Design of Hyperelliptic Cryptosystems in Small Characteristic and a Software Implementation over $\mathbf{F}_{2^n}$

Yasuyuki Sakai[1] and Kouichi Sakurai[2]*

[1] Mitsubishi Electric Corporation,
5-1-1 Ofuna, Kamakura, Kanagawa 247, Japan
e-mail: ysakai@iss.isl.melco.co.jp
[2] Kyushu University,
6-10-1 Hakozaki, Higashi-ku, Fukuoka 812-81, Japan
e-mail: sakurai@csce.kyushu-u.ac.jp

Abstract. We investigate the discrete logarithm problem over jacobians of hyperelliptic curves suitable for public-key cryptosystems. We focus on the case when the definition field has small characteristic 2, 3, 5 and 7, then we present hyperelliptic cryptosystems that resist against all known attacks. We further implement our designed hyperelliptic cryptosystems over finite fields $\mathbf{F}_{2^n}$ in software on Alpha and Pentium-II computers. Our results indicate that if we choose curves carefully, hyperelliptic cryptosystems do have practical performance.

1 Introduction

1.1 Hyperelliptic Cryptosystems

Koblitz [Ko88,Ko89] investigated jacobians of hyperelliptic curves defined over finite fields as a source of finite abelian groups suitable for cryptographic discrete logarithm problems. As a motivation of the cryptographic research, Koblitz gave the following conjectural remark [Ko88, page–99]: *"Thus, as far as we know, discrete log cryptosystems using* $\mathbf{J}(\mathbf{F}_{p^n})$ *seem to be secure for relatively small* p^n *(even when* $p = 2$*). From the standpoint of implementation, this feature may outweigh the added time required to compute the more complicated group operation."*

Frey and Rück's generalization [FR94] of MOV-attack [MOV93] solved in subexponential time the discrete logarithm problems over some of Koblitz's designed hyperelliptic cryptosystems [Ko89]. However, Sakai, Sakurai and Ishizuka designed hyperelliptic cryptosystems [SSI98] that resist against all known attacks including the Frey and Rück's method [FR94]. Furthermore, Sakai et al. [SSI98] analyzed the computational complexity on the group operation in jacobians. Their results theoretically support the Koblitz's conjecture referred to above.

* Partially done while visiting in Columbia Univ. Computer Science Dept.

K. Ohta and D. Pei (Eds.): ASIACRYPT'98, LNCS 1514, pp. 80–94, 1998.

In this work, we further explores hyperelliptic discrete logarithms for obtaining more efficient public key cryptosystems, and confirms experimentally the Koblitz's conjecture on the practical merit of hyperelliptic cryptosystems.

1.2 Our Investigated Topics

We consider the following topics to address as challenging problems after [Ko88] [Ko89,SSI98].

1. *Designing secure hyperelliptic cryptosystems with genus 2 curves over small characteristic fields*
   Koblitz [Ko88,Ko89] presented jacobians of curves $C : v^2 + h(u)v = f(u)$, where $deg(f(u)) = 5$ (genus 2), defined over $\mathbf{F}_2$. However, some discrete logarithms of the curves had been broken by Frey and Rück [FR94]. As a negative result, Sakai et al. [SSI98] experimentally showed that no secure curve exists with genus 2 among those defined over $\mathbf{F}_2$ and $h(u) = 1$.
   Recently developed other methods [BK98,Fr97,CMT97] have generated secure hyperelliptic cryptosystems with genus 2. However, these require the size of the characteristic of curve's definition field to be large.
2. *Designing secure hyperelliptic cryptosystems over* $\mathbf{F}_{2^n}$ *with smaller* n
   Sakai et al. [SSI98] examined jacobians over $\mathbf{F}_{2^n}$ with genus $g = 3, 11$ curves $v^2 + v = u^{2g+1}$, which resist against all known attacks. However, their construction requires a large extension-degree n. For example, for achieving the security as RSA with 1024-bit key, the jacobian of the curve $v^2+v = u^7$ must be defined over $\mathbf{F}_{2^{59}}$ or larger fields. They also confirmed that the jacobian of the curve $v^2 + v = u^{23}$ (genus 11) over $\mathbf{F}_{2^{47}}$ induces a secure hyperelliptic cryptosystem with the same level of security as RSA with 5000-bit key. This can be efficiently (without multi-precision library) implemented via software on 64-bit CPU (e.g. Alpha).
   However, no secure hyperelliptic cryptosystem is available from this curve with smaller n than 47. We want such a jacobian over $\mathbf{F}_{2^n}$ with hopefully $n \leq 32$ for an efficient software implementation on 32-bit CPU (e.g. Pentium).
3. *Implementing hyperelliptic cryptosystem in software*
   Indeed, the formulas for adding divisors in a jacobian are more complex compared to formulas for adding points in an elliptic curve. However, as we first remarked, Koblitz [Ko88] suggested that hyperelliptic cryptosystems defined over a small definition field may be efficient in practice.
   Sakai et al. [SSI98] evaluated encryption/decryption speed which should that hyperelliptic cryptosystems are indeed practical. However, their confirmation was only theoretical, and no performance via a practical implementation has been reported.

1.3 Our Results

On Design with Genus Two In the case of characteristic 2, we have found secure jacobians by considering a more wider class of $h(u)$ (degree of $h(u)$ has at most g). Moreover, in the case of $C : v^2 = f(u)$ over characteristic 3, 5 and 7 finite fields, where $deg(f(u)) = 5$ (genus 2), we have found many curves which resist against all known attacks.

On Design with Smaller Size of $\mathbf{F}_{2^n}$ By not choosing curves from $v^2 + v = u^{2g+1}$ but from a wider class $v^2 + v = f(u)$, we have found secure jacobians over $\mathbf{F}_{2^n}$ with "$n \leq 32$" that achieve the same (or higher) level of security as RSA with 1024-bit key.

On Implementation We have implemented operations in jacobians via software. One platform was Alpha 21164A (467MHz) with 64-bit word size, and another was Pentium-II (300MHz) with 32-bit word size. Programs were written in C-language and compiled with GCC. Our software implementation of secure jacobians, which have the same level of security as RSA with 1024-bit key, achieve good practical performance. In an exponentiation of a randomly chosen divisor, the jacobian over $\mathbf{F}_{2^{59}}$ of the genus 3 curve $C : v^2 + v = u^7$ achieved 83.3 msec. on Alpha 21164A (467MHz), and the jacobian over $\mathbf{F}_{2^{29}}$ of the genus 6 curve $C : v^2 + v = u^{13} + u^{11} + u^7 + u^3 + 1$ achieved 476 msec. on Pentium-II (300MHz). We have also implemented secure jacobians which have the same level of security as RSA with 5000-bit key. In an exponentiation of a randomly chosen divisor, the jacobian over $\mathbf{F}_{2^{47}}$ of the genus 11 curve $C : v^2 + v = u^{23}$ achieved 1.74 sec. on Alpha 21164A (467MHz). Note that those jacobians can be implemented without "a multi-precision library", because of the size of the definition fields.

