 = t_5, i_1 = t_4, i_0 = t_3$;	–
4	**Compute $S^{'} = s^{'}_2X^2 + s^{'}_1X + s^{'}_0 = rS \equiv (V_2 + V_1)I$ mod U_2:** $t_0 = v_{20} + v_{10}, t_1 = v_{21} + v_{11}, t_2 = v_{22} + v_{12}, t_3 = t_0i_0, t_4 = t_1i_1, t_5 = t_2i_2$; $t_6 = (i_1 + i_2)(t_1 + t_2), t_7 = (i_0 + i_2)(t_0 + t_2), t_8 = (i_0 + i_1)(t_0 + t_1)$; $(t_9, t_{13}) = t_5 \cdot (u_{22}, u_{21}), t_{10} = t_4 + t_5 + t_6 + t_9, (t_{11}, t_{12}) = t_{10} \cdot (u_{20}, u_{22})$; $t_{14} = (u_{20} + u_{21})(t_5 + t_{10}), s^{'}_0 = t_3 + t_{11}, s^{'}_1 = t_3 + t_4 + t_8 + t_{11} + t_{13} + t_{14}$; $s^{'}_2 = t_3 + t_4 + t_5 + t_7 + t_{12} + t_{13}$;	$11M$
5	**If $s^{'}_2 = 0$ then call the Cantor algorithm**	–
6	**Compute $S = (S^{'}/r)$ and make S monic:** $t_1 = (rs^{'}_2)^{-1}, t_2 = rt_1, w = t_1s^{'2}_2, w_i = rt_2, w^{'} = w_i^2, (s_0, s_1) = t_2 \cdot (s^{'}_0, s^{'}_1)$;	$1I + 6M + 2S$
7	**Compute $Z = X^5 + z_4X^4 + z_3X^3 + z_2X^2 + z_1X + z_0 = SU_1$:** $(z_0, e_0) = s_0 \cdot (u_{10}, u_{12}), (t_1, e_1) = s_1 \cdot (u_{11}, u_{12})$; $z_1 = (u_{10} + u_{11})(s_0 + s_1) + z_0 + t_1, z_2 = e_0 + t_1 + u_{10}, z_3 = e_1 + s_0 + u_{11}$;	$5M$
8	**Compute $U_t = X^4 + u_{t3}X^3 + u_{t2}X^2 + u_{t1}X + u_{t0} =$ $(S(Z + h_0w_i) + w_i^2((F + h_0V_1 + V_1^2)/U_1))/U_2)$:** $u_{t3} = w_2, t_0 = s_1^2, u_{t2} = w_3 + t_0, (e_0, e_1) = u_{t2} \cdot (u_{22}, u_{21})$; $(e_2, e_3) = t_0 \cdot (u_{12}, u_{11}), t_1 = w_4 + e_2 + e_0, u_{t1} = t_1 + w^{'}$; $t_2 = e_3 + s_0^2 + d_2 + t_1u_{22} + e_1, u_{t0} = t_2 + w^{'}w_2$;	$6M + 2S$
9	**Compute $V_t = v_{t3}X^3 + v_{t2}X^2 + v_{t1}X + v_{t0} \equiv wZ + h_0 + V_1$ mod U_t:** $t_1 = u_{22} + s_1, (e_0, e_1, e_2, e_3) = t_1 \cdot (u_{t0}, u_{t1}, u_{t2}, u_{t3}), t_2 = e_0 + z_0$; $t_3 = e_1 + z_1 + u_{t0}, t_4 = e_2 + z_2 + u_{t1}, t_5 = e_3 + z_3 + u_{t2}$; $(e_0, e_1, e_2, v_{t3}) = w \cdot (t_2, t_3, t_4, t_5), v_{t0} = e_0 + v_{10} + h_0$; $v_{t1} = e_1 + v_{11}, v_{t2} = e_2 + v_{12}$;	$8M$
10	**Compute $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30} = (F + h_0V_t + V_t^2)/U_t$:** $u_{32} = u_{t3} + v_{t3}^2, (e_0, e_1) = u_{32} \cdot (u_{t3}, u_{t2}), u_{31} = u_{t2} + e_0$; $u_{30} = u_{t1} + v_{t2}^2 + e_1 + u_{31}u_{t3}$;	$3M + 2S$
11	**Compute $V_3 = v_{32}X^2 + v_{31}X + v_{30} \equiv V_t + h_0$ mod U_3:** $(e_0, e_1, e_2) = v_{t3} \cdot (u_{32}, u_{31}, u_{30}), v_{32} = v_{t2} + e_0, v_{31} = v_{t1} + e_1, v_{30} = v_{t0} + e_2 + h_0$;	$3M$
Sum		$1I + 57M + 6S$

TABLE XXVI

EXPLICIT FORMULA FOR DOUBLING ON A HEC OF GENUS THREE OVER $\mathbb{F}_{2^n}$ WITH $h(X) = h_0$

Input	Genus 3 HEC $C : Y^2 + h_0 Y = F(X)$, $F = X^7 + f_3 X^3 + f_1 X + f_0$, h_0^2, h_0^{-1}; Reduced Divisor $D_1 = (U_1, V_1)$, $U_1 = X^3 + u_{12} X^2 + u_{11} X + u_{10}$, $V_1 = v_{12} X^2 + v_{11} X + v_{10}$;
Output	Reduced Divisor $D_2 = (U_2, V_2) = 2D_1$, $U_2 = X^3 + u_{22} X^2 + u_{21} X + u_{20}$, $V_2 = v_{22} X^2 + v_{21} X + v_{20}$;

Step	Expression	$h_0 = 1$	h_0^{-1} small	h_0 arbitrary
1	**Compute U_1^2 and V_1^2:** **If $u_{12} = 0$ then call the Cantor algorithm** $u_2 = u_{12}^2, u_1 = u_{11}^2, u_0 = u_{10}^2, v_2 = v_{12}^2, v_1 = v_{11}^2, v_0 = v_{10}^2$;	$6S$	$6S$	$6S$
2	**Compute $U_t = X^4 + u_{t3} X^3 + u_{t2} X^2 + u_{t1} X + u_{t0}$:** $t_1 = f_3 + u_1, t_2 = f_1 + u_0, t_3 = f_0 + v_0, t_4 = u_2^{-1}$; $(e_0, e_1, h) = t_4 \cdot (v_2, t_1, h_0), u_{t3} = 0, u_{t2} = e_0^2 + u_2$; $u_{t1} = h^2, w = u_2 u_{t2}, u_{t0} = e_1^2 + u_1 + w$;	$1I, 3M, 3S$	$1I, 4M, 3S$	$1I, 4M, 3S$
3	**Compute $V_t = v_{t3} X^3 + v_{t2} X^2 + v_{t1} X + v_{t0}$:** $t_5 = w + t_1, (e_0, e_1) = v_2 \cdot (u_{t2}, u_{t0}), t_6 = e_0 + v_1$; $t_8 = (u_2 + v_2)(u_{t0} + u_{t1}) + e_1 + t_2, t_9 = e_1 + t_3$; $(v_{t3}, e_0, e_1, v_{t0}) = h_0^{-1} \cdot (t_5, t_6, t_8, t_9)$; $v_{t2} = e_0 + h, v_{t1} = e_1 + h$;	$3M$	$3M$	$7M$
4	**Reduce U_t, i.e. $U_2 = X^3 + u_{22} X^2 + u_{21} X + u_{20}$:** $u_{22} = v_{t3}^2, u_{21} = u_{t2}, u_{20} = u_{22} u_{t2} + v_{t2}^2 + u_{t1}$;	$1M, 2S$	$1M, 2S$	$1M, 2S$
5	**Compute $V_2 = v_{22} X^2 + v_{21} X + v_{20}$:** $(e_0, e_1, e_2) = v_{t3} \cdot (u_{20}, u_{21}, u_{22})$; $v_{22} = v_{t2} + e_2, v_{21} = v_{t1} + e_1, v_{20} = v_{t0} + e_0$;	$3M$	$3M$	$3M$
Sum		$1I, 10M, 11S$	$1I, 11M, 11S$	$1I, 15M, 11S$

