SCIS 2003 The 2003 Symposium on
Cryptography and Information Security
Hamamatsu,Japan, Jan.26-29,2003
The Institute of Electronics,
Information and Communication Engineers

A generalized Harley algorithm for genus two hyperelliptic curves

Hiroki SUGIZAKI * Kazuto MATSUO † Jinhui CHAO ‡ Shigeo TSUJII *

Abstract— A fast addition algorithm for divisor classes of genus two hyperelliptic curves over finite fields of odd characteristics was proposed by Harley in 2000. Since then several improvements of the algorithm have been developed. Besides, extensions of the algorithm to the curves over finite fields of characteristic two are proposed by these authors and Lange independently. However, it seems a Harley algorithm for arbitrary characteristics has not yet been available until now. This paper shows a generalization of the Harley algorithm to genus two hyperelliptic curves over finite fields of arbitrary characteristics.

Keywords: Hyperelliptic curve cryptosystems, genus two curves, finite fields of arbitrary characteristics, Addition algorithms, Harley algorithm

1 Introduction

Hyperelliptic curve cryptosystems can be practically implemented by using Cantor's fast addition algorithm for divisor class groups of hyperelliptic curves [1]. In 2000, a new fast addition algorithm for a genus two hyperelliptic curve

$$Y^2 = X^5 + f_4X^4 + \cdots + f_0,\ f_i \in \mathbb{F}_q \tag{1}$$

over a finite field $\mathbb{F}_q$ of odd characteristic was proposed by Harley [2, 3, 4]. Hereafter, we will call it the Harley algorithm.

The Harley algorithm extends the chord-tangent law of elliptic curves to addition of divisor classes of hyperelliptic curves. In this algorithm, a divisor class is represented by Mumford's representation [5] similar to the Cantor algorithm, but the input divisors are divided into different cases, computation procedures for each cases are optimized individually. Moreover, the Chinese remainder theorem, the Newton iteration and the Karatsuba multiplication are applied effectively to speed up of the algorithm. As the result, the Harley algorithm reduced significantly the computational cost comparing with the Cantor algorithm.

In fact, an improved Harley algorithm shown in [6] takes $I+25M$ for an addition and $I+28M$ (or $I+26M$ if $f_4 = 0$) for a doubling, where I, M denote the costs of an inversion and a multiplication over finite fields respectively.

Moreover, [7] showed a variation of Harley algorithm without inversions which is improved further in [8]. Besides, extensions of Harley algorithm to hyperelliptic curves over finite fields of characteristic two were shown by [6, 9] independently and [8] also showed an inversionfree version algorithm for characteristic two.

So far, extensions of the Harley algorithm have been developed separately for either odd characteristic case or even characteristic case. A generalization of the Harley algorithm for curves over finite fields of arbitrary characteristics, which is important for studies of hyperelliptic curve cryptosystems, has not yet been available until now.

This paper shows a generalization of the Harley algorithm for genus two hyperelliptic curves over finite fields of arbitrary characteristics. This algorithm takes $39A+I+28M+S$ for an addition and $59A+I+38M$ for a doubling, where A, S denote the costs of an addition and a squaring in a finite field respectively.

2 Preliminaries

2.1 Hyperelliptic curves

Let $\mathbb{F}_q$ be a finite field of arbitrary characteristic. A genus 2 hyperelliptic curve C over $\mathbb{F}_q$ is defined as follows:

$$\begin{aligned} C :& Y^2 + H(X)Y = F(X), \\ & H(X) = h_2X^2 + h_1X + h_0, \\ & F(X) = f_5X^5 + f_4X^4 + \cdots + f_1X + f_0, \end{aligned} \tag{2}$$

where $h_i, f_i \in \mathbb{F}_q$, and (2) satisfies $\{(x,y) \in \overline{\mathbb{F}}_q{}^2 \mid y^2 + H(x)y + F(x) = 2y + H(x) = H'(x)y - F'(x) = 0\} = \emptyset$.

