Compact Representation of Domain Parameters of Hyperelliptic Curve Cryptosystems

Fangguo Zhang[1] Shengli Liu[2] and Kwangjo Kim[1]

[1] International Research center for Information Security (IRIS)
Information and Communications University(ICU),
58-4 Hwaam-dong Yusong-ku, Taejeon, 305-732 KOREA
{zhfg, kkj}@icu.ac.kr
[2] Dept. of Computer Science, Shanghai Jiaotong University,
Shanghai 200030, P.R.China
liu-sl@cs.sjtu.edu.cn

Abstract. To achieve the same level of security, hyperelliptic curve cryptosystems (HCC) use a smaller field than elliptic curve cryptosystems (ECC). HCC has a more potential application to the product that has limited memory and computing power, for instance Smart cards. We discussed how to represent the domain parameters of HCC in a compact way. The domain parameters include the field over which the curve is defined, the curve itself, the order of the Jocobian and the base point. In our method, the representation of HCC with genus g=4 over $F_{2^{41}}$ (It can provide the same level of security with 164 bits ECC) only uses 339 bits.

Key words Hyperelliptic curve cryptosystems(HCC), Jacobian, Domain parameters

1 Introduction

Elliptic Curve Cryptosystems (ECC) are receiving more attention. Elliptic curves have shown to be good resources to obtain Abelian groups. The discrete logarithm problem based on the Abelian group can be intractable, and no subexponential time algorithm is known to solve the problem, if the curve is properly chosen. Hyperelliptic Curve Cryptosystems (HCC) was proposed by N. Koblitz in [11] as a generalization of ECC, since an elliptic curve be a hyperelliptic curve of genus $g = 1$. The Jacobians of a hyperelliptic curve can serve as a source of finite Abelian groups, over which the discrete logarithm problems are defined. Every scheme based on ECC, such as DSA and ElGamal, has its variant based on HCC. Suppose that F_q is the field on which the Jacobian of a hyperelliptic curve of genus g is defined. Then, there are about q^g points on the Jacobian. The advantage of HCC over ECC is that a smaller ground field F_q can be used to achieve the same order of magnitude of the Abelian group. That means that HCC can be implemented with a smaller word length in computers than ECC. Therefore, HCC may avoid multiprecision integer arithmetic when implemented.

Let $\overline{F_q}$ be the algebraic closure of the field F_q. A hyperelliptic curve C of genus g over F_q with $g \geq 1$ is given by the following equation:

$$C : y^2 + h(x)y = f(x) \tag{1}$$

where $f(x)$ is a monic polynomial of degree $2g+1$, $h(x)$ is a polynomial of degree at most g, and there is no solutions $(x, y) \in \overline{F_q} \times \overline{F_q}$ simultaneously satisfying the equation $y^2 + h(x)y = f(x)$ and the partial derivative equations $2y + h(x) = 0$ and $h'(x)y - f'(x) = 0$.

We denote the Jacobian group over the hyperelliptic curve C of genus g over F_q by $J(C; F_q)$. The order of the Jacobian group is denoted by $\#J(C; F_q)$.

Like with ECC, not every hyperelliptic curve can be used for HCC. To build a secure HCC, the curves have to be chosen to satisfy the following properties:

1. A large prime number n of at least 160 bits can divide $\#J(C; F_q)$. The reason is the following. The complexity of Pohlig-Hellman algorithm for Hypercurve Discrete Logarithm Problem (HCDLP) is proportional to the square root of the largest prime in the factors of $\#J(C; F_q)$.
2. The large prime number n should not divide $q^k - 1$ for all small k's for which the discrete logarithm problem in F_{q^k} is feasible. This is to avoid the reduction attack proposed by Frey and R*ü*ck in [4]. The reduction attack reduces the HCDLP over the $J(C; F_q)$ to the logarithm problem in an extended field F_{q^k}. It is efficient especially for supersingular curves, see [5].
3. When q is prime, there should be no subgroup of order q in $J(C; F_q)$. Because there is an attack on anomalous curves investigated by Semaev [19], Satoh and Araki [18],Smart [21] for elliptic and generalized by R*ü*ck for hyperelliptic curves in [16].
4. $2g + 1 \leq \log q$. When $2g + 1 > \log q$, Adleman, DeMarrais and Huang gave a sub-exponential time algorithm to solve HCDLP in [1]. Further study by Gaudry in [7] suggested that $g \leq 4$.