TABLE XXVII

INVERSION-FREE EXPLICIT FORMULA FOR ADDITION ON A HEC OF GENUS THREE OVER $\mathbb{F}_p$

Input	Genus 3 HEC $C : Y^2 = F(X), F = X^7 + f_5X^5 + f_4X^4 + f_3X^3 + f_2X^2 + f_1X + f_0$; Reduced Divisors $D_1 = [U_{12}, U_{11}, U_{10}, V_{12}, V_{11}, V_{10}, Z_1]$ and $D_2 = [U_{22}, U_{21}, U_{20}, V_{22}, V_{21}, V_{20}, Z_2]$;	
Output	Reduced Divisor $D_3 = [U_{32}, U_{31}, U_{30}, V_{32}, V_{31}, V_{30}, Z_3] = D_1 + D_2$ (Projective + Projective);	

Step	Expression	Cost
1	**Precomputation:** $Z = Z_1Z_2, U^{'}_{12} = Z_2U_{12}, U^{'}_{11} = Z_2U_{11}, U^{'}_{10} = Z_2U_{10}, V^{'}_{12} = Z_2V_{12}, V_{11} = Z_2V_{11}, V_{10} = Z_2V_{10}$; $Z^{'} = Z^2, U^{'}_{22} = Z_1U_{22}, U^{'}_{21} = Z_1U_{21}, U^{'}_{20} = Z_1U_{20}, V^{'}_{22} = Z_1V_{22}, V^{'}_{21} = Z_1V_{21}, V^{'}_{20} = Z_1V_{20}$;	$13M + 1S$
2	**Compute the resultant r of U_1 and U_2:** $t_0 = U^{'}_{10} - U^{'}_{20}, t_1 = U^{'}_{11} - U^{'}_{21}, t_2 = U^{'}_{12} - U^{'}_{22}, t_3 = Zt_1 - U^{'}_{22}t_2, t_4 = Zt_0 - U^{'}_{21}t_2$; $t_5 = Zt_4 - U^{'}_{22}t_3, t_6 = U^{'}_{20}Zt_2 + U^{'}_{21}t_3, t_7 = -(t_4t_5 + t_3t_6), t_8 = t_2t_6 + t_1t_5$; $t_9 = t_2t_4 - t_1t_3, r = t_0t_7 - U^{'}_{20}(t_3t_9 + t_2t_8)$;	$19M$
3	**If $r = 0$ then call the Cantor algorithm**	–
4	**Compute the pseudo-inverse $I = i_2X^2 + i_1X + i_0 \equiv r/U_1$ mod U_2:** $i_2 = t_9, i_1 = t_8, i_0 = t_7$;	–
5	**Compute $S^{'} = s^{'}_2X^2 + s^{'}_1X + s^{'}_0 = rS \equiv (V_2 - V_1)I$ mod U_2:** $t_1 = V^{'}_{10} - V^{'}_{20}, t_2 = V^{'}_{11} - V^{'}_{21}, t_3 = V^{'}_{12} - V^{'}_{22}, t_4 = t_2i_1, t_5 = t_1i_0, t_6 = t_3i_2$; $t^{'}_6 = Zt_6, t_7 = U^{'}_{22}t_6, t_8 = t_4 + t_7 + t^{'}_6 - (t_2 + t_3)(i_1 + Zi_2), t_9 = U^{'}_{20} + U^{'}_{22}$; $t_{10} = (t_9 + U^{'}_{21})(t_8 - t^{'}_6), t_9 = (t_9 - U^{'}_{21})(t_8 + t_6), s^{'}_0 = -(U^{'}_{20}t_8 + t_5)$; $s^{'}_1 = Z(t_4 - t_7) + t_5 - (t_1 + t_2)(i_0 + Zi_1) + (t_9 - t_{10})/2$; $s^{'}_2 = Z(t^{'}_6 - t_4) - s^{'}_0 - (t_1 + t_3)(i_0 + Z^{'}i_2) - (t_9 + t_{10})/2$;	$16M$
6	**If $s^{'}_2 = 0$ then call the Cantor algorithm**	–
7	**Monic $S = X^2 + (s^{'}_1/s^{'}_2)X + s^{'}_0/s^{'}_2$:**	–
8	**Precomputation:** $w_0 = s^{'}_0Z, w_1 = s^{'}_1Z, w_2 = s^{'}_2Z, w_3 = w_2^2, w_4 = w_3Z, R = rZ, R^{'} = R^2$; $A = w_3Z^{'}, B = R^{'}w_2, D = Rw_2, E = w_3D^2, F = Bs^{'2}_2$;	$10M + 4S$
9	**Compute $Z = X^5 + z_4X^4 + z_3X^3 + z_2X^2 + z_1X + z_0 = SU_1$:** $z_0 = s^{'}_0U^{'}_{10}, z_1 = (s^{'}_0 + s^{'}_1)(U^{'}_{10} + U^{'}_{11}) - s^{'}_1U^{'}_{11} - s^{'}_0U^{'}_{10}$; $z_2 = (s^{'}_0 + s^{'}_2)(U^{'}_{10} + U^{'}_{12}) - s^{'}_2U^{'}_{12} - s^{'}_0U^{'}_{10} + s^{'}_1U^{'}_{11}$; $z_3 = w_0 + (s^{'}_1 + s^{'}_2)(U^{'}_{12} + U^{'}_{11}) - s^{'}_1U^{'}_{11} - s^{'}_2U^{'}_{12}, z_4 = w_1 + s^{'}_2U^{'}_{12}$;	$6M$
10	**Compute $U_t = X^4 + u_{t3}X^3 + u_{t2}X^2 + u_{t1}X + u_{t0} = (S(Z + 2w_iV_1) - w_i^2((F - V_1^2)/U_1))/U_2)$:** $u_{t3} = z_4 + w_1 - U^{'}_{22}s^{'}_2, t_1 = w_1z_4 - (w_2u_{t3})U_{22}, u_{t2} = w_2(z_3 + w_0 - U^{'}_{21}s^{'}_2) + t_1$; $t_2 = (U^{'}_{22} + U^{'}_{21})(w_2u_{t3} + u_{t2}), t_3 = Z^{'}(s^{'}_0z_3 - U^{'}_{21}u_{t2}), t_4 = z_2 + rV^{'}_{12}, t_5 = z_1 + rV^{'}_{11}$; $u_{t1} = Z[w_2(t_4 + rV^{'}_{12}) + (w_0 + w_1)(z_3 + z_4) - R^{'} - t_1] - (t_2 + t_3 + w_3U^{'}_{20})$; $u_{t0} = Z^{'}[w_2(t_5 + rV^{'}_{11}) + w_1(t_4 + rV^{'}_{12})] + Z[t_3 + R^{'}U^{'}_{12} - (w_2u_{t3})U^{'}_{20}] - U^{'}_{22}u_{t1}$	$23M$
11	**Compute $V_t = v_{t3}X^3 + v_{t2}X^2 + v_{t1}X + v_{t0} \equiv wZ + V_1$ mod U_t:** $t_1 = u_{t3} - z_4, v_{t0} = t_1u_{t0} + A(z_0 + V^{'}_{10}r), v_{t1} = t_1u_{t1} + w_4t_5 - s^{'}_2u_{t0}$; $v_{t2} = t_1u_{t2} + w_3t_4 - s^{'}_2u_{t1}, v_{t3} = t_1u_{t3} + w_2z_3 - u_{t2}$;	$11M$
12	**Compute $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30} = (F - V_t^2)/U_t$:** $t_1 = 2v_{t3}, u_{32} = -(Bu_{t3} + v_{t3}^2), u_{31} = B(w_3f_5 - u_{t2}) - t_1v_{t2} - u_{32}u_{t3}$; $u_{30} = s^{'}_2[B(w_4f_4 - u_{t1}) - t_1v_{t1}] - (v_{t2}^2 + u_{32}u_{t2} + u_{31}u_{t3})$; $u_{32} = u_{32}w_3, u_{31} = u_{31}w_2$;	$13M + 2S$
13	**Adjust:** $Z_3 = ED, U_{32} = u_{32}D, U_{31} = u_{31}D, U_{30} = u_{30}D$;	$4M$
14	**Compute $V_3 = v_{32}X^2 + v_{31}X + v_{30} \equiv V_t$ mod U_3:** $V_{32} = FZ^{'}v_{t2} - u_{32}v_{t3}, V_{31} = FZv_{t1} - u_{31}v_{t3}, V_{30} = Fv_{t0} - u_{30}v_{t3}$;	$8M$
Sum		$123M + 7S$