1.4 Our Approach

Our Considered Security We design hyperelliptic cryptosystems that resist against the following four known attacks:

1. The Pohlig-Hellman method [PH78].
2. Frey-Rück's generalization [FR94] of the Menezes-Okamoto-Vanstone attack [MOV93].
3. Adleman-DeMarrais-Huang's *smooth-divisor-attack* [ADH94].
4. Rück's generalization [Ru97] of the Semaev-Smart-Satoh-Araki attack [Sem98,Sm97,SA97] on elliptic curves with Frobenius trace one.

Our design further notes new attacks improving the parallerized Pollard-Lambda search [WZ98,GLV98].

On Choosing Curves and Counting the Order of their Jacobian In [Ko88], Koblitz investigated the jacobians of the hyperelliptic curves $v^2 + v = u^{2g+1}$ over finite field for cryptographically intractable discrete logarithm. In [Ko89], Koblitz also discussed the jacobians of the hyperelliptic curves of more general form $v^2 + h(u)v = f(u)$, however, the degree of the polynomial $f(u)$ is restricted to be 5 (i.e. genus 2) and the definition fields are only the case of characteristic 2. In order to obtain a broader class of jacobians suitable for secure discrete logarithms, we deal with a wider family of the hyperelliptic curves $v^2 + v = f(u)$ and $v^2 = f(u)$ over finite field of characteristic 2, 3, 5 and 7, where $\deg(f(u)) = 2g + 1$.

2 Preliminaries

In this section, we give a brief description of jacobians. See [Ko98] for more detail.

Let $\mathbf{F}$ be a finite field and let $\bar{\mathbf{F}}$ be the algebraic closure of $\mathbf{F}$. A hyperelliptic curve C of genus g over $\mathbf{F}$ is an equation of the form $C : v^2 + h(u)v = f(u)$ in $\mathbf{F}[u, v]$, where $h(u) \in \mathbf{F}[u]$ is a polynomial of degree at most g, $f(u) \in \mathbf{F}[u]$ is a monic polynomial of degree $2g + 1$, and there are no solutions $(u, v) \in \bar{\mathbf{F}} \times \bar{\mathbf{F}}$ which simultaneously satisfy the equation $v^2 + h(u)v = f(u)$ and the partial derivative equations $2v + h(u) = 0$ and $h'(u)v - f'(u) = 0$. Thus, a hyperelliptic curve does not have singular points.

A divisor on C is a finite formal sum of $\bar{\mathbf{F}}$-points $D = \sum m_i P_i$, $m_i \in \mathbf{Z}$. We define the degree of D to be $\deg(D) = \sum m_i$. If $\mathbf{K}$ is an algebraic extension of $\mathbf{F}$, we say that D is defined over $\mathbf{K}$ if for every automorphism σ of $\bar{\mathbf{F}}$ that fixes $\mathbf{K}$ one has $\sum m_i P_i^\sigma = D$, where P^σ denotes the point obtained by applying σ to the coordinates of P (and $\infty^\sigma = \infty$). Let $\mathbf{D}$ denote the additive group of divisors defined over $\mathbf{K}$ (where $\mathbf{K}$ is fixed), and let $\mathbf{D}^0$ denote the subgroup consisting of divisors of degree 0. The principal divisors form a subgroup $\mathbf{P}$ of $\mathbf{D}^0$. $\mathbf{J}(\mathbf{K}) = \mathbf{D}^0/\mathbf{P}$ is called the "*jacobian*" of the curve C. In this paper, we denote $\mathbf{J}(C; \mathbf{K})$ also the jacobian defined over $\mathbf{K}$ of the curve C.

The discrete logarithm problem on $\mathbf{J}(C; \mathbf{K})$ is the problem, given two divisors $D_1, D_2 \in \mathbf{J}(C; \mathbf{K})$ of determining an integer m such that $D_2 = mD_1$ if such m exists.

3 Security Against Known Attacks

We will choose jacobians to satisfy the following four conditions to resist against all known attacks.

C1 : $\sharp\mathbf{J}(C; \mathbf{F}_q)$ is divisible by a large prime
C2 : $\mathbf{J}(C; \mathbf{F}_q)$ can not be imbedded into a small finite field $\mathbf{F}_{q^k}$
C3 : $2g + 1 \le \log q$
C4 : Jacobian over a field of characteristic p has not a cyclic group structure of order p^n for small n.

Our design further notes new attacks improving the parallerized Pollard-Lambda search [WZ98,GLV98].

3.1 C1 : General Algorithms

The condition **C1** is to resist Pohlig-Hellman method [PH78]. The algorithm has a running time that is proportional to the square root of the largest prime factor of $\sharp\mathbf{J}(C; \mathbf{F}_q)$. Therefore, we need to choose curves such that $\sharp\mathbf{J}(C; \mathbf{F}_q)$ has a large prime factor.

3.2 C2 : Imbedding into a Small Finite Field

The condition **C2** is to resist Frey and Rück's generalization [FR94] of MOV-attack [MOV93] using Tate pairing. Their method reduces the logarithm problem over $\mathbf{J}(C; \mathbf{F}_q)$ to the logarithm problem over an extension field $\mathbf{F}_{q^k}$. Methods of avoiding MOV-attack have been discussed in [BS91,CTT94]. We take the similar approach by choosing curves such that the induced jacobian $\mathbf{J}(C; \mathbf{F}_q)$ cannot be imbedded via Tate pairing into $\mathbf{F}_{q^k}$ with small extension degree k. Therefore, we replace **C2** by the following sufficient condition:

C2' : The largest prime factor of $\sharp\mathbf{J}(C; \mathbf{F}_q)$ does not divide $(q)^k - 1$, $k < (\log q)^2$

3.3 C3 : Large Genus Hyperelliptic Curves

The condition **C3** is to resist Adleman-DeMarrais-Huang method [ADH94]. They found a sub-exponential algorithm for discrete logarithm over the rational subgroup of the jacobians of large genus hyperelliptic curves over finite fields. It is a heuristic algorithm under certain assumptions. Therefore, we need to choose curves such that the genus of curves is not so large.

3.4 C4 : Additive Embedding Attack

The condition **C4** is to resist Rück's generalization [Ru97] of the Semaev-Smart-Satoh-Araki attack [Sem98,Sm97,SA97] on elliptic cryptosystems with Frobenius trace one. The method uses an additive version of Tate pairing to solve the discrete logarithm of a jacobian over a finite field of characteristic p and has the running time $O(n^2 \log p)$ for a jacobian with cyclic group structure of order p^n.

We should remark that our design is in small characteristic $p = 2, 3, 5$ and 7. Subgroups of the jacobians that we consider have order prime to the characteristic p. Therefore, this additive embedding attack does not apply to our cryptosystems.

3.5 Improved Parallerized Pollard-Lambda Search

New attacks have been announced, which improved the parallerized Pollard-Lambda search [WZ98,GLV98]. For elliptic curves over $\mathbf{F}_{2^n}$ with coefficients in $\mathbf{F}_2$, this attacking time can be reduced by a factor of the square root of $2n$. For example, the time required to compute an elliptic curve logarithm on such a curve over $\mathbf{F}_{2^{163}}$ is reduced from the previous 2^{81} to 2^{77} elliptic curve operations.