Therefore, We will consider hyperelliptic curves $C : y^2 + h(x)y = f(x)$ of genus $g \leq 4$ over F_q, and $2^{160} \leq q^g \leq 2^{300}$.

When q is prime, according to Lemma 2 in [13], Equation (1) can be transformed to the form

$$y^2 = f(x)$$

by replacing y by $y - h(x)/2$. Here $f(x)$ has a degree $2g + 1$.

When $q = 2^m$, the following propositions hold.

Proposition 1. *[5] Let C be a genus 2 curve over F_{2^m} of the form $y^2+by = f(x)$ where $f(x)$ is monic of degree 5 and $b \in F^*_{2^m}$. Then C is supersingular.*

Proposition 2. *[20] For every integer $h \geq 2$, there are no hyperelliptic supersingular curves over $\overline{F_2}$ of genus $2^n - 1$.*

From the above two propositions, we know that HCC can employ hyperelliptic curves over F_{2^m} of genus 3 or 4 of form

$$y^2 + y = f(x).$$

When $g = 2$, we avoid supersingular curves, and use curves of form

$$y^2 + xy = f(x)$$

instead.

When a public cryptosystem is employed in practice, the corresponding parameters should be distributed and stored. It is attractive if the parameters can be represented in a compact way, especially for the case when the available memory is limited (for instance, smart cards). In [22], Smart studied how the ECC parameters are represented with a very small number of bits. In this paper, we will investigate how to compress the parameters of a HCC with a given genus g. To define a HCC, the following parameters are necessary:

1. The finite field F_q;
2. A hyperelliptic curve defined over F_q;
3. The order of the Jacobian over the hyperelliptic curve;
4. The base point of the Jacobian.

2 Compact Representation of the Domain Parameters of a HCC

2.1 The finite field F_q

The discussion is restricted to two kinds of fields, namely large prime fields (with $q = 2^m - 1$ as a Mersenne number) and fields of characteristic 2, i.e. $q = 2^m$.

Large prime fields:

There is a good reason to choose q as a Mersenne number. No integer division is required for modular reduction in modular multiplication modulo a Mersenne number $q = 2^m - 1$, see [23] [9]. Suppose $a, b, t, u \in F_q$, and $c = ab = 2^m t + u$, we have $c = (t + u)\ mod\ q$.

There is no Mersenne number between 2^{160} and 2^{300}. Therefore, ECC cannot take advantage of the shortcut for modular multiplication modulo a Mersenne number, when $2^{160} \leq q \leq 2^{300}$. However, things are different for HCC since $2^{160} \leq q^g \leq 2^{300}$ is required. When $g = 2$, Mersenne numbers $q = 2^m - 1$ with $m = 89, 107$ *or* 127 can be used. When $g = 3$, Mersenne numbers with $m = 61$ *or* 89 can be applied. It is easy to see that 7 bits are enough to represent these Mersenne numbers (hence the finite field F_q).

Fields of characteristic 2:

We restrict F_{2^m} to those fields with primitive trinomial bases as their generators. With primitive trinomial bases, modular reduction is efficient. In the mean time, only three terms are required to represent the field, namely, $x^m + x^c + 1$.

We can choose $80 < m < 128$ for $g = 2$, $53 \leq m < 90$ for $g = 3$ and $41 \leq m < 75$ for $g = 4$. That trinomial $x^m + x^c + 1$ is primitive implies that $x^m + x^{m-c} + 1$ is also primitive. For instance, both $x^{97} + x^6 + 1$ and $x^{97} + x^{91} + 1$ are primitive. Hence, we can always choose a primitive trinomial $x^m + x^c + 1$ with $c \leq m/2$ to represent the fields. To thwart the *Weil Descent* attack [6], m is usually chosen as a prime number. Therefore, 12 bits, 6 bits for m and the other 6 bits for c, are enough to represent the field.

Between 40 and 128, there are 11 prime numbers from which m can be chosen, namely, 41, 47, 71, 73, 79, 89, 97, 103, 113, 119, and 127.