TABLE XXVIII

INVERSION-FREE EXPLICIT FORMULA FOR ADDITION ON A HEC OF GENUS THREE OVER $\mathbb{F}_{2^n}$ WITH $h(X) = 1$

Input	Genus 3 HEC $C : Y^2 = F(X), F = X^7 + f_3X^3 + f_1X + f_0$; Reduced Divisors $D_1 = [U_{12}, U_{11}, U_{10}, V_{12}, V_{11}, V_{10}, Z_1]$ and $D_2 = [U_{22}, U_{21}, U_{20}, V_{22}, V_{21}, V_{20}, Z_2]$;	
Output	Reduced Divisor $D_3 = [U_{32}, U_{31}, U_{30}, V_{32}, V_{31}, V_{30}, Z_3] = D_1 + D_2$ (Projective + Projective);	

Step	Expression	Cost
1	**Precomputation:** $Z = Z_1Z_2, U_{12}^{'} = Z_2U_{12}, U_{11}^{'} = Z_2U_{11}, U_{10}^{'} = Z_2U_{10}, V_{12}^{'} = Z_2V_{12}, V_{11} = Z_2V_{11}, V_{10} = Z_2V_{10}$; $Z^{'} = Z^2, U_{22}^{'} = Z_1U_{22}, U_{21}^{'} = Z_1U_{21}, U_{20}^{'} = Z_1U_{20}, V_{22}^{'} = Z_1V_{22}, V_{21}^{'} = Z_1V_{21}, V_{20}^{'} = Z_1V_{20}$;	$13M + 1S$
2	**Compute the resultant r of U_1 and U_2:** $w_0 = U_{10}^{'} + U_{20}^{'}, w_1 = U_{11}^{'} + U_{21}^{'}, w_2 = U_{12}^{'} + U_{22}^{'}, e_0 = U_{22}^{'}w_2, e_1 = U_{21}^{'}w_2$; $e_2 = U_{20}^{'}w_2, w_3 = w_1Z + e_0, w_4 = w_0Z + e_1, t_1 = Zw_4 + U_{22}^{'}w_3, t_2 = Ze_2 + U_{21}^{'}w_3$; $t_3 = w_4t_1 + w_3t_2, t_4 = w_2t_2 + w_1t_1, t_5 = w_1w_3 + w_2w_4, r = w_0t_3 + U_{20}^{'}(w_3t_5 + w_2t_4)$;	$19M$
3	**If $r = 0$ then call the Cantor algorithm**	–
4	**Compute the pseudo-inverse $I = i_2X^2 + i_1X + i_0 \equiv r/U_1$ mod U_2:** $i_2 = t_5, i_1 = t_4, i_0 = t_3$;	–
5	**Compute $S^{'} = s_2^{'}X^2 + s_1^{'}X + s_0^{'} = rS \equiv (V_2 + V_1)I$ mod U_2:** $t_0 = V_{10}^{'} + V_{20}^{'}, t_1 = V_{11}^{'} + V_{21}^{'}, t_2 = V_{12}^{'} + V_{22}^{'}, t_3 = t_0i_0, t_4 = t_1i_1, t_5 = t_2i_2$; $t_6 = (i_1 + Zi_2)(t_1 + t_2), t_7 = (i_0 + Z^{'}i_2)(t_0 + t_2), t_8 = (i_0 + Zi_1)(t_0 + t_1), t_9 = U_{22}^{'}t_5$; $t_{10} = t_4 + Zt_5 + t_6 + t_9, t_{11} = U_{20}^{'}t_{10}, t_{12} = U_{22}^{'}t_{10}, t_{13} = U_{21}^{'}t_5, t_{14} = (U_{20}^{'} + U_{21}^{'})(Zt_5 + t_{10})$; $s_0^{'} = t_3 + t_{11}, s_1^{'} = Z(t_4 + t_{13}) + t_3 + t_8 + t_{11} + t_{14}, s_2^{'} = Z(t_4 + t_{13}) + Z^{'}t_5 + t_3 + t_7 + t_{12}$;	$17M$
6	**If $s_2^{'} = 0$ then call the Cantor algorithm**	–
7	**Monic $S = X^2 + (s_1^{'}/s_2^{'})X + s_0^{'}/s_2^{'}$:**	–
8	**Precomputation:** $d_0 = s_0^{'}Z, d_1 = s_1^{'}Z, d_2 = s_2^{'}Z, d_3 = s_2^{'2}, d_4 = d_2^2, d_5 = Zd_4, d_6 = d_3Z, R_1 = rZ$; $R_2 = R_1^2, R_3 = ZR_2, A = d_4Z^{'}, B = d_3R_3, D = s_2^{'}B, E = DZ^{'}, F = d_2E, G = d_2R_1$;	$13M + 3S$
9	**Compute $Z = X^5 + z_4X^4 + z_3X^3 + z_2X^2 + z_1X + z_0 = SU_1$:** $z_0 = s_0^{'}U_{10}^{'}, z_1 = (s_0^{'} + s_1^{'})(U_{10}^{'} + U_{11}^{'}) + s_1^{'}U_{11}^{'} + s_0^{'}U_{10}^{'}$; $z_2 = (s_0^{'} + s_2^{'})(U_{10}^{'} + U_{12}^{'}) + s_2^{'}U_{12}^{'} + s_0^{'}U_{10}^{'} + s_1^{'}U_{11}^{'}$; $z_3 = w_0 + (s_1^{'} + s_2^{'})(U_{12}^{'} + U_{11}^{'}) + s_1^{'}U_{11}^{'} + s_2^{'}U_{12}^{'}$;	$6M$
10	**Compute $U_t = X^4 + u_{t3}X^3 + u_{t2}X^2 + u_{t1}X + u_{t0} =$ $(S(Z + h_0w_i) + w_i^2((F + h_0V_1 + V_1^2)/U_1))/U_2)$:** $u_{t3} = w_2, t_0 = s_1^{'2}, w = t_0Z^{'}, u_{t2} = d_3w_3 + w, t_1 = d_6w_4 + wU_{12}^{'} + u_{t2}U_{22}^{'}, u_{t1} = t_1 + R_3$; $t_2 = Z(wU_{11}^{'} + u_{t2}U_{21}) + (s_0^{'}Z^{'})^2 + d_4e_2 + t_1U_{22}^{'}, u_{t0} = t_2 + R_3w_2$;	$13M + 2S$
11	**Compute $V_t = v_{t3}X^3 + v_{t2}X^2 + v_{t1}X + v_{t0} \equiv wZ + V_1 + 1$ mod U_t:** $t_1 = U_{22}^{'}s_2^{'} + d_1, v_{t0} = t_1u_{t0} + A[z_0 + r(V_{10}^{'} + Z)], v_{t1} = t_1u_{t1} + d_5(z_1 + rV_{11}^{'}) + s_2^{'}u_{t0}$; $v_{t2} = t_1u_{t2} + d_4(z_1 + rV_{12}^{'}) + s_2^{'}u_{t1}, v_{t3} = s_2^{'}t_1u_{t3} + d_2z_3 + u_{t2}$;	$15M$
12	**Compute $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30} = (F + V_t + V_t^2)/U_t$:** $u_{32} = Bu_{t3} + v_{t3}^2, u_{31} = R_3u_{t2} + u_{32}u_{t3}, u_{30} = d_3(R_3u_{t1} + u_{31}u_{t3}) + v_{t2}^2 + u_{32}u_{t2}$; $u_{32} = u_{32}d_4, u_{31} = u_{31}d_6$;	$9M + 2S$
13	**Adjust:** $Z_3 = FG, U_{32} = u_{32}G, U_{31} = u_{31}G, U_{30} = u_{30}G$;	$4M$
14	**Compute $V_3 = v_{32}X^2 + v_{31}X + v_{30} \equiv V_t + 1$ mod U_3:** $V_{32} = Ev_{t2} + u_{32}v_{t3}, V_{31} = DZv_{t1} + u_{31}v_{t3}, V_{30} = Dv_{t0} + u_{30}v_{t3} + Z_3$;	$7M$
Sum		$116M + 8S$