This could be applicable to our designed hyperelliptic cryptosystems in characteristic 2, because the coefficients of our curves belong to $\mathbf{F}_2$. We should note that the power of this attack is not so strong as the four listed above. However, this attack is very important to our selection of the size of security-parameter, which effects the performance analysis of our cryptosystems. Therefore, we should consider the security against possible extension of this kind of attack in our design of hyperelliptic cryptosystems.

4 Our Order Counting Method

Beth and Schaefer [BS91] used zeta-function for their constructing elliptic cryptosystems and Koblitz [Ko88,Ko89,Ko98] also used zeta-function of a hyperelliptic curve to construct jacobians of hyperelliptic curves defined over finite fields.

A technical difficulty in our computation on general hyperelliptic curves is that the zeta-function has a complicated form with larger degree. Therefore, it is not easy to compute its exact solutions unlike the previous cases [BS91,Ko88,Ko89,Ko98]. However, it is known that the order of a jacobian can be computed without deciding the solution of its zeta-function [St93, Chapter V]. Therefore, we use this algorithm for our problems.

Throughout this section, F denotes an algebraic function field of genus g whose constant field is the finite field $\mathbf{F}_q$ and $\mathbf{P}$ denotes the set of places of

$F/\mathbf{K}$. The definition, the theorem and the corollary shown below are given in the article [St93].

Definition 1. *[St93] The polynomial $L(t) := (1-t)(1-qt)Z(t)$ is called the L-polynomial of function field $F/\mathbf{F}_q$, where $Z(t)$ denotes the zeta-function of $F/\mathbf{F}_q$.*

Theorem 1. *[St93]*

(a) $L(t) \in \mathbf{Z}[t]$ and deg $L(t) = 2g$
(b) $L(t) = q^g t^{2g} L(1/qt)$
(c) $L(1) = h$, the class number of $F/\mathbf{F}_q$
(d) We write $L(t) = \sum_{i=0}^{2g} a_i t^i$. Then the following holds:
  (1) $a_0 = 1$ and $a_{2g} = q^g$.
  (2) $a_{2g-i} = q^{g-i} a_i$ for $0 \leq i \leq g$.
  (3) $a_1 = N - (q+1)$ where N is the number of places $P \in \mathbf{P}_F$ of degree one.
(e) $L(t)$ factors in $\mathbf{C}[t]$ in the form $L(t) = \prod_{i=1}^{2g}(1-\alpha_i t)$. The complex numbers $\alpha_1, \cdots, \alpha_{2g}$ are algebraic integers, and they can be arranged in such a way that $\alpha_i \alpha_{g+i} = q$ holds for $i = 1, \cdots, g$.
(f) If $L_r(t) := (1-t)(1-q^r t)Z_r(t)$ denotes the L-polynomial of the constant field extension $F_r = F\mathbf{F}_{q^r}$, then $L_r(t) = \prod_{i=1}^{2g}(1-\alpha_i t)$

Corollary 1. *[St93] Let $S_r := N_r - (q^r + 1)$. Then we have:*
$a_0 = 1$, and $i a_i = S_i a_0 + S_{i-1} a_1 + \cdots + S_1 a_{i-1}$, for $i = 1, \cdots, g$.

We can determine the order of jacobians by the Theorem and the Corollary in the following algorithm. We should note that it is easy to count $N_1, \cdots, N_g$ if $\mathbf{F}_q$ is small.

```
Order Counting
Input   Hyperelliptic curve $C : v^2 + h(u)v = f(u)$ over $\mathbf{F}_q$
        and extension degree $n$
Output  The order $\sharp\mathbf{J}(C;\mathbf{F}_{q^n})$
Step1   Determine $N_r = \sharp\mathbf{J}(C;\mathbf{F}_{q^r})$, for $r = 1, \cdots, g$
        by counting the number of rational points of $C$ over $\mathbf{F}_{q^r}$
Step2   Determine the coefficients of $L_{\mathbf{F}_q}(t) = \sum_{i=0}^{2g} a_i t^i$ in the following:
            $a_0 = 1$
            for $1 \leq i \leq g$:        $a_i = (\sum_{k=1}^{i}(N_k - (q^k + 1))a_{i-k})/i$
            for $g+1 \leq i \leq 2g$:   $a_i = q^{i-g} a_{2g-i}$
Step3   Compute $L_{\mathbf{F}_{q^n}}(1) = \prod_{k=1}^{n} L_{\mathbf{F}_q}(\zeta^k)$,
        where $\zeta$ runs over the $n$-th root of unity
Step4   Return $\sharp\mathbf{J}(C;\mathbf{F}_{q^n}) = L_{\mathbf{F}_{q^n}}(1)$
```

5 Jacobians over Finite Fields of Characteristic 2

Koblitz [Ko88,Ko89] considered the security of the discrete logarithm problem over jacobians of genus 2 curves when the definition fields have characteristic 2. However, Frey and Rück [FR94] generalized MOV-reduction to hyperelliptic curves. They have found that some of hyperelliptic cryptosystems presented by Koblitz [Ko89] are breakable in sub-exponential time. In this section, we discuss the security of genus 2 curves which have the form $v^2 + h(u)v = f(u)$ defined over characteristic 2 finite fields. We also discuss the security of genus 3, 4, 5 and 6 curves.

5.1 Genus 2 Curves

First, we examine the order of jacobians $\sharp\mathbf{J}(C; \mathbf{F}_{2^n})$ in the case of $h(u) = 1$, where the degree of $f(u)$ equals to 5. We also examine their factorizations.

Extension degree n of $\mathbf{F}_{2^n}$ were examined from 59 to 89. The reason is that: $\sharp\mathbf{J}(C; \mathbf{F}_{2^{59}})$ has the size of 119-bit. $\sharp\mathbf{J}(C; \mathbf{F}_{2^{89}})$ has the size of 179-bit. Namely, if the jacobians are secure, their level of security are in the range from approximately RSA-512 to RSA-1024. [1] ("RSA-n" denotes RSA with n-bit key.) We have examined whether P_{max} of $\sharp\mathbf{J}(C; \mathbf{F}_{2^n})$ divide $(2^n)^k - 1$ to confirm the security condition **C2'**. (P_{max} denotes the largest prime factor of $\sharp\mathbf{J}(C; \mathbf{F}_q)$.) As a result, for example, in the case of $f(u) = u^5 + u^3$, $\sharp\mathbf{J}(C; \mathbf{F}_{2^{89}})$ has the size of 179-bit and its P_{max} has the size of 134-bit. However, P_{max} divides $(2^n)^{12} - 1$. Therefore, the jacobian does not satisfy **C2'** (see also [FR94]).

In the case of $h(u) = 1$, We have failed to obtain secure jacobians, which satisfy **C1** and **C2'**. However, in [Ko98], Koblitz examined the case of $h(u) = u$ and showed examples of secure jacobians. We have examined the case of more wider classes such that $h(u)$ has degree at most g. As a result, $\sharp\mathbf{J}(C; \mathbf{F}_{2^{89}})$ of $C : v^2 + (u^2 + u + 1)v = u^5 + u + 1$ has the size of 179-bit. Its P_{max} has the size of 178-bit. We also confirmed the jacobian satisfies **C2'**. The factorization of the jacobian is given in Appendix A.