2.2 The hyperelliptic curve defined over F_q

As suggested in Section 1, the following hyperelliptic curves (HC) will be considered.

g	*HC over F_q, where q is prime, $f_i \in F_q$*
2	$y^2 = x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$
3	$y^2 = x^7 + f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$
4	$y^2 = x^9 + f_8x^8 + f_7x^7 + f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$

Table 1. Hyperelliptic curves over F_q of genus g when q is prime and $g = 2, 3, 4$

g	*HC over F_q, where $q = 2^m$, $f_i \in F_q$*
2	$y^2 + xy = x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$
3	$y^2 + y = x^7 + f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$
4	$y^2 + y = x^9 + f_8x^8 + f_7x^7 + f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0$

Table 2. Hyperelliptic curves over F_q of genus g when $q = 2^m$ and $g = 2, 3, 4$

Now we are ready to show how to represent the curves in fewer bits.

To represent the hyperelliptic curves over F_q, where q is prime, we have the following theorems:

Theorem 1. *When q is prime, hyperelliptic curves of genus $g = 2$ over F_q can be transformed to the form*

$$y^2 = x^5 + a_3x^3 + a_2x^2 + a_1x + a_0. \tag{2}$$

A hyperelliptic curve of genus 3 over F_q can be transformed to the form

$$y^2 = x^7 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0. \tag{3}$$

A hyperelliptic curve of genus 4 over F_q can be transformed to the form

$$y^2 = x^9 + a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0. \tag{4}$$

where $a_i \in F_q$.

Proof. When the characteristic of the field F_q is not 2, a hyperelliptic curve of genus 2 over F_q is given by the following equation

$$y^2 = x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0, \tag{5}$$

where $f_i \in F_q$.

Changing variables x by $u^2x - f_4/5$ and y by u^5y in Equation (5) we get

$$\begin{aligned} u^{10}y^2 = u^{10}x^5 &+ \left(f_3u^6 - \frac{2}{5}u^6{f_4}^2\right)x^3 + \left(f_2u^4 + \frac{4}{25}u^4{f_4}^3 - \frac{3}{5}f_3f_4u^4\right)x^2 \\ &+ \left(-\frac{2}{5}f_2u^2f_4 - \frac{3}{125}u^2{f_4}^4 + \frac{3}{25}f_3u^2{f_4}^2 + f_1u^2\right)x \\ &- \frac{1}{5}f_1f_4 + f_0 + \frac{1}{25}f_2f_4^2 + \frac{4}{3125}{f_4}^5 - \frac{1}{125}f_3f_4^3. \end{aligned}$$

Let

$$a_3 = \left(f_3u^6 - \frac{2}{5}u^6{f_4}^2\right)/u^{10},$$

$$a_2 = \left(f_2u^4 + \frac{4}{25}u^4{f_4}^3 - \frac{3}{5}f_3f_4u^4\right)/u^{10},$$

$$a_1 = \left(-\frac{2}{5}f_2u^2f_4 - \frac{3}{125}u^2{f_4}^4 + \frac{3}{25}f_3u^2{f_4}^2 + f_1u^2\right)/u^{10},$$

and

$$a_0 = \left(-\frac{1}{5}f_1f_4 + f_0 + \frac{1}{25}f_2f_4^2 + \frac{4}{3125}{f_4}^5 - \frac{1}{125}f_3f_4^3\right)/u^{10}.$$

Then Equation (2) follows.

A hyperelliptic curve of genus 3 over F_q (recall that q is prime) is given by

$$y^2 = x^7 + f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0.$$

With the change of variables $x \to x - f_6/7$ and $y \to y$, we get Equation (3).

With the change of variables $x \to x - f_8/9$ and $y \to y$, Equation (4) is obtained from

$$y^2 = x^9 + f_8x^8 + f_7x^7 + f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0.$$

In fact, when the characteristic of the field F_q is not 2 and $2g{+}1$, hyperelliptic curves of genus g over F_q have the form of

$$y^2 = x^{2g+1} + a_{2g-1}x^{2g-1} + a_{2g-2}x^{2g-2} + \ldots + a_1x + a_0,$$

where $a_i \in F_q$ for $i = 1, 2, \ldots, 2g-1, 2g+1$.

The results is given in Table 3 as a comparison with Table 1.

For a field of characteristic 2, we have two facts as follows:

Fact 1. The map $\sigma : x \to x^2$ is an isomorphism, and its inversion is given by $\sigma^{-1} : y \to y^{1/2}$.