TABLE XXIX

INVERSION-FREE EXPLICIT FORMULA FOR MIXED ADDITION ON A HEC OF GENUS THREE OVER $\mathbb{F}_p$

Input	Genus 3 HEC $C : Y^2 = F(X), F = X^7 + f_5X^5 + f_4X^4 + f_3X^3 + f_2X^2 + f_1X + f_0$; Reduced Divisors $D_1 = [U_{12}, U_{11}, U_{10}, V_{12}, V_{11}, V_{10}, Z_1]$ and $D_2 = [U_{22}, U_{21}, U_{20}, V_{22}, V_{21}, V_{20}, 1]$;	
Output	Reduced Divisor $D_3 = [U_{32}, U_{31}, U_{30}, V_{32}, V_{31}, V_{30}, Z_3] = D_1 + D_2$ (Projective + Affine);	

Step	Expression	Cost
1	**Precomputation:** $U^{'}_{22} = Z_1U_{22}, U^{'}_{21} = Z_1U_{21}, U^{'}_{20} = Z_1U_{20}, V^{'}_{22} = Z_1V_{22}, V^{'}_{21} = Z_1V_{21}, V^{'}_{20} = Z_1V_{20}$;	$6M$
2	**Compute the resultant r of U_1 and U_2:** $t_0 = U_{10} - U^{'}_{20}, t_1 = U_{11} - U^{'}_{21}, t_2 = U_{12} - U^{'}_{22}, t_3 = t_1 - U_{22}t_2, t_4 = t_0 - U_{21}t_2$; $t_5 = t_4 - U_{22}t_3, t_6 = U^{'}_{20}t_2 + U_{21}t_3, t_7 = -(t_4t_5 + t_3t_6), t_8 = t_2t_6 + t_1t_5$; $t_9 = t_2t_4 - t_1t_3, r = t_0t_7 - U_{20}(t_3t_9 + t_2t_8)$;	$15M$
3	**If $r = 0$ then call the Cantor algorithm**	–
4	**Compute the pseudo-inverse $I = i_2X^2 + i_1X + i_0 \equiv r/U_1$ mod U_2:** $i_2 = t_9, i_1 = t_8, i_0 = t_7$;	–
5	**Compute $S^{'} = s^{'}_2X^2 + s^{'}_1X + s^{'}_0 = rS \equiv (V_2 - V_1)I$ mod U_2:** $t_1 = V_{10} - V^{'}_{20}, t_2 = V_{11} - V^{'}_{21}, t_3 = V_{12} - V^{'}_{22}, t_4 = t_2i_1, t_5 = t_1i_0, t_6 = t_3i_2$; $t_7 = U_{22}t_6, t_8 = t_4 + t_6 + t_7 - (t_2 + t_3)(i_1 + i_2), t_9 = U_{20} + U_{22}$; $t_{10} = (t_9 + U_{21})(t_8 - t_6), t_9 = (t_9 - U_{21})(t_8 + t_6), s^{'}_0 = -(U_{20}t_8 + t_5)$; $s^{'}_1 = t_4 + t_5 + (t_9 - t_{10})/2 - t_7 - (t_1 + t_2)(i_0 + i_1)$; $s^{'}_2 = t_6 - s^{'}_0 - (t_9 + t_{10})/2 - t_4 - (t_1 + t_3)(i_0 + i_2)$;	$10M$
6	**If $s^{'}_2 = 0$ then call the Cantor algorithm**	–
7	**Monic $S = X^2 + (s^{'}_1/s^{'}_2)X + s^{'}_0/s^{'}_2$:**	–
8	**Precomputation:** $w_0 = s^{'}_0Z_1, w_1 = s^{'}_1Z_1, w_2 = s^{'}_2Z_1, w_3 = s^{'2}_2, w_4 = w_2^2, R = rZ_1, R^{'} = r^2$; $A = w_3Z_1, B = Rw_2, D = BR, E = B^2, F = w_2Z_1, G = EF$;	$9M + 4S$
9	**Compute $Z = X^5 + z_4X^4 + z_3X^3 + z_2X^2 + z_1X + z_0 = SU_1$:** $z_0 = s^{'}_0U_{10}, z_1 = (s^{'}_0 + s^{'}_1)(U_{10} + U_{11}) - s^{'}_1U_{11} - s^{'}_0U_{10}$; $z_2 = (s^{'}_0 + s^{'}_2)(U_{10} + U_{12}) - s^{'}_2U_{12} - s^{'}_0U_{10} + s^{'}_1U_{11}$; $z_3 = w_0 + (s^{'}_1 + s^{'}_2)(U_{12} + U_{11}) - s^{'}_1U_{11} - s^{'}_2U_{12}, z_4 = w_1 + s^{'}_2U_{12}$;	$6M$
10	**Compute $U_t = X^4 + u_{t3}X^3 + u_{t2}X^2 + u_{t1}X + u_{t0} = (S(Z + 2w_iV_1) - w_i^2((F - V_1^2)/U_1))/U_2)$:** $u_{t3} = z_4 + w_1 - U_{22}w_2, t_1 = s^{'}_1z_4 - (s^{'}_2u_{t3})U_{22}, u_{t2} = s^{'}_2(z_3 + w_0 - U_{21}w_2) + t_1$; $t_2 = (U_{22} + U_{21})(s^{'}_2u_{t3} + u_{t2}), t_3 = s^{'}_0z_3 - U_{21}u_{t2}, t_4 = z_2 + rV_{12}, t_5 = z_1 + rV_{11}$; $u_{t1} = s^{'}_2(t_4 + rV_{12}) + (s^{'}_0 + s^{'}_1)(z_3 + z_4) - rR - (t_1 + t_2 + t_3 + AU_{20})$; $u_{t0} = s^{'}_2(t_5 + rV_{11}) + s^{'}_1(t_4 + rV_{12}) + t_3 + R^{'}U_{12} - (s^{'}_2u_{t3})U_{20} - U_{22}u_{t1}$	$20M$
11	**Compute $V_t = v_{t3}X^3 + v_{t2}X^2 + v_{t1}X + v_{t0} \equiv wZ + V_1$ mod U_t:** $t_1 = u_{t3} - z_4, v_{t0} = t_1u_{t0} + A(z_0 + rV_{10}), v_{t1} = t_1u_{t1} + w_2(s^{'}_2t_5 - u_{t0})$; $v_{t2} = t_1u_{t2} + w_2(s^{'}_2t_4 - u_{t1}), v_{t3} = t_1u_{t3} + w_2z_3 - u_{t2}Z_1$;	$12M$
12	**Compute $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30} = (F - V_t^2)/U_t$:** $t_1 = 2v_{t3}, u_{32} = -(Du_{t3} + v_{t3}^2), u_{31} = D(w_4f_5 - u_{t2}Z_1) - [u_{32}u_{t3} + t_1(v_{t2}Z_1)]$; $u_{30} = E(w_4f_4 - u_{t1}Z_1) - [(v_{t2}Z_1)^2 + u_{32}(u_{t2}Z_1) + u_{31}u_{t3} + t_1Fv_{t1}]$; $u_{32} = u_{32}w_4, u_{31} = u_{31}w_2$;	$15M + 2S$
13	**Adjust:** $Z_3 = Gw_4r, U_{32} = u_{32}B, U_{31} = u_{31}B, U_{30} = u_{30}B$;	$5M$
14	**Compute $V_3 = v_{32}X^2 + v_{31}X + v_{30} \equiv V_t$ mod U_3:** $V_{32} = Gv_{t2} - u_{32}v_{t3}, V_{31} = Gv_{t1} - u_{31}v_{t3}, V_{30} = Gv_{t0} - u_{30}v_{t3}$;	$6M$
Sum		$104M + 6S$