The curve (2) can be transformed into the form

$$\begin{aligned} C/\mathbb{F}_q :& Y^2 + H(X)Y = F(X), \\ & H(X) = h_2X^2 + h_1X + h_0, \\ & F(X) = X^5 + f_4X^4 + \cdots + f_1X + f_0 \end{aligned} \tag{3}$$

by the coordinate transformation

$$(X, Y) \mapsto (f_5^{-1}X, f_5^{-3}Y). \tag{4}$$

* Department of Information and System Engineering, Chuo University, 1-13-27 Kasuga,Bunkyo-ku,Tokyo,112-8551 Japan

† Research and Development Initiative, Chuo University, 42-8 Ichigaya Honmura-cho, Shinjuku-ku, Tokyo, 162-8473 Japan

‡ Department of Electrical, Electronic and Communication Engineering, Chuo University, 1-13-27 Kasuga, Bunkyo-ku, Tokyo, 112-8551 Japan

Remark 1. The curve (3) can be further transformed into the form

$$C/\mathbb{F}_q : Y^2 + H(X)Y = F(X), \quad (5)$$
$$H(X) = X^2 + h_1X + h_0,$$
$$F(X) = X^5 + f_4X^4 + \cdots + f_1X + f_0$$

by the coordinate transformation

$$(X, Y) \mapsto (h_2^{-2}X, h_2^{-5}Y). \quad (6)$$

However, we will use (3) for the definition equation of C in order to keep a general representation of the inversion of a divisor as described 2.3.

Points on C are defined as

$$(x, y) \in \overline{\mathbb{F}_q}^2 \text{ s.t. } y^2 + H(x)y - F(x) = 0 \quad (7)$$

together with a point at infinity P_∞ on C.

For a point $P = (x, y)$, $-P = (x, -y - H(x))$ is also on C and it is called the opposite of P. The opposite of P_∞ is defined as $-P_\infty = P_\infty$.

A point P such that $P = -P$ is called a ramification point. A ramification point $P = (x, y)$ satisfies

$$2y + H(x) = 0. \quad (8)$$

Conversely, a point P is a ramification point if P satisfies (8) or $P = P_\infty$.

2.2 Divisors and Jacobian variety

A divisor $\mathcal{D}$ on C is defined as a finite formal sum of points on C

$$\mathcal{D} = \sum_{P_i \in C} \mathrm{ord}_{P_i}(\mathcal{D})P_i, \mathrm{ord}_{P_i}(\mathcal{D}) \in \mathbb{Z}. \quad (9)$$

Divisors form an Abelian group $\mathfrak{D}$.

The degree of $\mathcal{D}$ is defined as

$$\mathrm{deg}\mathcal{D} = \sum_{P_i \in C} \mathrm{ord}_{P_i}(\mathcal{D}). \quad (10)$$

Divisors with zero degree form a subgroup $\mathfrak{D}^0$ of $\mathfrak{D}$.

For a rational function f on C, a divisor (f) is defined as

$$(f) = \sum m_{P_i}P_i - \sum m_{Q_j}Q_j, \quad (11)$$

where P_i are zeros of f with multiplicities m_{P_i}, and Q_j are poles of f with multiplicities m_{Q_j}. (f) is called a principal divisor. The set of principal divisors is a subgroup $\mathfrak{D}^l$ of $\mathfrak{D}^0$.

The Jacobian variety of C is defined as

$$\mathcal{J}_C = \mathfrak{D}^0/\mathfrak{D}^l. \quad (12)$$

The divisor classes fixed by the qth-power Frobenius map form a subgroup $\mathcal{J}_C(\mathbb{F}_q)$ of $\mathcal{J}_C$. $\mathcal{J}_C(\mathbb{F}_q)$ is a finite Abelian group on which discrete logarithm problems can be defined.

This paper will consider addition of elements in $\mathcal{J}_C(\mathbb{F}_q)$.

2.3 Divisor classes and its representation

For $\mathcal{D}_1, \mathcal{D}_2 \in \mathfrak{D}^0$, if $\mathcal{D}_1 - \mathcal{D}_2 \in \mathfrak{D}^l$ then we write $\mathcal{D}_1 \sim \mathcal{D}_2$. Any divisor class in $\mathcal{J}_C(\mathbb{F}_q)$ can be represented by

$$\mathcal{D} = \sum_i m_iP_i - \left(\sum_i m_i\right) P_\infty, \ m_i \geq 0, \quad (13)$$

where $P_i \neq -P_j \ \forall \ i \neq j$. A divisor in the form of (13) is called a semi-reduced divisor. $\sum_i m_i$ is called the weight of $\mathcal{D}$.