Fact 2. For $a \in F_{2^m}$, the equation $x^2 + x = a$ has a solution in F_{2^m} if and only if $Tr(a) = 0$. Here $Tr(a) = \sum_{i=1}^{m} a^{2^{i-1}}$ is the trace function of F_{2^m}.

g	*HC over F_q, where q is prime, $a_i \in F_q$*
2	$y^2 = x^5 + a_3x^3 + a_2x^2 + a_1x + a_0$
3	$y^2 = x^7 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$
4	$y^2 = x^9 + a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$

Table 3. Hyperelliptic curves over F_q of genus g when q is prime and $g = 2, 3, 4$

Theorem 2. *When a hyperelliptic curve of genus $g = 2$ over F_{2^m} has a form*

$$y^2 + xy = x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0, \tag{6}$$

it can be transformed to a form of

$$y^2 + xy = x^5 + a_3x^3 + \epsilon x^2 + a_1x;\ \textit{here } \epsilon \in F_2, a_1 \neq 0 \tag{7}$$

When a hyperelliptic curve of genus $g = 3$ over F_{2^m} has a form

$$y^2 + y = x^7 + f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0, \tag{8}$$

it can be transformed to a form of

$$y^2 + y = x^7 + a_5x^5 + a_3x^3 + a_2x^2 + \epsilon;\ \textit{here } \epsilon \in F_2 \tag{9}$$

When a hyperelliptic curve of genus $g = 4$ over F_{2^m} has a form

$$y^2 + y = x^9 + f_8x^8 + f_7x^7 + f_6x^6 + f_5x^5 + f_4x^4 + f_3x^3 + f_2x^2 + f_1x + f_0, \tag{10}$$

it can be transformed to a form of

$$y^2 + y = x^9 + a_7x^7 + a_5x^5 + a_3x^3 + a_2x^2 + \epsilon;\ \textit{here } \epsilon \in F_2 \tag{11}$$

where $a_i \in F_q$.

Proof. Changing variable y by $y + f_4^{1/2}x^2 + f_0^{1/2}$ in Equation (6) leads to

$$y^2 + xy = x^5 + a_3x^3 + a_2x^2 + a_1x, \tag{12}$$

when $Tr(a_2) = 0$, let β be a solution of the equation $x^2 + x = a_2$, with the change of variables $x \to x$ and $y \to y + \beta x$, then obtained equation

$$y^2 + xy = x^5 + a_3x^3 + a_1x; \tag{13}$$

when $Tr(a_2) = 1$, since m is odd, so $Tr(a_2 + 1) = 0$, let β be a solution of the equation $x^2 + x = a_2 + 1$, with the change of variables $x \to x$ and $y \to y + \beta x$, then the obtained equation is:

$$y^2 + xy = x^5 + a_3x^3 + x^2 + a_1x. \tag{14}$$

So Equation (7) can be obtained from $y^2+xy = x^5+f_4x^4+f_3x^3+f_2x^2+f_1x+f_0$; Changing variable y by $y + f_6^{1/2}x^3 + f_4^{1/2}x^2 + f_1x$ in Equation (8), we obtain

$$y^2 + y = x^7 + a_5x^5 + a_3x^3 + a_2x^2 + a_0, \tag{15}$$

and discuss the value of $Tr(a_0)$, changing variables $x \to x$ and $y \to y + \beta$, here β is a solution of the equation $x^2 + x = a_0$ or $x^2 + x = a_0 + 1$. Then this leads to Equation (9);

Changing variable y by $y + f_8^{1/2}x^4 + f_6^{1/2}x^3 + \left(f_8^{1/2} + f_4\right)^{1/2} x^2 + f_1x$ in Equation (10), we obtain

$$y^2 + y = x^9 + a_7x^7 + a_5x^5 + a_3x^3 + a_2x^2 + a_0, \tag{16}$$

and discuss the value of $Tr(a_0)$, changing variables $x \to x$ and $y \to y + \beta$, here β is a solution of the equation $x^2 + x = a_0$ or $x^2 + x = a_0 + 1$. Then this leads to Equation (11).