5.2 Curves of Genus Larger than 2

Next, we examine $\sharp\mathbf{J}(C; \mathbf{F}_{2^n})$ and their factorizations in the case of curves $C : v^2 + v = f(u)$ have genus 3, 4, 5 or 6, where degree of $f(u)$ equals to 7, 9, 11 or 13, respectively.

Table 1 shows the list of the size of $\sharp\mathbf{J}(C; \mathbf{F}_{2^n})$ and the size of P_{max}. The factorizations are given in Appendix A. Extension degree n of $\mathbf{F}_{2^n}$ were examined by n such that $\sharp\mathbf{J}(C; \mathbf{F}_{2^n})$ has the size of larger than 160-bit. Namely, if listed jacobians are secure, their level of security are approximately same as RSA-1024 or with a larger key. The listed equations of curves C have largest P_{max} in fixed extension degree n. In the case of genus 5, all prime factors of $\sharp\mathbf{J}(C; \mathbf{F}_{2^{37}})$ have much smaller size than 160-bit. Therefore, $\mathbf{J}(C; \mathbf{F}_{2^{41}})$ are listed.

We have examined whether P_{max} of $\sharp\mathbf{J}(C; \mathbf{F}_{2^n})$ divide $(q^n)^k - 1$ to confirm the security condition **C2'**. All listed $\sharp\mathbf{J}(C; \mathbf{F}_{2^n})$ satisfy **C2'**. Namely, P_{max} does

[1] The notation "*same level of security*" is based on the following: One of the most efficient algorithm of integer factoring is the number field sieve method. The method takes $exp(c(\ln n)^{1/3}(\ln\ln n)^{2/3})$ time, where $1.5 < c < 1.9$ and n denotes the size of an integer. On the other hand, Pohlig-Hellman method, which is an efficient algorithm for discrete logarithm problem for elliptic curve, takes $\sqrt{P_{max}}$. Therefore, for example, EC-160 has approximately same level of security as RSA-1024.

genus	$\mathbf{J}$	$C : v^2 + v = f(u)$	size of $\sharp\mathbf{J}$	size of P_{max}
3	$\mathbf{J}(C; \mathbf{F}_{2^{59}})$	$f(u) = u^7$	178-bit	165-bit
4	$\mathbf{J}(C; \mathbf{F}_{2^{41}})$	$f(u) = u^9 + u^7 + u^3 + 1$	164-bit	161-bit
5	$\mathbf{J}(C; \mathbf{F}_{2^{41}})$	$f(u) = u^{11} + u^5 + u + 1$	205-bit	201-bit
6	$\mathbf{J}(C; \mathbf{F}_{2^{29}})$	$f(u) = u^{13} + u^{11} + u^7 + u^3 + 1$	174-bit	170-bit

Table 1. Jacobians over char 2 finite fields of genus 3,4,5 and 6 curves

not divide $(q^n)^k - 1$ with small k. Therefore, the curves shown in Table 1 are secure and have the same or higher level of security as RSA-1024. We implement group operations of the jacobians in software in a later section.

6 Jacobians over Finite Fields of Characteristic Larger than Two

In this section, we examine genus 2 curves over characteristic 3, 5 and 7 finite fields. Moreover, we examine genus 3 and 4 curves.

6.1 Genus 2 Curves

First, we examine the curve $C : v^2 = f(u)$, where $f(u)$ has degree 5. Tables 2, 3 and 4 show the list of the size of $\sharp\mathbf{J}(C; \mathbf{F}_{p^n})$ and the size of P_{max} in the case of $p = 3, 5, 7$, respectively. Tabulated are in the case that the coefficients of the curves are in {0,1}. The factorizations of $\sharp\mathbf{J}(C; \mathbf{F}_{p^n})$ are given in Appendix A.

In the case of characteristic 3, extension degree n of $\mathbf{F}_{3^n}$ were examined from 37 to 59. $\sharp\mathbf{J}(C; \mathbf{F}_{3^{37}})$ has the size of 118-bit. $\sharp\mathbf{J}(C; \mathbf{F}_{3^{59}})$ has the size of 188-bit. As in the last section, if listed jacobians are secure, their levels of security are in the range from approximately RSA-512 to RSA-1024. The listed equations of curves C have largest P_{max} in fixed extension degree n. As in the case of characteristic 3, extension degree n of $\mathbf{F}_{5^n}$ was examined from 23 to 43, and extension degree n of $\mathbf{F}_{7^n}$ was examined from 19 to 37.

We have examined whether P_{max} of $\sharp\mathbf{J}(C; \mathbf{F}_{p^n})$ divide $(p^n)^k - 1$ to confirm the security condition **C2'**. All listed $\sharp\mathbf{J}(C; \mathbf{F}_{p^n})$ in tables 2, 3 and 4 satisfy **C2'**. Namely, P_{max} does not divide $(p^n)^k - 1$ with small k.

6.2 Curves of Genus Larger than 2

Next, we examine the order of jacobians $\sharp\mathbf{J}(C; \mathbf{F}_{p^n})$ and their factorizations of genus 3 and 4 curves $C : v^2 = f(u)$, where degree of $f(u)$ equals to 7 and 9, respectively.

Table 5 shows the list of the size of $\sharp\mathbf{J}(C; \mathbf{F}_{p^n})$ and the size of the largest prime factor of $\sharp\mathbf{J}(C; \mathbf{F}_{p^n})$. As in the case of genus 2, there exist secure jacobians which satisfy the condition **C2'**. The factorizations of $\sharp\mathbf{J}(C; \mathbf{F}_{p^n})$ are given in Appendix A.

$\mathbf{J}$	C	size of $\sharp\mathbf{J}$	size of P_{max}
$\mathbf{J}(C;\mathbf{F}_{3^{37}})$	$v^2 = u^5 + u^3 + u + 1$	118-bit	97-bit
$\mathbf{J}(C;\mathbf{F}_{3^{41}})$	$v^2 = u^5 + u^2 + u + 1$	130-bit	116-bit
$\mathbf{J}(C;\mathbf{F}_{3^{43}})$	$v^2 = u^5 + u^4 + 1$	137-bit	118-bit
$\mathbf{J}(C;\mathbf{F}_{3^{47}})$	$v^2 = u^5 + u^3 + u + 1$	149-bit	135-bit
$\mathbf{J}(C;\mathbf{F}_{3^{53}})$	$v^2 = u^5 + u^4 + u + 1$	169-bit	147-bit
$\mathbf{J}(C;\mathbf{F}_{3^{59}})$	$v^2 = u^5 + u^4 + u^3 + u + 1$	188-bit	185-bit