TABLE XXX

INVERSION-FREE EXPLICIT FORMULA FOR MIXED ADDITION ON A HEC OF GENUS THREE OVER $\mathbb{F}_{2^n}$ WITH $h(X) = 1$

Input	Genus 3 HEC $C : Y^2 = F(X), F = X^7 + f_3X^3 + f_1X + f_0$; Reduced Divisors $D_1 = [U_{12}, U_{11}, U_{10}, V_{12}, V_{11}, V_{10}, Z_1]$ and $D_2 = [U_{22}, U_{21}, U_{20}, V_{22}, V_{21}, V_{20}, 1]$;	
Output	Reduced Divisor $D_3 = [U_{32}, U_{31}, U_{30}, V_{32}, V_{31}, V_{30}, Z_3] = D_1 + D_2$ (Projective + Affine);	

Step	Expression	Cost
1	**Precomputation:** $Z = Z_1^2, U'_{22} = Z_1U_{22}, U'_{21} = Z_1U_{21}, U'_{20} = Z_1U_{20}, V'_{22} = Z_1V_{22}, V'_{21} = Z_1V_{21}, V'_{20} = Z_1V_{20}$;	$6M + 1S$
2	**Compute the resultant r of U_1 and U_2:** $w_0 = U_{10} + U'_{20}, w_1 = U_{11} + U'_{21}, w_2 = U_{12} + U'_{22}, e_0 = U_{22}w_2, e_1 = U_{21}w_2$; $e_2 = U_{20}w_2, w_3 = w_1 + e_0, w_4 = w_0 + e_1, t_1 = w_4 + U_{22}w_3, t_2 = e_2 + U_{21}w_3$; $t_3 = w_4t_1 + w_3t_2, t_4 = w_2t_2 + w_1t_1, t_5 = w_1w_3 + w_2w_4, r = w_0t_3 + U_{20}(w_3t_5 + w_2t_4)$;	$15M$
3	**If $r = 0$ then call the Cantor algorithm**	–
4	**Compute the pseudo-inverse $I = i_2X^2 + i_1X + i_0 \equiv r/U_1$ mod U_2:** $i_2 = t_5, i_1 = t_4, i_0 = t_3$;	–
5	**Compute $S' = s'_2X^2 + s'_1X + s'_0 = rS \equiv (V_2 + V_1)I$ mod U_2:** $t_0 = V_{10} + V'_{20}, t_1 = V_{11} + V'_{21}, t_2 = V_{12} + V'_{22}, t_3 = t_0i_0, t_4 = t_1i_1, t_5 = t_2i_2$; $t_6 = (i_1 + i_2)(t_1 + t_2), t_7 = (i_0 + i_2)(t_0 + t_2), t_8 = (i_0 + i_1)(t_0 + t_1), t_9 = U_{22}t_5$; $t_{10} = t_4 + t_5 + t_6 + t_9, t_{11} = U_{20}t_{10}, t_{12} = U_{22}t_{10}, t_{13} = U_{21}t_5, t_{14} = (U_{20} + U_{21})(t_5 + t_{10})$; $s'_0 = t_3 + t_{11}, s'_1 = t_3 + t_4 + t_8 + t_{11} + t_{13} + t_{14}, s'_2 = t_3 + t_4 + t_5 + t_7 + t_{12} + t_{13}$;	$11M$
6	**If $s'_2 = 0$ then call the Cantor algorithm**	–
7	**Monic $S = X^2 + (s'_1/s'_2)X + s'_0/s'_2$:**	–
8	**Precomputation:** $d_1 = s'_2Z_1, d_2 = s_2^{'2}, d_3 = d_1^2, d_4 = rd_1, R = rZ_1, R' = r^2, A = d_2Z_1, B = AR', D = Bd_1Z$;	$7M + 3S$
9	**Compute $Z = X^5 + z_4X^4 + z_3X^3 + z_2X^2 + z_1X + z_0 = SU_1$:** $z_0 = s'_0U_{10}, z_1 = (s'_0 + s'_1)(U_{10} + U_{11}) + s'_1U_{11} + s'_0U_{10}$; $z_2 = (s'_0 + s'_2)(U_{10} + U_{12}) + s'_2U_{12} + s'_0U_{10} + s'_1U_{11}$; $z_3 = w_0 + (s'_1 + s'_2)(U_{12} + U_{11}) + s'_1U_{11} + s'_2U_{12}$;	$6M$
10	**Compute $U_t = X^4 + u_{t3}X^3 + u_{t2}X^2 + u_{t1}X + u_{t0} =$ $(S(Z + h_0w_i) + w_i^2((F + h_0V_1 + V_1^2)/U_1))/U_2)$:** $u_{t3} = w_2, t_0 = s_1^{'2}, u_{t2} = d_2w_3 + t_0Z_1, t_1 = d_2w_4 + t_0U_{12} + u_{t2}U_{22}, u_{t1} = t_1 + R'Z_1$; $t_2 = t_0U_{11} + s_0^{'2}Z_1 + d_2e_2 + t_1U_{22} + u_{t2}U_{21}, u_{t0} = t_2 + R'w_2$;	$12M + 2S$
11	**Compute $V_t = v_{t3}X^3 + v_{t2}X^2 + v_{t1}X + v_{t0} \equiv wZ + V_1 + 1$ mod U_t:** $t_1 = U_{22}s'_2 + s'_1, v_{t0} = t_1u_{t0} + d_2[z_0 + r(V_{10} + Z_1)], v_{t1} = t_1u_{t1} + d_2(z_1 + rV_{11}) + s'_2u_{t0}$; $v_{t2} = t_1u_{t2} + d_2(z_2 + rV_{12}) + s'_2u_{t1}, v_{t3} = s'_2(t_1u_{t3} + z_3) + u_{t2}$;	$14M$
12	**Compute $U_3 = X^3 + u_{32}X^2 + u_{31}X + u_{30} = (F + V_t + V_t^2)/U_t$:** $u_{32} = Bu_{t3} + v_{t3}^2, u_{31} = R^2u_{t2} + u_{32}u_{t3}, u_{30} = Z(Bu_{t1} + v_{t2}^2) + d_2u_{31}u_{t3} + Z_1u_{32}u_{t2}$; $u_{32} = u_{32}d_3, u_{31} = u_{31}A$;	$11M + 3S$
13	**Adjust:** $Z_3 = d_3^2R'd_4, U_{32} = u_{32}d_4, U_{31} = u_{31}d_4, U_{30} = u_{30}d_4$;	$5M + 1S$
14	**Compute $V_3 = v_{32}X^2 + v_{31}X + v_{30} \equiv V_t + 1$ mod U_3:** $V_{32} = Dv_{t2} + u_{32}v_{t3}, V_{31} = Dv_{t1} + u_{31}v_{t3}, V_{30} = Dv_{t0} + u_{30}v_{t3} + Z_3$;	$6M$
Sum		$93M + 10S$