To compare with the representations of hyperelliptic curves in Table 2, we illustrate the results of Theorem 2 in Table 4.

g	$HC\ over\ F_q,\ where\ q = 2^m,\ a_i \in F_q$
2	$y^2 + xy = x^5 + a_3x^3 + \epsilon x^2 + a_1x,\ here\ \epsilon \in F_2, a_1 \neq 0$
3	$y^2 + y = x^7 + a_5x^5 + a_3x^3 + a_2x^2 + \epsilon,\ here\ \epsilon \in F_2$
4	$y^2 + y = x^9 + a_7x^7 + a_5x^5 + a_3x^3 + a_2x^2 + \epsilon,\ here\ \epsilon \in F_2$

Table 4. Hyperelliptic curves over F_q of genus g when $q = 2^m$ and $g = 2, 3, 4$

2.3 The order of the Jacobian over the hyperelliptic curve

To insure the security of hyperelliptic curve cryptosystems, the order of the Jacobian of the curve C, denoted by $\#J(C, F_q)$, should be chosen such that $\#J(C, F_q)$ contains a large prime divisor. Suppose that $\#J(C, F_q) = vn$, where n is a prime. Then the best known algorithm up to now for the HCDLP is of complexity $O(\sqrt{n})$. In this sequel, we limit $v \leq 2^6$.

According to Corollary 55 in [13], we have

$$(\sqrt{q} - 1)^{2g} \leq \#J(C, F_q) \leq (\sqrt{q} + 1)^{2g}.$$

Then we can use $\log{(\sqrt{q} + 1)^{2g}}$ bit to represent $\#J(C, F_q)$. Let $t = q^g + 1 - \#J(C, F_q)$. It is easy to see that

$$|t| \leq -\sum_{j=1}^{2g-1} \binom{2g}{j} q^{g-j/2}(-1)^j \leq 2gq^{g-1/2}.$$

Hence t has $1 + \log_2{(2gq^{g-1/2})}$ bits. It is easy to see that $\#J(C, F_q)$ is uniquely determined by t when q and g are known. That means that $1 + \log_2{(2gq^{g-1/2})}$ bits are enough to represent $\#J(C, F_q)$ (n as well). Consequently, the factorization of $\#J(C, F_q)$ can be represented by $7 + \log_2{(2gq^{g-1/2})}$ bits, where $1 + \log_2{(2gq^{g-1/2})}$ bits describing n and 6 bits describing v.

2.4 The base point of the Jacobian group

We consider the hyperelliptic curve $C : y^2 + h(x)y = f(x)$ of genus $g \leq 4$ over F_q. The order of the Jocobian of the curve is given by $\#J(C, F_q) = vn$, where n is prime and $v \leq 64$. The divisor of order n over the Jocobian is called the base point. This divisor generates a cyclic subgroup of order n. Any divisor D of $J(C, F_q)$ can be described by a pair of polynomials, one monomial of degree g and the other polynomial of degree $g-1$, namely $D = [a(x), b(x)] = [x^g + a_{g-1}x^{g-1} + \ldots + a_1 x + a_0, b_{g-1}x^{g-1} + \ldots + b_1 x + b_0]$, where $a_i, b_i \in F_q$. Therefore, every divisor D can be described as a $2g$-dimension vector $(a_{g-1}, \ldots, a_0, b_{g-1}, \ldots, b_0)$.

N. Kobliz gave algorithms to get random elements (divisors) of $J(C; F_q)$ in [11]. When an element from Kobliz's algorithms has an order that cannot divide v, then the element can be used as a base point.

The following two probabilistic algorithms show how to find a base point over F_q.

Algorithm 1. Algorithm of finding base point on $J(C, F_q)$ when q is a prime.

1. Repeat randomly choosing $\alpha \in F_q$ and calculating $f(\alpha)$ until $f(\alpha)$ is quadratic.
2. Determine the square root β of $f(\alpha)$.
3. Let $a(x) = x - \alpha$, $b(x) = \beta$. Then $[a(x), b(x)]$ is an element of the Jocobian $J(C, F_q)$.
4. Compute $D = v \cdot [a(x), b(x)]$. If $D = [1, 0]$ goto 1.
5. Output D.

Algorithm 2. Algorithm of finding base point on $J(C, F_q)$ where $q = 2^m$.

1. Randomly choose $\alpha \in F_q$ and calculate $h(\alpha)$ and $f(\alpha)$.
2. Let $c = f(\alpha)/h(\alpha)^2$. If the trace of c to F_2 is 1, i.e., $Tr(c) = 1$, goto 1. Otherwise, let $\beta = \sum_{i=0}^{(m-1)/2} c^{2^{2i}}$.
3. Let $a(x) = x - \alpha$, $b(x) = \beta$, then $[a(x), b(x)]$ is an element of $J(C; F_{2^m})$.
4. Compute $D = v \cdot [a(x), b(x)]$. If $D = [1, 0]$ goto 1.
5. Output D.