Table 2. genus 2 curves over char 3 fields

$\mathbf{J}$	C	size of $\sharp\mathbf{J}$	size of P_{max}
$\mathbf{J}(C;\mathbf{F}_{5^{23}})$	$v^2 = u^5 + u^4 + u^3 + 1$	107-bit	103-bit
$\mathbf{J}(C;\mathbf{F}_{5^{29}})$	$v^2 = u^5 + u^4 + u^3 + u + 1$	135-bit	129-bit
$\mathbf{J}(C;\mathbf{F}_{5^{31}})$	$v^2 = u^5 + u^2 + 1$	144-bit	140-bit
$\mathbf{J}(C;\mathbf{F}_{5^{37}})$	$v^2 = u^5 + u^4 + u^3 + 1$	172-bit	149-bit
$\mathbf{J}(C;\mathbf{F}_{5^{41}})$	$v^2 = u^5 + u^4 + u^3 + 1$	191-bit	118-bit
$\mathbf{J}(C;\mathbf{F}_{5^{43}})$	$v^2 = u^5 + u^2 + 1$	200-bit	196-bit

Table 3. genus 2 curves over char 5 fields

$\mathbf{J}$	C	size of $\sharp\mathbf{J}$	size of P_{max}
$\mathbf{J}(C;\mathbf{F}_{7^{19}})$	$v^2 = u^5 + u^4 + u^3 + u^2 + 1$	107-bit	76-bit
$\mathbf{J}(C;\mathbf{F}_{7^{23}})$	$v^2 = u^5 + u^4 + u^3 + u + 1$	130-bit	118-bit
$\mathbf{J}(C;\mathbf{F}_{7^{29}})$	$v^2 = u^5 + u^4 + u^2 + 1$	164-bit	157-bit
$\mathbf{J}(C;\mathbf{F}_{7^{31}})$	$v^2 = u^5 + u^4 + u^3 + u^2 + 1$	175-bit	154-bit
$\mathbf{J}(C;\mathbf{F}_{7^{37}})$	$v^2 = u^5 + u^3 + u^2 + u + 1$	208-bit	203-bit

Table 4. genus 2 curves over char 7 fields

char	genus	$\mathbf{J}$	C	size of $\sharp\mathbf{J}$	size of P_{max}
3	3	$\mathbf{J}(C;\mathbf{F}_{3^{37}})$	$v^2 = u^7 + u^5 + u^3 + u^2 + 1$	176-bit	171-bit
	4	$\mathbf{J}(C;\mathbf{F}_{3^{29}})$	$v^2 = u^9 + u^6 + u^5 + u^3 + 1$	184-bit	178bit
5	3	$\mathbf{J}(C;\mathbf{F}_{5^{23}})$	$v^2 = u^7 + u^6 + u^2 + 1$	161-bit	154-bit
	4	$\mathbf{J}(C;\mathbf{F}_{5^{19}})$	$v^2 = u^9 + u^7 + u^6 + u^5 + u^3 + u + 1$	177-bit	168-bit
7	3	$\mathbf{J}(C;\mathbf{F}_{7^{19}})$	$v^2 = u^7 + u^6 + u^5 + u^3 + u + 1$	161-bit	152-bit
	4	$\mathbf{J}(C;\mathbf{F}_{7^{17}})$	$v^2 = u^9 + u^8 + u^6 + u^5 + u^3 + u + 1$	191-bit	181-bit

Table 5. genus 3 and 4 curves over char 3, 5 and 7 fields

7 Implementation and Timings

In this section, we show software implementation and timings of group operations in jacobians over characteristic 2 finite fields obtained in previous sections.

7.1 Computing in Jacobians

we show here an algorithm for addition and doubling of elements $D \in \mathbf{J}(C;\mathbf{F}_{2^n})$. A divisor D is regarded simply as a pair of polynomials $D =$ div $(a(u), b(u))$ such that deg $b <$ deg a and deg $a \leq g$. We give here a brief description of the algorithm for the addition: $D_3 = D_1 + D_2$, where $D_3 = \mathrm{div}(a_3, b_3)$, $D_1 = \mathrm{div}(a_1, b_1)$, $D_2 = \mathrm{div}(a_2, b_2)$ (see [Ca87,Ko89] for more details).

```
Addition
Input:   two divisors $D_1 = \mathrm{div}(a_1,b_1)$, $D_2 = \mathrm{div}(a_2,b_2) \in \mathbf{J}$
Output:  $D_3 = \mathrm{div}(a_3,b_3) = D_1 + D_2$
Step A1  Compute $d_1, s_1$ and $s_2$ which satisfy
             $d_1 = \gcd(a_1,a_2)$ and $d_1 = s_1a_1 + s_2a_2$
Step A2  If $d_1 = 1$ then
             $a := a_1a_2,\ b := s_1a_1b_2 + s_2a_2b_1 \pmod{a}$
       else
             Compute $d_2, s_1', s_2'$ and $s_3$ which satisfy
                 $d_2 = \gcd(d_1, b_1 + b_2 + h)$ and $d_2 = s_1'a_1 + s_2'a_2 + s_3(b_1 + b_2 + h)$
             $a := a_1a_2/d_2^2,\ b := (s_1'a_1b_2 + s_2'a_2b_1 + s_3(b_1b_2 + f))/d_2 \pmod{a}$
Step A3  While $\deg(a_3) > g$ do the following:
             $a_3 := (f - b - b^2)/a,\ b_3 := -h - b \pmod{a_3}, a := a_3, b := b_3$
Step A4  Return $D_3 = \mathrm{div}(a_3,b_3)$
```

If a_1 and a_2 have no common factor, **Step A2** to be simpler case. Note that the case $\gcd(a_1, a_2) = 1$ is extremely likely if the definition field is large and a_1 and a_2 are the coordinates of two randomly chosen elements of the jacobian.

When $a_1 = a_2$ and $b_1 = b_2$, i.e., doubling an element of $\mathbf{J}(C; \mathbf{F}_{2^n})$, we can take $s_2 = 0$. Moreover, in the case of char $\mathbf{F}$ =2 and $h(u) = 1$, $d_1 = 1$, $s_1 = s_2 = 0$, $S_3 = 1$, and $a = {a_1}^2$, $b = {b_1}^2 + f \pmod{a}$. Therefore, In the case of $\mathbf{J}(C; \mathbf{F}_{2^n})$ and $C : v^2 + v = f(u)$, the doubling can be done in the algorithm as follows.

```
Doubling
Input:   a divisor $D_1 = \mathrm{div}(a_1,b_1) \in \mathbf{J}$
Output:  $D_2 = \mathrm{div}(a_2,b_2) = D_1 + D_1$
Step D1  $a := a_1^2,\ b := b_1^2 + f \pmod{a}$
Step D2  While $\deg(a_2) > g$ do the following:
             $a_2 := (f - b - b^2)/a,\ b_2 := -h - b \pmod{a_2}, a := a_2, b := b_2$
Step D3  Return $D_2 = \mathrm{div}(a_2,b_2)$
```

Addition and doubling take $O(g^3)$ field multiplications. The details of the estimation on the computational cost can be found in [SSI98].

7.2 Field Operations

All operations in addition and doubling of $D \in \mathbf{J}(C; \mathbf{F}_{2^n})$ are done by operations in a finite field, because our divisors D (pair of two polynomials) have coefficients in their definition field. We will use a polynomial basis in our implementations.