A semi-reduced divisor whose weight is less than or equal to genus is called a reduced divisor. Any divisor class in $\mathcal{J}_C(\mathbb{F}_q)$ can be uniquely represented by a reduced divisor.

A semi-reduced divisor $\mathcal{D}$ can be represented by a pair of polynomials:

$$\mathcal{D} = (U, V), \quad (14)$$

where $U, V \in \overline{\mathbb{F}}_q[X]$. Denote $P_i = (x_i, y_i)$,

$$U = \prod (X - x_i)^{m_i}, \quad (15)$$

and V is the unique polynomial satisfies

$$F - HV - V^2 \equiv 0 \bmod U, \deg V < \deg U. \quad (16)$$

Moreover, for $P_i = (x_i, y_i)$ in (13),

$$y_i = V(x_i). \quad (17)$$

We call such a representation of $\mathcal{D}$ by (14),(15),(16) as Mumford's representation [5].

For $\mathcal{D} = (U, V)$, $U, V \in \mathbb{F}_q[X]$ is equivalent to $\mathcal{D} \in \mathcal{J}_C(\mathbb{F}_q)$. Therefore, we assume $U, V \in \mathbb{F}_q[X]$ hereafter.

For a weight two divisor $\mathcal{D} = (U, V)$,

$$-\mathcal{D} = (U, h_2U - V - H). \quad (18)$$

In particular, if $\mathcal{D} = P_r + P_r' - 2P_\infty$ where both P_r, P_r' are ramification points, then $-\mathcal{D} = \mathcal{D}$. For a weight one divisor $\mathcal{D} = (X + u_0, v_0)$,

$$-\mathcal{D} = (X + u_0, -v_0 - H(-u_0)). \quad (19)$$

3 Harley algorithm

This section outlines the Harley algorithm for the hyperelliptic curve (1) according to [2, 3, 4].

The semi-reduced divisors of (1) can be also represented by Mumford's representation similar to (3). Indeed, one can simply keep (15) but replace (16) with

$$F - V^2 \equiv 0 \bmod U, \deg V < \deg U, \quad (20)$$

and (18) with

$$-\mathcal{D} = (U, -V). \quad (21)$$

All I/O divisors $\mathcal{D}_1 = (U_1, V_1), \mathcal{D}_2 = (U_2, V_2), \mathcal{D}_3 = (U_3, V_3)$ are assumed to be reduced divisors. The Harley algorithm assigns different computation procedures to different input divisors for both operations of addition $\mathcal{D}_3 = \mathcal{D}_1 + \mathcal{D}_2$ and doubling $\mathcal{D}_2 = 2\mathcal{D}_1$.

In the addition operation, if the size of the finite field $\mathbb{F}_q$ is large enough, both weights of $\mathcal{D}_1$ and $\mathcal{D}_2$ almost always equal two. Besides, both $\mathcal{D}_1$ and $\mathcal{D}_2$ do not contain the same point or the points opposite to each other. In other words, $\mathcal{D}_1$ and $\mathcal{D}_2$ satisfy $\deg U_1 = \deg U_2 = 2$ and $\gcd(U_1, U_2) = 1$. The procedure of addition for divisors satisfied these conditions consists of a composition part and a reduction part.

In the composition part, a semi-reduced divisor $\mathcal{D} = (U, V)$ such that $\mathcal{D} \sim -\mathcal{D}_3$ and $U = U_1 U_2$ is computed. V can be obtained by the Chinese remainder theorem. In the reduction part, one computes the reduced divisor $\mathcal{D}_3$ such that $\mathcal{D}_3 \sim -\mathcal{D}$.

In the doubling operation, if the size of the finite field $\mathbb{F}_q$ is large enough, the weight of $\mathcal{D}_1$ almost always equals two. Besides, $\mathcal{D}_1$ does not contain ramification points except P_∞. In other words, $\deg U_1 = 2$ and $\gcd(U_1, V_1) = 1$. Similar to the addition, the doubling procedure for a divisor satisfied these conditions also consists of a composition part and a reduction part.