When α is randomly chosen from F_q, both the probability that $f(\alpha)$ in Step 1 of Algorithm 1 and the probability that $Tr(c) = 1$ in Step 2 of Algorithm 2 are given approximately 0.5.

Let ρ denote the probability that $D \neq [1, 0]$ in Step 4 when $f(\alpha)$ is a square in Algorithm 1 (or $Tr(c) = 1$ in Algorithm 2). Now we determine the value of ρ. Suppose that the number of divisors $[a(x), b(x)]$ in $J(C; F_q)$ such that $D = v \cdot [a(x), b(x)] = [1, 0]$ is given by N. Then each of the N divisors is an element of a subgroup of order w of $J(C; F_q)$, where w is a divisor of v, and denoted by $w|v$. Let $v = p_1^{e_1} p_2^{e_2} \ldots p_s^{e_s}$. The number of divisors of v is $(e_1+1)(e_2+1)\ldots(e_s+1)$. The number of subgroups of order w for all w such that $w|v$ is $(e_1+1)(e_2+1)\ldots(e_s+1)$ as well. The number of elements in such a subgroup is not more than v. Therefore, we have $N \leq v(e_1+1)(e_2+1)\ldots(e_s+1)$. Recall that we limit $v \leq 2^6$, so $N \leq 7^3 v$, and $\rho \geq (nv - 7^3 v)/nv = 1 - 7^3/n$.

The probability that t different α's are tried in the above algorithm without obtaining a base point D $(D \neq [1, 0])$ in Step 5 is $1 - (1 - 0.5\rho)^t$.

When the value of α is limited to $-2^7 < \alpha < 2^7$, the above algorithms fail with a probability about $1-(1-0.5\rho)^{255} \approx 1.73 \times 10^{-77}$ (there are 255 choices for α). The approximation comes from the fact that n is a prime of 160 bits. It means that there is a big chance to get a base point that can be represented by α, which only needs 8 bits.

The above analysis shows that we can use 8 bits to represent the base point.

The following two examples give a comparison between the general representation and compact representation of a HCC.

Example 1. Let q is a prime of 89 bits. A hyperelliptic curve of genus $g = 2$ over F_q is chosen for HCC. Then the general and compact representations of the HCC parameters are given in the following table:

Parameters	*general(bits)*	*compact(bits)*
Field	89	7
Hyperelliptic curve	$5 \cdot 89$	$4 \cdot 89$
Order of the Jacobian	$2 \cdot 89$	143
Base point	$4 \cdot 89$	8
Total	1068	514

Table 5. Comparison of general representation and compact representation for HCC over F_q for q prime and $g = 2$

Example 2. Let $q = 2^{41}$. A hyperelliptic curve of genus $g = 4$ over $F_{2^{41}}$ is chosen for HCC. Then the general and compact representations of the HCC parameters are given in the following table:

Parameters	*general(bits)*	*compact(bits)*
Field	≥ 12	$6+6=12$
Hyperelliptic curve	$9 \cdot 41$	$4 \cdot 41 + 1$
Order of the Jacobian	$4 \cdot 41$	154
Base point	$8 \cdot 41$	8
Total	≥ 853	339

Table 6. Comparison of general representation and compact representation for HCC over F_{2^m} and $g = 4$

From above two examples, the number of bits of our compact representation is less than half of general representation.

Note that the security level of the HCC in the first example corresponds to that of ECC over a field of 178 bits. The security level of the HCC in the second example corresponds to that of ECC over a field of 164 bits. A similar strength set of parameters for DSA would require 1024 bits for p, 160 bits for q and 1024 bits for the generator g, making 2208 bits in all.

3 Conclusion

How to represent the parameters of HCC in a very small number of bits and an efficient way are given. The domain parameters include the finite field on which the HCC is based, the representation of a hyperelliptic curve, the order of the Jacobian of the hyperelliptic curve, and the base point on the Jacobian. We shorten the representation of the prime field by choosing Mersenne numbers, and that of the field of characteristic 2 by choosing primitive trinomial base. How to eradicate an parameter in the equation of an hyperelliptic curve is also discussed. We also give the number of bits to represent the order of the Jacobian. As to the base point, we show it can be chosen with 8 bits for representation with high probability.