7.3 Representation of Field Elements in Memory

Elements in $\mathbf{F}_{2^n}$ can be represented as n-bit words in computer memory. If CPU has m-bit size of resisters, $\mathbf{F}_{2^n}$ such as $n \leq m$ are regarded as simply ordinary ”*unsigned integer*”. However, unfortunately, if $n > m$, we need to use ”*multi-precision*” operations for computing. In general, we need such multi-precision operations for RSA and elliptic curve cryptosystems. On the other hand, the order of an abelian variety $\mathbf{A}(\mathbf{F}_q)$ of genus g lies in the range: $(q^{\frac{1}{2}} - 1)^{2g} \leq \sharp\mathbf{A}(\mathbf{F}_q) \leq (q^{\frac{1}{2}} + 1)^{2g}$ [St93]. Therefore, if we choose curves carefully, hyperelliptic cryptosystems, which have larger genus g curves compared to elliptic curve, can be implemented without multi-precision library, because an element of the definition field can be stored in computer registers. Such hyperelliptic cryptosystems may have practical performance even though the algorithm for addition of D, shown in the last sub-section, is much more expensive than the algorithm for addition of points on elliptic curves.

7.4 Generating Random Divisors

From cryptosystems point of view, we need to have a method of generating a "*random*" divisor $D \in \mathbf{J}(C; \mathbf{F}_{q^n})$. In [Ko89], Koblitz has given such a method in the following way. In our implementation, we have generated divisors D in the method.

We may regard C as defined over $\mathbf{F}_{q^n}$. Let C have the equation $v^2 + h(u)v = f(u)$. Choose the coordinate $u = x \in \mathbf{F}_q$ at random and attempt to solve $v^2 + h(x)v = f(x)$. In the case of q is even, $h(x) \neq 0$ and the change of variables $z = v/h(x)$ leads to the equation $z^2 + z = a$, where $a = f(x)/h(x)^2$. It is easy to see that this equation has a solution $z \in \mathbf{F}_q$ if $\mathrm{Tr}_{\mathbf{F}_q/\mathbf{F}_2} a = 0$ and does not have a solution if this trace is 1. In the latter case, we must choose another $u = x \in \mathbf{F}_q$ and start again. In the former case, we can find z as follows: If $q = 2^n$ is an odd power of 2, simply set $\sum_{j=0}^{(n-1)/2} a^{2^{2j}}$.

7.5 Timings

We have implemented group operations in jacobians over $\mathbf{F}_{2^n}$ and timed an exponentiation, an addition and a doubling of randomly generated divisors using the algorithms shown in the previous subsection. An exponentiation was done with a simple repeated-doubling method.

The platforms used were Alpha 21164A (467MHz) and Pentium-II (300MHz). Alpha has 64-bit registers and Pentium-II has 32-bit registers. Programs were written in C-language. When extension degree n of $\mathbf{F}_{2^n}$ has a larger size than the register size of the CPU, we used GNU-MP library (gmp-2.0.2) for multi-precision operations.

Table 6 shows the processing time of an exponentiation, an addition and a doubling of a randomly given divisor implemented on Alpha 21164A (467MHz) and Pentium-II (300MHz). The order of each jacobians $\sharp\mathbf{J}(C; \mathbf{F}_{2^n})$ have the largest prime factor which has a larger size than 160-bit, namely, they have the same or higher level of security as RSA-1024 and EC-160.

All jacobians of Table 6 are defined over finite fields $\mathbf{F}_{2^n}$ with $n \leq 64$. Therefore, we can implement with no multi-precision library over Alpha 21164A (467MHz). $\mathbf{J}(C; \mathbf{F}_{2^{59}})$ of genus 3 curve $C : v^2 + v = u^7$ achieved 83.3 msec. in an exponentiation. Moreover, in the case of Pentium, we should focus on the case

g	$\mathbf{J}(v^2+v=f(u);\mathbf{F}_{2^n})$		Addition(msec.)		Doubling(msec.)		Exp.(msec.)	
	$\mathbf{F}_{2^n}$	$f(u)$	Alpha	Pentium	Alpha	Pentium	Alpha	Pentium
3	$\mathbf{F}_{2^{59}}$	u^7	0.54	67.6	0.26	34.1	83.3	$1.17 \cdot 10^4$
4	$\mathbf{F}_{2^{41}}$	$u^9+u^7+u^3+1$	0.55	67.2	0.26	33.3	96.6	$1.09 \cdot 10^4$
5	$\mathbf{F}_{2^{41}}$	$u^{11}+u^5+u+1$	0.88	109	0.48	58.7	183	$2.36 \cdot 10^4$
6	$\mathbf{F}_{2^{29}}$	$u^{13}+u^{11}+u^7+u^3+1$	0.83	2.68	0.44	1.45	159	476

Table 6. Timings of jacobians which have the same level of security as RSA-1024 on Alpha 21164A (467MHz) and Pentium-II (300MHz)

g	$\mathbf{J}$	C	size of P_{max}	Addition (msec.)	Doubling (msec.)	Exp. (msec.)
3	$\mathbf{J}(C;\mathbf{F}_{2^{89}})$	$v^2+v=u^7$	246-bit	85.3	42.8	$2.57 \cdot 10^4$
3	$\mathbf{J}(C;\mathbf{F}_{2^{113}})$	$v^2+v=u^7$	310-bit	118	58.9	$3.79 \cdot 10^4$
11	$\mathbf{J}(C;\mathbf{F}_{2^{47}})$	$v^2+v=u^{23}$	310-bit	5.04	3.13	$1.74 \cdot 10^3$

Table 7. Timings of jacobians of $C : v^2 + v = u^{2g+1}$ which have the same level of security as RSA-2048 or RSA-5000 over Alpha 21164A (467MHz)

$\mathbf{J}(C;\mathbf{F}_{2^{29}})$ of genus 6 curve $C : v^2 + v = u^{13} + u^{11} + u^7 + u^3 + 1$. An exponentiation took 476 msec. on Pentium-II (300MHz). This jacobian achieves good performance and faster than other jacobians of smaller genus curves, because of the field size.

Moreover, we have implemented genus 3 and 11 curves, which have the same level of security as RSA-2048 and RSA-5000. Table 7 shows the processing time of $C : v^2 + v = u^{2g+1}$ implemented over Alpha 21164A (467MHz). Even if the genus is 11, which has the same level of security as RSA-5000, exponentiation took 1.79 sec because of its small size of the definition field.

In the case of elliptic curve cryptosystems, many techniques for an efficient implementation has been developed, and timings were reported. For example, in [WBV96], an elliptic curve (over $\mathbf{F}_{2^{177}}$) exponentiation with 177-bit exponent achieved 72 msec. on Pentium 133 MHz. In [MOC97], an elliptic curve (over $\mathbf{F}_p, p = 2^{169} - 1825$) exponentiation with 169-bit exponent (of a random point) achieved 32.54 msec. on Sparc 110 MHz. On the other hand, in the case of hyperelliptic cryptosystems, no such a report has been published. Our hyperelliptic curves exponentiation, which have smaller definition fields, are a few times slower than the elliptic curves cases. However, our implementation suggests that hyperelliptic curve cryptosystems may have practical performance.

Acknowledgments

The second author would like to thank Iwan Duursma, who informed him of a method for computing the order of a jacobian without evaluating the exact roots of zeta. The second author also wishes to thank Zvi Galil and Moti Yung for their generous hospitality while his visiting Columbia Univ. Computer Science Dept.