The composition part can be obtained by replacing the Chinese remainder theorem with the Newton iteration and the reduction part has the same procedure as of addition.

In the case when the input divisors of addition or doubling do not satisfy above conditions, the algorithm needs other procedures. Hence the first stage of the Harley algorithm classifies the input divisors according the weights of the divisors and the points contained in the divisors, then selects corresponding procedures. See [3, 6, 10] for details of these classification and procedures.

4 A most-frequent-case algorithm

This section shows a generalization of the Harley algorithm to hyperelliptic curve (3) over finite fields of arbitrary characteristics. Especially, we show a classification of the input divisor classes and procedures for the most frequent case.

Hereafter, elements of $\mathbb{F}_q$ and polynomials in X over $\mathbb{F}_q$ are denoted by small and capital letters respectively.

4.1 Classification for input divisor classes

The generalized Harley algorithm also follows the strategy of the Harley algorithm to apply different procedures for different cases of input divisors.

In addition $\mathcal{D}_3 = \mathcal{D}_1 + \mathcal{D}_2$, $\mathcal{D}_i = (U_i, V_i) \in \mathcal{J}_C(\mathbb{F}_q)$, we also make use of the facts that when the size of $\mathbb{F}_q$ is large enough, the weights of both $\mathcal{D}_1, \mathcal{D}_2$ almost always equal two, and both of $\mathcal{D}_1, \mathcal{D}_2$ do not contain the same point or the points opposite to each other. In other words, $\mathcal{D}_1, \mathcal{D}_2$ satisfy $\deg U_1 = \deg U_2 = 2$ and $\gcd(U_1, U_2) = 1$. We call such case "the most frequent case of addition" in this paper.

In the first stage of addition, one computes the resultant of U_1 and U_2 then uses it to classify input divisors. e.g. in the most frequent case of addition, the resultant is not equal to zero. In 4.2, we will show a procedure for the most frequent case of addition.

In doubling $\mathcal{D}_2 = 2\mathcal{D}_1$, $\mathcal{D}_i = (U_i, V_i) \in \mathcal{J}_C(\mathbb{F}_q)$, once again if the size of $\mathbb{F}_q$ is large enough, the weight of $\mathcal{D}_1$ almost always equals two. Besides, $\mathcal{D}_1$ does not contain ramification points except P_∞. In other words, $\mathcal{D}_1$ satisfies $\deg U_1 = 2$ and $\gcd(U_1, 2V_1 + H) = 1$. We call such case "the most frequent case of doubling" in this paper.

Similar to the procedures of addition, one computes the resultant of U_1 and $2V_1 + H$ at first. The case that the resultant is not equal to zero is the most frequent case of doubling. In 4.3, we will show a procedure for the most frequent case of doubling.

In both addition and doubling, when the input divisors do not satisfy the conditions of the most frequent case, the output divisor has to be computed by the procedures other than 4.2 and 4.3. These procedures can be easily obtained from the procedures shown in [3, 6] with minor modifications according to 4.2 and 4.3.

4.2 A most-frequent-case addition algorithm

Here, we show a procedure for the most frequent case of addition $\mathcal{D}_3 = \mathcal{D}_1 + \mathcal{D}_2$, $\mathcal{D}_i = (U_i, V_i) \in \mathcal{J}_C(\mathbb{F}_q)$.

Firstly, in the composition part, one computes a semi-reduced divisor $\mathcal{D} = (U, V)$ such that $\mathcal{D} \sim -\mathcal{D}_3$ and $U = U_1 U_2$. V is obtained as

$$V = SU_1 + V_1, \tag{22}$$

$$S \equiv (V_2 - V_1)U_1^{-1} \bmod U_2, \deg S \leq 1 \tag{23}$$

by applying the Chinese remainder theorem to

$$V \equiv V_1 \bmod U_1, \tag{24}$$

$$V \equiv V_2 \bmod U_2. \tag{25}$$

Next, in the first step of the reduction part, one computes the reduced divisor $\mathcal{D}_3' = (U_3', V_3')$ such that $\mathcal{D}_3' \sim \mathcal{D}$. U_3' is computed as