TABLE XXXI

INVERSION-FREE EXPLICIT FORMULA FOR DOUBLING ON A HEC OF GENUS THREE OVER $\mathbb{F}_p$

Input	Genus 3 HEC $C : Y^2 = F(X), F = X^7 + f_5X^5 + f_4X^4 + f_3X^3 + f_2X^2 + f_1X + f_0$; Reduced Divisors $D_1 = [U_{12}, U_{11}, U_{10}, V_{12}, V_{11}, V_{10}, Z_1]$;	
Output	Reduced Divisor $D_2 = [U_{22}, U_{21}, U_{20}, V_{22}, V_{21}, V_{20}, Z_2] = 2D_1$ (Projective);	

Step	Expression	Cost
1	**Precomputation:** $Z = Z_1^2, U'_{12} = Z_1U_{12}, U'_{11} = Z_1U_{11}, U'_{10} = Z_1U_{10}, V'_{12} = Z_1V_{12}, V'_{11} = Z_1V_{11}, V'_{10} = Z_1V_{10}$;	$6M + 1S$
2	**Compute the resultant r of U_1 and V_1:** $t_1 = V'_{11} - U_{12}V_{12}, t_2 = V'_{10} - U_{11}V_{12}, t_3 = Z_1t_2 - U_{12}t_1, t_4 = U'_{10}V_{12} + U_{11}t_1$; $t_5 = t_2t_3 + t_1t_4, t_6 = -(V_{11}t_3 + V_{12}t_4), t_7 = V_{11}t_1 - V_{12}t_2$; $r = Z(V'_{10}t_5 - U'_{10}(t_1t_7 + V_{12}t_6))$;	$17M$
3	**If $r = 0$ then call the Cantor algorithm**	–
4	**Compute the pseudo-inverse $I = i_2X^2 + i_1X + i_0 \equiv r/V_1$ mod U_1:** $i_2 = t_7, i_1 = t_6, i_0 = t_5$;	–
5	**Compute $Z = z_2X^2 + z_1X + z_0 \equiv (F - V_1^2)/U_1$ mod U_1:** $t_1 = U_{12}^2, t_2 = f_4Z - (2U'_{10} + V_{12}^2), t_3 = f_5Z + t_1 - 2U'_{11}$; $z_2 = Z(t_3 + 2t_1), z_1 = U'_{12}(2U'_{11} - t_3) + Zt_2$; $z_0 = f_3Z^2 + t_1(t_3 - U'_{11}) + U'_{12}(2U'_{10} - t_2) + U'_{11}(U'_{11} - f_5Z) - 2V'_{12}V'_{11}$;	$10M + 3S$
6	**Compute $S' = s'_2X^2 + s'_1X + s'_0 = 2rS \equiv ZI$ mod U_1:** $t_1 = i_1z_1, t_2 = i_0z_0, t_3 = i_2z_2, t_4 = u_{12}t_3, t_5 = (i_1 + Z_1i_2)(z_1 + z_2) - (t_1 + Z_1t_3 + t_4)$; $t_6 = U_{10}t_5, t_7 = U_{10} + U_{12}, t_8 = t_7 + U_{11}, t_9 = t_7 - U_{11}, t_7 = t_8(Z_1t_3 + t_5)$; $t_{11} = t_9(t_5 - Z_1t_3), s'_2 = Z_1(t_1 - Z_1t_3) + t_6 + (i_0 + Zi_2)(z_0 + z_2) - (t_2 + (t_7 + t_{11})/2)$; $s'_1 = Z_1(t_4 - t_1) + (i_0 + Z_1i_1)(z_0 + z_1) + (t_{11} - t_7)/2 - t_2, s'_0 = t_2 - t_6$;	$16M$
7	**If $s'_2 = 0$ then call the Cantor algorithm**	–
8	**Monic $S = X^2 + (s'_1/s'_2)X + s'_0/s'_2$:**	–
9	**Precomputation:** $w_0 = s'_0Z, w_1 = s'_1Z, w_2 = s'_2Z, R = rZ, A = w_2^2, B = 2Rw_2, D = 2RB, E = B^2, F = AD$;	$7M + 2S$
10	**Compute $G = X^5 + g_4X^4 + g_3X^3 + g_2X^2 + g_1X + g_0 = SU_1$:** $g_0 = s'_0U_{10}, g_1 = (s'_0 + s'_1)(U_{10} + U_{11}) - s'_1U_{11} - s'_0U_{10}$; $g_2 = (s'_0 + s'_2)(U_{10} + U_{12}) - s'_2U_{12} - s'_0U_{10} + s'_1U_{11}$; $g_3 = w_0 + (s'_1 + s'_2)(U_{12} + U_{11}) - s'_1U_{11} - s'_2U_{12}, g_4 = w_1 + s'_2U_{12}$;	$6M$
11	**Compute $U_t = X^4 + u_{t3}X^3 + u_{t2}X^2 + u_{t1}X + u_{t0} = ((G + w_iV_1)^2 - w_i^2F)/U_1^2$:** $u_{t3} = 2w_1, u_{t2} = w_1^2 + 2w_0w_2, u_{t1} = 2[w_1w_0 + 2R(s'_2V_{12} - R)]$; $u_{t0} = w_0^2 + 4r[U_{12}(2R - s'_2V_{12}) + s'_1V'_{12} + s'_2V'_{11}]$;	$8M + 2S$
12	**Compute $V_t = v_{t3}X^3 + v_{t2}X^2 + v_{t1}X + v_{t0} \equiv wG + V_1$ mod U_t:** $t_1 = u_{t3} - g_4, v_{t0} = t_1u_{t0} + A(g_0 + 2rV_{10}), v_{t1} = t_1u_{t1} + A(g_1 + 2rV_{11}) - w_2u_{t0}$; $v_{t2} = t_1u_{t2} + A(g_2 + 2rV_{12}) - w_2u_{t1}, v_{t3} = t_1u_{t3} - u_{t2} + w_2g_3$;	$13M$
13	**Compute $U_2 = X^3 + u_{22}X^2 + u_{21}X + u_{20} = (F - V_t^2)/U_t$:** $t_1 = 2v_{t3}, u_{22} = -(Du_{t3} + v_{t3}^2), u_{21} = D(f_5A - u_{t2}) - (u_{22}u_{t3} + t_1v_{t2})$; $u_{20} = E(f_4A - u_{t1}) - (v_{t2}^2 + u_{22}u_{t2} + u_{21}u_{t3} + w_2t_1v_{t1}), u_{21} = u_{21}w_2, u_{22} = u_{22}A$;	$13M + 2S$
14	**Adjust:** $Z_2 = FBw_2, U_{22} = u_{22}B, U_{21} = u_{21}B, U_{20} = u_{20}B$;	$5M$
15	**Compute $V_2 = v_{22}X^2 + v_{21}X + v_{20} \equiv V_t$ mod U_2:** $V_{22} = Fv_{t2} - u_{22}v_{t3}, V_{21} = Fv_{t1} - u_{21}v_{t3}, V_{20} = Fv_{t0} - u_{20}v_{t3}$;	$6M$
Sum		$107M + 10S$