References

ADH94. L.M. Adleman, J. DeMarrais and M. Huang, "A Subexponential Algorithm for Discrete Logarithm over the Rational Subgroup of the Jacobians of Large Genus Hyperelliptic Curves over Finite Fields", *Proc. of ANTS1, LNCS*, vol. 877, Springer-Verlag, (1994), 28–40

BK98. J. Buhler and N. Koblitz, Joe Buhler and Neal Koblitz, "Lattice basis reduction, Jacobi sums and hyperelliptic cryptosystems,", Bull. Austral. Math. Soc. (1998)

BS91. T. Beth and F. Scaefer, "Non supersingular elliptic curves for public key cryptosystems", *Advances in Cryptology - EUROCRYPT '91, Lecture Notes in Computer Science*, **547**, pp.316–327 (1991).

Ca87. D.G. Cantor, "Computing in the Jacobian of a Hyperelliptic Curve", *Math. Comp*, **48**, No.177 (1987), 95–101

CMT97. J. Chao, N. Matsuda, and S. Tsujii, "Efficient construction of secure hyperelliptic discrete logarithms", *Information and Communications Security*, Springer-Verlag, LNCS 1334 (1997), 292–301.

CTT94. J. Chao, K. Tanaka, and S. Tsujii, "Design of elliptic curves with controllable lower boundary of extension degree for reduction attacks", *Advances in Cryptology - Crypto'94*, Springer-Verlag, (1994), 50–55.

Fr97. G. Frey, "Aspects of DL-systems based on hyperelliptic curves", *Keynote Lecture in Waterloo-Workshop on Elliptic Curve Discrete Logarithm Problem*, 4th of Nov. (1997).

FR94. G. Frey and H.G. Rück, "A Remark Concerning m-Divisibility and the Discrete Logarithm in the Divisor Class Group of Curves", *Math. Comp*, **62**, No.206 (1994), 865–874

GLV98. Robert Gallant, Robert Lambert and Scott Vanstone, "Improving the parallelized Pollard lambda search on binary anomalous", A draft is available from `http://grouper.ieee.org/groups/1363/contrib.html`, (April,1998)

Ko88. N. Koblitz, "A Family of Jacobians Suitable for Discrete Log Cryptosystems", *Advances in Cryptology - Crypto'88*, Springer-Verlag, (1990), 94–99

Ko89. N. Koblitz, "Hyperelliptic Cryptosystems", *J.Cryptology*, **1** (1989), 139–150

Ko98. N. Koblitz, "Algebraic Aspects of Cryptography", Springer-Verlag, (1998)

MOC97. A. Miyaji, T. Ono and H. Cohen, "Efficient Elliptic curve Exponentiation", *Information and Communications Security*, Springer-Verlag, (1997), 282–290.

MOV93. A.J. Menezes, T. Okamoto and S.A. Vanstone, "Reducing elliptic curve logarithm to logarithm in a finite field", *IEEE Trans. on IT*, **39**, (1993), 1639–1646

PH78. S.C. Pohlig and M.E. Hellman, "An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance", *IEEE Trans. on IT*, **24**, (1978), 106–110

Ru97. H.G. Rück, "On the discrete logarithms in the divisor class group of curves", To appear in *Math. Comp.* (1997)

SA97. T. Satoh and K. Araki, "Fermat Quotients and the Polynomial Time Discrete Log Algorithm for Anomalous Elliptic Curves", *preprint*, (1997)

Sem98. I.A. Semaev, "Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p", *Math. Comp.*, Vol.76 (1998), 353–356.

Sm97. N.P. Smart, "The Discrete Logarithm Problem on Elliptic Curves of Trace One", *preprint*, (1997) SSI98. Y. Sakai, K. Sakurai and H. Ishizuka, "Secure hyperelliptic cryptosystems and their performance", *Pre-Proc. PKC'98* (1998)

St93. H. Stichtenoth, "Algebraic Function Fields and Codes", Springer-Verlag, (1993)

WBV96. E.D. Win, A. Bosselaers, and S. Vandenberghe, "A Fase Software Implementation for Arithmetic Operations in GF(2^n)" *Advances in Cryptology - Asiacrypt'96*, Springer-Verlag, (1996), 65–76.

WZ98. Michael Wiener and Robert Zuccherato, "Faster Attacks on Elliptic Curve Cryptosystems," A draft is available from `http://grouper.ieee.org/groups/1363/contrib.html`, (April,1998)

A Jacobians which have the Same Level of Security as RSA-1024

In this Appendix, we show jacobians such that the largest prime factor (P_{max}) of $\sharp\mathbf{J}(C;\mathbf{F}_{q^n})$ has the size of approximately 2^{160}.

A.1 Characteristic 2

Genus 2 curves

$\mathbf{J}(C;\mathbf{F}_{2^{89}}), C : v^2 + (u^2+u+1)v = u^5 + u + 1/\mathbf{F}_2$ (P_{max}:178-bit)

$\sharp\mathbf{J} = 2 \cdot 191561942608242456073498418252108663615312031512914969$

Genus 3 curves

$\mathbf{J}(C;\mathbf{F}_{2^{59}}), C : v^2 + v = u^7/\mathbf{F}_2$ (P_{max}:165-bit)

$\sharp\mathbf{J} = 7 \cdot 827 \cdot 33090679324276404784037550343359349791850702512053$

Genus 4 curves

$\mathbf{J}(C;\mathbf{F}_{2^{41}}), C : v^2 + v = u^9 + u^7 + u^3 + 1/\mathbf{F}_2$ (P_{max}:161-bit)

$\sharp\mathbf{J} = 11 \cdot 2125818615244041340661452662120917241919480417187$

Genus 5 curves

$\mathbf{J}(C;\mathbf{F}_{2^{41}}), C : v^2 + v = u^{11} + u^5 + u + 1/\mathbf{F}_2$ (P_{max}:201-bit)

$\sharp\mathbf{J} = 29 \cdot 1773173014354747890253199550169173842018096398692873319662133$

Genus 6 curves

$\mathbf{J}(C;\mathbf{F}_{2^{29}}), C : v^2 + v = u^{13} + u^{11} + u^7 + u^3 + 1/\mathbf{F}_2$ (P_{max}:170-bit)

$\sharp\mathbf{J} = 23 \cdot 1040988300089925365337867649065425169641062000079783$

$\mathbf{J}(C;\mathbf{F}_{2^{29}}), C : v^2 + v = u^{13} + u^{11} + u^9 + u^5 + 1/\mathbf{F}_2$ (P_{max}:171-bit)

$\sharp\mathbf{J} = 13 \cdot 1841646667025959098054051155819603805847557201575621$

A.2 Characteristic 3

Genus 2 curves

$\mathbf{J}(C;\mathbf{F}_{3^{59}}), C : v^2 = u^5 + u^4 + u^3 + u + 1/\mathbf{F}_3$ (P_{max}:185-bit)

$\sharp\mathbf{J} = 5 \cdot 39933562220320460133120368418577581396339849557868704977$