$$U_3' = s_1^{-2} \frac{F - HV - V^2}{U} \tag{26}$$

according to [7, 10]. In fact, when $s_1 = 0$ one needs another procedure which is omitted here. V_3' is the unique polynomial satisfied (16) for U_3' which can be obtained by

$$V_3' = S(U_3' - U_1) - s_1(u_{31}' - u_{11})U_3' + V_1 \tag{27}$$

from

$$V_3' \equiv V \bmod U_3' \tag{28}$$

and (22).

Finally, the output divisor $\mathcal{D}_3 = -\mathcal{D}_3'$ is obtained as

$$\mathcal{D}_3 = (U_3, V_3) = (U_3', h_2 U_3' - V_3' - H) \tag{29}$$

using (18).

The details of the procedure are optimized following [6, 7, 10]. e.g. rather than computing (22) explicitly, computation of U_3' is based on a formula obtained by substituting (22) into (26). Furthermore, the Karatsuba multiplication is used to reduce the computation cost further. Moreover, similar to [6, 7], two inversions required in the procedures can be replaced with one inversion and four multiplications using Montgomery's multiple inversion technique[11]. Consequently, we obtain a most-frequent-case addition algorithm which takes $39A + I + 28M + S$. Table 1 in Appendix shows further details of the proposed addition algorithm and the costs of each steps in the algorithm.

4.3 A most-frequent-case doubling algorithm

Here, we show a procedure for the most frequent case of doubling $\mathcal{D}_2 = 2\mathcal{D}_1, \mathcal{D}_i = (U_i, V_i) \in \mathcal{J}_C(\mathbb{F}_q)$.

Firstly, in the composition part, one computes the semi-reduced divisor $\mathcal{D} = (U, V)$ such that $\mathcal{D} \sim -\mathcal{D}_2$ and $U = U_1^2$. V is obtained as

$$V = SU_1 + V_1, \tag{30}$$

$$S \equiv \frac{F - HV_1 - V_1^2}{U_1}(2V_1 + H)^{-1} \bmod U_2, \deg S \leq 1 \tag{31}$$

by applying the Newton iteration to

$$V \equiv V_1 \bmod U_1. \tag{32}$$

In the reduction part, one follows the same steps as for addition. i.e., the output divisor $\mathcal{D}_2 = (U_2, V_2)$ is obtained as

$$U_2 = s_1^{-2}\frac{F - HV - V^2}{U} \tag{33}$$

$$V_2 = U_2(h_2 - s_1(u_{21} - u_{11})) - S(U_2 - U_1) - V_1 - H. \tag{34}$$

The details of the doubling procedure are also optimized following [6, 7, 10]. Consequently, we obtain a most-frequent-case doubling algorithm which takes $59A + I + 38M$.

Table 2 in Appendix shows further details of the proposed doubling algorithm and the costs of each steps in the algorithm.

5 Conclusion

This paper showed a generalized Harley algorithm for genus two hyperelliptic curves over finite fields of arbitrary characteristics. The proposed algorithm takes $39A+I+28M+S$ for an addition and $59A+I+38M$ for a doubling, where A, I, M, S denote the costs of an addition, an inversion, a multiplication and a squaring in finite field respectively.

Acknowledgement

A part of this research was supported by Telecommunications Advancement Organization of Japan (TAO).

References

[1] D. G. Cantor, *Computing in the Jacobian of hyperelliptic curve*, Math. Comp. **48** (1987), no.177, 95–101.

[2] P. Gaudry and R. Harley, *Counting points on hyperelliptic curves over finite fields*, ANTS-IV (W.Bosma, ed.), Lecture Notes in Computer Science, no. 1838, Springer-Verlag, 2000, pp.297–312.

[3] R. Harley, *adding.text*, `http://cristal.inria.fr/harley/hyper/`, 2000.

[4] R. Harley, *doubling.c*, `http://cristal.inria.fr/harley/hyper/`, 2000.