TABLE XXXII

AFFINE INVERSION-FREE EXPLICIT FORMULA FOR DOUBLING ON A HEC OF GENUS THREE OVER $\mathbb{F}_p$

Input	Genus 3 HEC $C : Y^2 = F(X), F = X^7 + f_5X^5 + f_4X^4 + f_3X^3 + f_2X^2 + f_1X + f_0$; Reduced Divisors $D_1 = [U_{12}, U_{11}, U_{10}, V_{12}, V_{11}, V_{10}, 1]$;	
Output	Reduced Divisor $D_2 = [U_{22}, U_{21}, U_{20}, V_{22}, V_{21}, V_{20}, Z] = 2D_1$ (Affine);	

Step	Expression	Cost
1	**Compute the resultant r of U_1 and V_1:** $t_1 = V_{11} - U_{12}V_{12}, t_2 = V_{10} - U_{11}V_{12}, t_3 = t_2 - U_{12}t_1, t_4 = U_{10}V_{12} + U_{11}t_1$; $t_5 = t_2t_3 + t_1t_4, t_6 = -(V_{11}t_3 + V_{12}t_4), t_7 = V_{11}t_1 - V_{12}t_2, r = V_{10}t_5 - U_{10}(t_1t_7 + V_{12}t_6)$;	$15M$
2	**If $r = 0$ then call the Cantor algorithm**	–
3	**Compute the pseudo-inverse $I = i_2X^2 + i_1X + i_0 \equiv r/V_1$ mod U_1:** $i_2 = t_7, i_1 = t_6, i_0 = t_5$;	–
4	**Compute $Z = z_2X^2 + z_1X + z_0 \equiv (F - V_1^2)/U_1$ mod U_1:** $t_1 = 2u_{10}, t_2 = 2u_{11}, t_3 = u_{12}^2, t_4 = f_4 - (t_1 + v_{12}^2), t_5 = f_5 + t_3 - t_2, t_{10} = 2v_{12}, z_2 = t_5 + 2t_3$; $z_1 = u_{12}(t_2 - t_5) + t_4, z_0 = f_3 + t_3(t_5 - u_{11}) + u_{12}(t_1 - t_4) + u_{11}(u_{11} - f_5) - t_{10}v_{11}$;	$7M$
5	**Compute $S^{'} = s_2^{'}X^2 + s_1^{'}X + s_0^{'} = 2rS \equiv ZI$ mod U_1:** $t_1 = i_1z_1, t_2 = i_0z_0, t_3 = i_2z_2, t_4 = u_{12}t_3, t_5 = (i_1 + i_2)(z_1 + z_2) - (t_1 + t_3 + t_4), t_6 = u_{10}t_5$; $t_7 = u_{10} + u_{12}, t_8 = t_7 + u_{11}, t_9 = t_7 - u_{11}, t_7 = t_8(t_3 + t_5), t_{11} = t_9(t_5 - t_3)$; $s_0^{'} = t_2 - t_6, s_1^{'} = t_4 + (i_0 + i_1)(z_0 + z_1) + (t_{11} - t_7)/2 - (t_1 + t_2)$; $s_2^{'} = t_1 + t_6 + (i_0 + i_2)(z_0 + z_2) - (t_2 + t_3 + (t_7 + t_{11})/2)$;	$10M$
6	**If $s_2^{'} = 0$ then call the Cantor algorithm**	–
7	**Monic $S = X^2 + (s_1^{'}/s_2^{'})X + s_0^{'}/s_2^{'}$:**	–
8	**Precomputation:** $A = s_2^{'2}, B = 2rs_2^{'}, D = 2rB, E = B^2, F = AD$;	$3M + 2S$
9	**Compute $G = X^5 + g_4X^4 + g_3X^3 + g_2X^2 + g_1X + g_0 = SU_1$:** $g_0 = s_0^{'}U_{10}, g_1 = (s_0^{'} + s_1^{'})(U_{10} + U_{11}) - s_1^{'}U_{11} - s_0^{'}U_{10}$; $g_2 = (s_0^{'} + s_2^{'})(U_{10} + U_{12}) - s_2^{'}U_{12} - s_0^{'}U_{10} + s_1^{'}U_{11}$; $g_3 = s_0^{'} + (s_1^{'} + s_2^{'})(U_{12} + U_{11}) - s_1^{'}U_{11} - s_2^{'}U_{12}, g_4 = s_1^{'} + s_2^{'}U_{12}$;	$6M$
10	**Compute $U_t = X^4 + u_{t3}X^3 + u_{t2}X^2 + u_{t1}X + u_{t0} = ((G + w_iV_1)^2 - w_i^2F)/U_1^2$:** $u_{t3} = 2s_1^{'}, u_{t2} = s_1^{'2} + 2s_0^{'}s_2^{'}, u_{t1} = 2[s_1^{'}s_0^{'} + 2r(s_2^{'}V_{12} - r)]$; $u_{t0} = s_0^{'2} + 4r[U_{12}(2r - s_2^{'}V_{12}) + s_1^{'}V_{12} + s_2^{'}V_{11}]$;	$8M + 2S$
11	**Compute $V_t = v_{t3}X^3 + v_{t2}X^2 + v_{t1}X + v_{t0} \equiv wG + V_1$ mod U_t:** $t_1 = u_{t3} - g_4, v_{t0} = t_1u_{t0} + A(g_0 + 2rV_{10}), v_{t1} = t_1u_{t1} + A(g_1 + 2rV_{11}) - s_2^{'}u_{t0}$; $v_{t2} = t_1u_{t2} + A(g_2 + 2rV_{12}) - s_2^{'}u_{t1}, v_{t3} = t_1u_{t3} - u_{t2} + s_2^{'}g_3$;	$13M$
12	**Compute $U_2 = X^3 + u_{22}X^2 + u_{21}X + u_{20} = (F - V_t^2)/U_t$:** $t_1 = 2v_{t3}, u_{22} = -(Du_{t3} + v_{t3}^2), u_{21} = D(f_5A - u_{t2}) - (u_{22}u_{t3} + t_1v_{t2})$; $u_{20} = E(f_4A - u_{t1}) - (v_{t2}^2 + u_{22}u_{t2} + u_{21}u_{t3} + s_2^{'}t_1v_{t1}), u_{21} = u_{21}s_2^{'}, u_{22} = u_{22}A$;	$13M + 2S$
13	**Adjust:** $Z = FBs_2^{'}, U_{22} = u_{22}B, U_{21} = u_{21}B, U_{20} = u_{20}B$;	$5M$
14	**Compute $V_2 = v_{22}X^2 + v_{21}X + v_{20} \equiv V_t$ mod U_2:** $V_{22} = Fv_{t2} - u_{22}v_{t3}, V_{21} = Fv_{t1} - u_{21}v_{t3}, V_{20} = Fv_{t0} - u_{20}v_{t3}$;	$6M$
Sum		$86M + 6S$