References

1. L. Adleman, J. De Marrais, M.-D Huang, *A Subexponential Algorithm for Discrete Logarithms over the Rational Subgroup of the Jacobians of Large Genus Hyperelliptic Curves over Finite Fields*, in ANTS-1, Algorithmic Number Theory , Editors L.M. Adlemand and M-D. Huang, Springer-Verlag, LNCS 877, pp. 28-40, 1994.
2. L. Adleman, M.-D Huang, *Counting rational points on curves and abelian varieties over finite fields*, In ANTS-2:, LNCS 1122, Springer-Verlag, pp. 1-16, 1996.
3. D.G. Cantor, *Computing in the Jacobian of a hyperelliptic curve*, Mathematics of Computation, Volume 48, pp.95-101, 1987.
4. G. Frey and H.Rück, *A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves*, Mathematics of Computation, 62, pp.865-874, 1994.
5. S.D. Galbraith, *Supersingular curves in cryptography*. Available at http://www.cs.bris.ac.uk/ stenve
6. S.D. Galbraith, *Weil descent of Jacobians*. Presented at WCC 2001. Available at http://www.cs.bris.ac.uk/ stenve.
7. P. Gaudry, *An algorithm for solving the discrete log problem on hyperelliptic curves*, In B.Preneel(ed.), Eurocrypt 2000, LNCS 1807, Springer-Verlag, pp.19-34, 2000.
8. P. Gaudry and R. Harley, *Counting Points on Hyperelliptic Curves over finite fields*. Available at http://www.cs.bris.ac.uk/Tools/Reports/Abstract/2000-gaudry.htm
9. D.E. Knuth, and E. Donald E., *Seminumerical Algorithms*, Addison-Wesley, 1981.
10. N. Koblitz, *Elliptic Curve Cryptosystems*, Mathematics of Computation,48, pp.203-209, 1987.
11. N. Koblitz, *Hyperelliptic cryptography*, J.of Crypto., No.1, pp. 139-150, 1989.
12. P. Lockhart, *On the discriminant of a hyperelliptic curve*, Trans. Amer. Math. Soc. 342 No.2, pp. 729-752, 1994.
13. A. Menezes, Y. Wu, R. Zuccherato, *An Elementary Introduction to Hyperelliptic Curves*. In: Koblitz, N., Algebraic Aspects of Cryptography, Springer-Verlag Berlin Heidelberg 1998. Available at http://www.cacr.math.uwaterloo.ca/techreports/1997/tech_reports97.html
14. V.S. Miller, *Use of Elliptic Curve in Cryptography*, In Advances in Cryptology-CRYPTO'85(Santa Barbara,Calif.,1985), LNCS.218, Spring-Verlag, pp.417-426, 1986.

15. J. Pila, *Frobenius maps of abelian varieties and finding roots of unity in finite fields*. Math.Comp., 55, pp.745-763, 1996.
16. H.G.Rück, *On the discrete logarithms in the divisor class group of curves*, Math.Comp., 68, pp.805-806, 1999.
17. T. Satoh, *Canonical Lifting of Elliptic Curves and p-Adic Point Counting - Theoretical Background*, Workshop on Elliptic Curve Cryptography - ECC'00, 2000. Available at
http://www.exp-math.uni-essen.de/ galbra/eccslides/eccslides.html
18. T. Satoh, and K. Araki, *Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves*, Commentari Math. Univ. St. Pauli 47 (1998), 81-92.
19. I.A. Semaev, *Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p*, Mathematics of Computation 67 (1998), 353-356.
20. J. Scholten, and Huijun Zhu, *Hyperelliptic Supersingular Curves over Fields of Characteristic 2*. Available at
http://www.math.berkeley.edu/ zhu/preprints.html
21. N.P. Smart, *The discrete logarithms problem on elliptic curves of trace one*, Journal of Cryptology 12 (1999), 193-196.
22. N.P. Smart, *Compressed ECC Parameters*. Available at
http://www.secg.org/collateral/compressed_ecc.pdf
23. J.A. Solinas, *Generalized Mersenne number*, Technical Reports, CACR, Waterloo, 1999. Available at:
http://www.cacr.math.uwaterloo.ca/techreports/1999/tech_reports99.html