[5] D. Mumford, *Tata lectures on theta II*, Progress in Mathematics, no.43, Birkhäuser, 1984.

[6] H. Sugizaki, K. Matsuo, J. Chao, S. Tsujii, *A Fast Addition Algorithm of Genus Two Hyperelliptic Curves*, ISEC2002-9, IEICE Japan, 2002.

[7] Y. Miyamoto, H. Doi, K. Matsuo, J. Chao, S. Tsujii, *A Fast Addition Algorithm of Genus Two Hyperelliptic Curves*, Proc. of SCIS 2002, pp. 497–502, 2002 (in Japanese).

[8] T. Lange, *Weighted coordinates on genus 2 hyperelliptic curves*, Cryptology ePrint Archive, Report 2002/153, 2002, `http://eprint.iacr.org/`.

[9] T. Lange, *Efficient arithmetic on genus two Hyperelliptic Curves over Finite Fields via Explicit Formula*, Cryptology ePrint Archive, Report 2002/121, 2002, `http://eprint.iacr.org/`.

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

[11] H. Cohen, *A Course in Computational Algebraic Number Theory*, GTM 138, Springer, 1993, pp.481

Appendix

Input	Weight two reduced divisors $\mathcal{D}_1 = (U_1, V_1), \mathcal{D}_2 = (U_2, V_2)$ such that $\gcd(U_1, U_2) = 1$ $C : Y^2 + H(X)Y = F(X)$	
Output	A weight two reduced divisor $\mathcal{D}_3 = (U_3, V_3)$	
Step	Procedure	Cost
1	Compute the resultant r of U_1 and U_2.	$4A + 3M + S$
	$w_1 \leftarrow u_{11} - u_{21};\ w_0 \leftarrow u_{21}w_1 + u_{20} - u_{10};\ r \leftarrow (u_{20} - u_{10})w_0 + u_{20}w_1^2;$	
2	If $r = 0$ then call the other procedure.	—
3	Compute $I = w_1X + w_0 \equiv r(U_1)^{-1} \bmod U_2$.	—
4	Compute $T = t_1X + t_0 \equiv I(V_2 - V_1) \bmod U_2$.	$8A + 5M$
	$t_2 \leftarrow (v_{21} - v_{11})w_1;\ t_0 \leftarrow (v_{20} - v_{10})w_0;$	
	$t_1 \leftarrow (v_{21} - v_{11} + v_{20} - v_{10})(w_1 + w_0) - t_2 - t_0;$	
	$t_1 \leftarrow t_1 - t_2u_{21};\ t_0 \leftarrow t_0 - t_2u_{20};$	
5	If $t_1 = 0$ then call the sub-procedure.	—
6	Compute $S = s_1X + s_0$. (Multiple inversion technique)	$I + 6M$
	$w_0 \leftarrow (rt_1)^{-1};\ w_1 \leftarrow w_0r;\ w_2 \leftarrow w_0t_1;\ w_3 \leftarrow w_1r;\ s_1 \leftarrow w_2t_1;\ s_0 \leftarrow w_2t_0.$	
7	Compute $U_3 = X^2 + u_{31}X + u_{30} = s_1^{-2}((SU_1 + V_1)^2 + H(SU_1 + V_1) - F)/U_1U_2$.	$14A + 7M$
	$u_{31} \leftarrow u_{11} - u_{21} + w_3(2s_0 + h_2 - w_3);$	
	$u_{30} \leftarrow w_3(w_3(s_0(s_0 + h_2) + u_{21} - f_4 + u_{11}) - u_{21}(2s_0 + h_2) + 2v_{11} + h_1 + 2s_0u_{11})$	
	$+u_{10} - u_{20} + u_{21}(u_{21} - u_{11});$	
8	Compute $V_3 = v_{31}X + v_{30}$.	$13A + 5M$
	$w_1 \leftarrow u_{11} - u_{31};\ w_0 \leftarrow u_{10} - u_{30};\ w_2 \leftarrow s_1w_1;\ w_3 \leftarrow s_0w_0;$	
	$w_4 \leftarrow (s_1 + s_0)(w_1 + w_0) - w_2 - w_3;$	
	$v_{31} \leftarrow u_{31}(h_2 + w_2) - w_4 - v_{11} - h_1;\ v_{30} \leftarrow u_{30}(h_2 + w_2) - w_3 - v_{10} - h_0;$	
Total		$39A + I + 28M + S$