TABLE XXXIII

INVERSION-FREE EXPLICIT FORMULA FOR DOUBLING ON A HEC OF GENUS THREE OVER $\mathbb{F}_{2^n}$ WITH $h(X) = 1$

Input	Genus 3 HEC $C : Y^2 + Y = F(X), F = X^7 + f_3X^3 + f_1X + f_0$; Reduced Divisors $D_1 = [U_{12}, U_{11}, U_{10}, V_{12}, V_{11}, V_{10}, Z_1]$;	
Output	Reduced Divisor $D_2 = [U_{22}, U_{21}, U_{20}, V_{22}, V_{21}, V_{20}, Z_2] = 2D_1$ (Projective);	

Step	Expression	Cost
1	**Precomputation:** $Z = Z_1^2, Z^{'} = Z^2, \tilde{Z} = Z^{'2}, Z_i = ZZ^{'}, u_2 = U_{12}^2$; $u_1 = U_{11}^2, u_0 = U_{10}^2, v_2 = V_{12}^2, v_1 = V_{11}^2, v_0 = V_{10}^2$;	$1M + 9S$
2	**Compute** $U_t = X^4 + u_{t3}X^3 + u_{t2}X^2 + u_{t1}X + u_{t0}$**:** $t_1 = f_3Z + u_1, t_2 = f_1Z + u_0, t_3 = f_0Z + v_0, u_{t3} = 0$; $t_4 = v_2Z_1, u_2^{'} = u_2^2, u_{t2} = t_4^2 + u_2u_2^{'}, w_0 = u_2u_{t2}$; $w_1 = u_2Z_1, w_2 = w_1^2, u_{t0} = t_1^2Z^{'} + u_1w_2 + w_0$;	$9M + 4S$
3	**Compute** $V_t = v_{t3}X^3 + v_{t2}X^2 + v_{t1}X + v_{t0}$**:** $v_{t3} = t_1w_2 + w_0, v_{t2} = v_2u_{t2} + v_1w_2 + u_2Z_i, w_3 = v_2u_{t0}, w_4 = w_2Z$; $v_{t1} = (v_2 + u_2)(\tilde{Z} + u_{t0}) + w_3 + t_2w_4 + u_2\tilde{Z}, v_{t0} = w_3 + t_3w_4$;	$10M$
4	**Reduce** U_t**, i.e.** $U_2 = X^3 + u_{22}X^2 + u_{21}X + u_{20}$**:** $u_{22} = v_{t3}^2, w_5 = w_4^2, u_{21} = u_{t2}w_5, u_{20} = u_{22}u_{t2} + w_2v_{t2}^2 + w_5Z_i$; $u_{22} = u_{22}w_2, A = w_5u_2^{'}, B = AZ$;	$7M + 3S$
13	**Adjust:** $Z_2 = Bw_4, U_{22} = u_{22}w_4, U_{21} = u_{21}w_4, U_{20} = u_{20}w_4$;	$4M$
5	**Compute** $V_2 = v_{22}X^2 + v_{21}X + v_{20}$**:** $V_{22} = Bv_{t2} + v_{t3}u_{22}, V_{21} = Av_{t1} + v_{t3}u_{21}, V_{20} = Av_{t0} + v_{t3}u_{20}$;	$6M$
Sum		$37M + 16S$

TABLE XXXIV

AFFINE INVERSION-FREE EXPLICIT FORMULA FOR DOUBLING ON A HEC OF GENUS THREE OVER $\mathbb{F}_{2^n}$ WITH $h(X) = 1$

Input	Genus 3 HEC $C : Y^2 + Y = F(X), F = X^7 + f_3X^3 + f_1X + f_0$; Reduced Divisors $D_1 = [U_{12}, U_{11}, U_{10}, V_{12}, V_{11}, V_{10}, 1]$;	
Output	Reduced Divisor $D_2 = [U_{22}, U_{21}, U_{20}, V_{22}, V_{21}, V_{20}, Z] = 2D_1$ (Affine);	

Step	Expression	Cost
1	**Precomputation:** $u_2 = U_{12}^2, u_2^{'} = u_2^2, u_1 = U_{11}^2, u_0 = U_{10}^2, v_2 = V_{12}^2, v_1 = V_{11}^2, v_0 = V_{10}^2$;	$7S$
2	**Compute** $U_t = X^4 + u_{t3}X^3 + u_{t2}X^2 + u_{t1}X + u_{t0}$**:** $t_1 = f_3 + u_1, t_2 = f_1 + u_0, t_3 = f_0 + v_0, u_{t3} = 0, e = u_2u_2^{'}$; $u_{t2} = v_2^2 + e, u_{t1} = 1, u_{t0} = t_1^2 + u_1u_2^{'} + u_2u_{t2}$;	$3M + 2S$
3	**Compute** $V_t = v_{t3}X^3 + v_{t2}X^2 + v_{t1}X + v_{t0}$**:** $v_{t3} = t_1u_2 + u_{t2}, v_{t2} = v_2u_{t2} + v_1u_2^{'} + u_2, w = v_2u_{t0}$; $v_{t1} = (v_2 + u_2)(u_{t0} + 1) + w + t_2u_2^{'} + u_2, v_{t0} = w + t_3u_2^{'}$;	$7M$
4	**Reduce** U_t**, i.e.** $U_2 = X^3 + u_{22}X^2 + u_{21}X + u_{20}$**:** $u_{22} = v_{t3}^2, u_{21} = u_{t2}u_2^{'}, u_{20} = u_{22}u_{t2} + v_{t2}^2 + u_2^{'}, u_{22} = u_{22}u_2^{'}$;	$3M + 2S$
5	**Adjust:** $Z = eu_2^{'}, U_{22} = u_{22}u_2, U_{21} = u_{21}u_2, U_{20} = u_{20}u_2$;	$4M$
6	**Compute** $V_2 = v_{22}X^2 + v_{21}X + v_{20}$**:** $V_{22} = ev_{t2} + v_{t3}u_{22}, V_{21} = ev_{t1} + v_{t3}u_{21}, V_{20} = ev_{t0} + v_{t3}u_{20}$;	$6M$
Sum		$23M + 11S$