Genus 3 curves

$\mathbf{J}(C;\mathbf{F}_{3^{37}}), C : v^2 = u^7 + u^5 + u^3 + u^2 + 1/\mathbf{F}_3$ (P_{max}:171-bit)

$\sharp\mathbf{J} = 5 \cdot 7 \cdot 2608502325966498106517804088886290895899401162747777$

$\mathbf{J}(C;\mathbf{F}_{3^{37}}), C : v^2 = u^7 + u^6 + u^5 + u^4 + 1/\mathbf{F}_3$ (P_{max}:164-bit)

$\sharp\mathbf{J} = 47 \cdot 149 \cdot 13036924465204430321626282159677955928081271322337$

Genus 4 curves

$\mathbf{J}(C;\mathbf{F}_{3^{29}}), C : v^2 = u^9 + u^8 + u^7 + u^4 + u^3 + u^2 + u + 1/\mathbf{F}_3$ (P_{max}:177-bit)

$\sharp\mathbf{J} = 137 \cdot 161936596667550201850764509341446010074223174018351807$

$\mathbf{J}(C;\mathbf{F}_{3^{29}}), C : v^2 = u^9 + u^6 + u^5 + u^3 + 1/\mathbf{F}_3$ $(P_{max}$:178-bit$)$

$\sharp\mathbf{J} = 2 \cdot 43 \cdot 257968616884115037815521227015137018694634859077456601$

$\mathbf{J}(C;\mathbf{F}_{3^{29}}), C : v^2 = u^9 + u^7 + u^5 + u^4 + u^3 + u + 1/\mathbf{F}_3$ $(P_{max}$:178-bit$)$

$\sharp\mathbf{J} = 2 \cdot 53 \cdot 209295417307275986159417399889573453667424414714367781$

$\mathbf{J}(C;\mathbf{F}_{3^{29}}), C : v^2 = u^9 + u^7 + u^6 + u^2 + 1/\mathbf{F}_3$ $(P_{max}$:178-bit$)$

$\sharp\mathbf{J} = 3 \cdot 37 \cdot 199867665359855144576119236835076831435089172471787133$

$\mathbf{J}(C;\mathbf{F}_{3^{29}}), C : v^2 = u^9 + u^7 + u^6 + u^5 + u^4 + u^3 + u + 1/\mathbf{F}_3$ $(P_{max}$:178-bit$)$

$\sharp\mathbf{J} = 5 \cdot 19 \cdot 233529568966516115208148703607774647750483532026125357$

A.3 Characteristic 5

Genus 2 curves

$\mathbf{J}(C;\mathbf{F}_{5^{43}}), C : v^2 = u^5 + u^2 + 1/\mathbf{F}_5$ $(P_{max}$:196-bit$)$

$\sharp\mathbf{J} = 2 \cdot 2 \cdot 5 \cdot 64623485355705134605734078473194763210739812239980205784653$

Genus 3 curves

$\mathbf{J}(C;\mathbf{F}_{5^{23}}), C : v^2 = u^7 + u^6 + u^2 + 1/\mathbf{F}_5$ $(P_{max}$:154-bit$)$

$\sharp\mathbf{J} = 3 \cdot 43 \cdot 13132293573869607525341363618339646743815332017$

Genus 4 curves

$\mathbf{J}(C;\mathbf{F}_{5^{19}}), C : v^2 = u^9 + u^6 + u^4 + u^3 + 1/\mathbf{F}_5$ $(P_{max}$:166-bit$)$

$\sharp\mathbf{J} = 2 \cdot 967 \cdot 68432754693421761179795901150463384835984065125361$

$\mathbf{J}(C;\mathbf{F}_{5^{19}}), C : v^2 = u^9 + u^7 + u^6 + u^5 + u^3 + u + 1/\mathbf{F}_5$ $(P_{max}$:168-bit$)$

$\sharp\mathbf{J} = 3 \cdot 151 \cdot 292161172338621074756327634541902615881173270592929$

$\mathbf{J}(C;\mathbf{F}_{5^{19}}), C : v^2 = u^9 + u^8 + u^7 + u^4 + u + 1/\mathbf{F}_5$ $(P_{max}$:167-bit$)$

$\sharp\mathbf{J} = 17 \cdot 73 \cdot 106647119155998044412946215375749800145892212819953$

A.4 Characteristic 7

Genus 2 curves

$\mathbf{J}(C;\mathbf{F}_{7^{29}}), C : v^2 = u^5 + u^4 + u^2 + 1/\mathbf{F}_7$ $(P_{max}$:157-bit$)$

$\sharp\mathbf{J} = 79 \cdot 13123788704224285743106665024398819031341821830 1$

Genus 3 curves

$\mathbf{J}(C;\mathbf{F}_{7^{19}}), C : v^2 = u^7 + u^6 + u^5 + u^3 + u + 1/\mathbf{F}_7$ $(P_{max}$:152-bit$)$

$\sharp\mathbf{J} = 2^3 \cdot 41 \cdot 45155893888076543451041824833966116595614725 03$

Genus 4 curves

$\mathbf{J}(C;\mathbf{F}_{7^{17}}), C : v^2 = u^9 + u^8 + u^6 + u^5 + u^3 + u + 1/\mathbf{F}_7$ $(P_{max}$:181-bit$)$

$\sharp\mathbf{J} = 2^4 \cdot 97 \cdot 188701387296773136203522548345057408767223350900238191 1$

B Curves of $v^2 + v = u^{2g+1}$ over $\mathbf{F}_2$

In this Appendix, we show jacobians of $C : v^2 + v = u^{2g+1}$ in the case of $g = 3, 11$.

Genus 3 curves

$\mathbf{J}(C;\mathbf{F}_{2^{59}}), C : v^2 + v = u^7/\mathbf{F}_2$ $(P_{max}$:165-bit$)$

$\sharp\mathbf{J} = 7 \cdot 827 \cdot 3309067932427640478403755034335934979185070251205 3$

$\mathbf{J}(C;\mathbf{F}_{2^{89}}), C : v^2 + v = u^7/\mathbf{F}_2$ $(P_{max}$:264-bit$)$

$\sharp\mathbf{J} = 7{\cdot}179{\cdot}2671{\cdot}70857183122325533127823325792054243244435303883153993362539 1$
34263544967267

$\mathbf{J}(C;\mathbf{F}_{2^{113}}), C : v^2 + v = u^7/\mathbf{F}_2$ $(P_{max}$:310-bit$)$

$\sharp\mathbf{J} = 7{\cdot}1583{\cdot}75937{\cdot}133087154459125850390435059436398888423626351517540604207 6$
$3267396674295645712955192381380503 93$

Genus 11 curves

$\mathbf{J}(C;\mathbf{F}_{2^{47}}), C : v^2 + v = u^{23}/\mathbf{F}_2$ $(P_{max}$:310-bit$)$

$\sharp\mathbf{J} = 3{\cdot}23{\cdot}29{\cdot}34687{\cdot}254741{\cdot}381077{\cdot}836413{\cdot}4370719{\cdot}122803256446193{\cdot}101578405621916$
$029{\cdot}1396360023741601228722804905934361404439480177105909460096120108013$
$8678351892941240936676 87457$