Table 1: A most-frequent-case Addition algorithm

Input	A weight two reduced divisor $\mathcal{D}_1 = (U_1, V_1)$ such that $\gcd(U_1, 2V_1 + H) = 1$ $C : Y^2 + H(X)Y = F(X)$	
Output	A weight two reduced divisor $\mathcal{D}_2 = (U_2, V_2)$	
Step	Procedure	Cost
1	Compute the resultant r of U_1 and $2V_1 + H$.	$8A + 7M$
	$h_3 \leftarrow h_2u_{11};\ w_2 \leftarrow h_2u_{10};\ w_1 \leftarrow h_3 - 2v_{11} - h_1;\ w_0 \leftarrow u_{11}w_1 + 2v_{10} - w_2 + h_0;$	
	$r \leftarrow (h_0 + 2v_{10})w_0 - u_{10}(h_2(2v_{10} + h_0 - w_2) + w_1(2v_{11} + h_1))$	
2	If $r = 0$ then call the other procedure.	—
3	Compute $I = -X + w_0 \equiv r(2V_1 + H)^{-1} \bmod U_1$.	—
4	Compute $T = t_1X + t_0 \equiv I(F - HV_1 - V_1^2)/U_1 \bmod U_1$.	$25A + 14M$
	$s_1 \leftarrow h_2v_{11};\ s_0 \leftarrow h_2v_{10};\ w_2 \leftarrow u_{11}(3u_{11} - 2f_4);\ w_3 \leftarrow v_{11}(h_1 + v_{11});\ w_4 \leftarrow 2u_{10}f_4;$	
	$w_5 \leftarrow u_{11}(6u_{10} + 2s_1 - 2f_3 + u_{11}(3f_4 - 4u_{11})) + f_2 - s_0 - w_3 - w_4;$	
	$w_6 \leftarrow w_2 - s_1 - 2u_{10} + f_3;\ t_1 \leftarrow w_1w_5 + w_0w_6;\ w_5 \leftarrow u_{10}w_6;$	
	$w_6 \leftarrow f_2 - w_3 - w_4 - s_0 + u_{11}(s_1 - f_3 + 4u_{10} + u_{11}(f_4 - u_{11}));\ t_0 \leftarrow w_0w_6 - w_1w_5;$	
5	If $t_1 = 0$ then call the sub-procedure.	—
6	Compute $S = s_1X + s_0$. (Multiple inversion technique)	$I + 6M$
	$w_0 \leftarrow (rt_1)^{-1};\ w_1 \leftarrow w_0r;\ w_2 \leftarrow w_0t_1;\ w_3 \leftarrow w_1r;\ s_1 \leftarrow w_2t_1;\ s_0 \leftarrow w_2t_0;$	
7	Compute $U_2 = X^2 + u_{21}X + u_{20} = s_1^{-2}((SU_1 + V_1)^2 + H(SU_1 + V_1) - F)/U_1^2$.	$11A + 4M$
	$u_{21} \leftarrow w_3(2s_0 + h_2 - w_3);\ u_{20} \leftarrow w_3(w_3(2u_{11} - f_4 + s_0(s_0 + h_2)) - h_3 + 2v_{11} + h_1);$	
8	Compute $V_2 = v_{21}X + v_{20}$.	$13A + 5M$
	$w_1 \leftarrow u_{11} - u_{21};\ w_0 \leftarrow u_{10} - u_{20};\ w_2 \leftarrow s_1w_1;\ w_3 \leftarrow s_0w_0;$	
	$w_4 \leftarrow (s_1 + s_0)(w_1 + w_0) - w_2 - w_3;$	
	$v_{21} \leftarrow u_{21}(h_2 + w_2) - w_4 - v_{11} - h_1;\ v_{20} \leftarrow u_{20}(h_2 + w_2) - w_3 - v_{10} - h_0;$	
Total		$59A + I + 38M$

Table 2: A most-frequent-case Doubling algorithm