FACTA UNIVERSITATIS (NIŠ)
Ser. Math. Inform. Vol. 33, No 5 (2018), 673–681
https://doi.org/10.22190/FUMI1805673K

EFFICIENT ENCODINGS TO HYPERELLIPTIC CURVES OVER FINITE FIELDS *

Amirmehdi Yazdani Kashani and Hassan Daghigh

Abstract. Many cryptosystems are based on the difficulty of the discrete logarithm problem in finite groups. In this case elliptic and hyperelliptic cryptosystems are more noticed because they provide good security with smaller size keys. Since these systems were used for cryptography, it has been an important issue to transform a random value in finite field into a random point on an elliptic or hyperelliptic curve in a deterministic and efficient method. In this paper we propose a deterministic encoding to hyperelliptic curves over finite field. For cryptographic desires it is important to have an injective encoding. In finite fields with characteristic three we obtain an injective encoding for genus two hyperelliptic curves.
Keywords: Cryptosystem; hyperelliptic curves; injective encoding; finite field.

1. Introduction

We first recall that a hyperelliptic curve H of gunes g is a curve by the equation $y^2 = f(x)$, where f a squarefree, monic polynomial of degree 2g + 1. Every hyperelliptic curve of genus 1 is called an elliptic curve. In fact an elliptic curve over the finite field $\mathbb{F}_q$ is the set $E(\mathbb{F}_q)$ which includes all of the points (x, y) such that

$$y^2 = x^3 + ax + b$$

where $a, b \in \mathbb{F}_q$ with an additional point that is called infinity. The points on $E(\mathbb{F}_q)$ with ∞ form an additive abelian group but for $g \geq 2$ there is not a group law on the points of a hyperelliptic curve. However the divisor group of H is denoted by $Div(H)$ is a free abelian group. A divisor $D \in Div(H)$ is a formal sum $D = \sum m_P P$ where $m_P \in \mathbb{Z}$ and $m_P = 0$ for all but finitely many $P \in H$. Then the degree of D is defined by $degD = \sum m_P$. The divisors of degree 0 form a subgroup of $Div(H)$ which is denoted by $Div^0(H)$. For every $f \in \mathbb{F}_q(H)$ the divisor of f is defined

Received December 06, 2018; accepted January 14, 2019
2010 *Mathematics Subject Classification.* Primary 14H52; Secondary 11T71
*The authors were supported in part by a grant from University of Kashan (808718/1)

by $div(f) = \sum ord_P(f)P$ where $ord_P(f)$ is the order of vanishing of f at P. A divisor $D \in Div(H)$ is called a principal divisor if it has the form $D = div(f)$ for some $f \in \mathbb{F}_q(H)$. Two divisors D_1, D_2 are called linearly equivalent if $D_1 - D_2$ is principal. The group of principal divisors of H is denoted by $Princ(H)$. Since every principal divisor has degree 0, $Princ(H)$ is a subgroup of $Div^0(H)$. The jacobian of H over $\mathbb{F}_q$ is defined by $J(H) = Div^0(H)/Princ(H)$. Since in many cryptosystems we need to a group we use the group $J(H)$ rather than the set of points on H. We have:

$$(\sqrt{q} - 1)^{2g} \leq \#J(H) \leq (\sqrt{q} + 1)^{2g}$$

Therefore $\#J(H) \approx q^g$.

2. Background

Encoding from finite fields element into the points of a given elliptic or hyperelliptic curve is a more challenging problem and requires to be studied more carefully. Before 2006 the usual method was Try and Increment. It was to take $x \in \mathbb{F}_q$ and check whether this value corresponds to a valid abscissa of a point on the elliptic curve. If not, try another abscissa until one of them works. One defect of this algorithm is that the number of operation is not constant. namely the number of steps depends on the input x.

Algorithm 1 Try and Increment Method

Require: : u an integer.
Ensure: : Q, a point of $E(\mathbb{F}_q)$.
for $i = 0$ **to** $k - 1$ **do**
(a) set $x = u + i$
(b) If x^3+ax+b is a quadratic residue in $\mathbb{F}_q$, then return $Q = (x, (x^3+ax+b)^{\frac{1}{2}})$
end for
return $\perp$

The twisted curves method was to apply curve and its twist as suggested in [6]. If E is defined by $y^2=x^3+ax+b$ over $\mathbb{F}_q$, the twist of E is a curve E^d defined by

$$dy^2=x^3+ax+b,$$

where d is a quadratic non-residue in $\mathbb{F}_q$. Then for every x there exists y such that the point (x, y) belongs to E or E^d. This method was little noticed since it requires calculation on two curves and this doubles the running time.

When $q \equiv 2 \ (mod\ 3)$ the map $x \to x^3$ is a bijection from $\mathbb{F}_q^*$ to itself. If E is defined by the equation $y^2 = x^3 + b$, the map

$$f : u \longrightarrow ((u^2 - b)^{\frac{1}{3}}, u)$$

gives a bijection from $\mathbb{F}_q$ to affine points on the curve E. Therefore these curves are supersingular for every b. The MOV attack gives an efficient computable method

which enables to reduce the DLP on a supersingular elliptic curve to DLP on a finite field [15]. Therefore in order to avoid this attack, much larger parameters must be used.

In 2006 the first algorithm for encoding to elliptic curves in deterministic polynomial time was proposed by Shallue and Woestijne [16]. The algorithm is based on the Skalba equality which says that there exist four maps $X_1(t), X_2(t), X_3(t), X_4(t)$ such that

$$f(X_1(t))f(X_2(t))f(X_3(t)) = (X_4(t))^2,$$

where $f(x) = X^3 + aX + b$. Then in a finite field for a fixed parameter t, there exists $1 \leqslant j \leqslant 3$ such that $f(X_j(t))$ is a quadratic residue. This implies that $(X_j(t), \sqrt{f(X_j(t))})$ is a point on $E : y^2 = f(x)$. For $q \equiv 3(mod\ 4)$ computing the square root $\sqrt{f(X_j(t))}$ is simply an exponentiation but for $q \equiv 1(mod\ 4)$, no deterministic algorithm has been found for computing the square root. If we have a non quadratic residue in $\mathbb{F}_q$ we can apply Tonelli Shanks algorithm to compute the square root. Using Skalba equality the authors of [16] show that a modification of Tonelli-Shanks algorithm can compute square roots deterministicaly in time $O(\log^4 q)$. Shallue-Woestijne method runs in time $O(\log^4 q)$ for any field size $q = p^n$ and in time $O(\log^3 q)$ when $q \equiv 3\,(mod\ 4)$. The maps were simplified and generalized to hyperelliptic curves of the forms $y^2 = x^n + ax + b$ and $y^2 = x^n + ax^2 + bx$ by Ulas in 2007 [18]. We recall these maps for elliptic curves in the following result.

Lemma 2.1. *Let* $f(x) = x^3 + ax + b$ *and*

$$\begin{aligned} X_1(t,u) &= u \\ X_2(t,u) &= \frac{-b}{a}(1 + \frac{1}{t^4 f(u)^2 + t^2 f(u)}) \\ X_3(t,u) &= t^2 f(u) X_2(t,u) \\ U(t,u) &= t^3 f(u)^2 f(X_2(t,u)). \end{aligned}$$

Then

$$U(t,u)^2 = f(X_1(t,u)).f(X_2(t,u)).f(X_3(t,u))$$

In 2009 Icart proposed another method for encoding to elliptic curves [13]. If $q \equiv 2\,(mod\ 3)$ the map $x \to x^3$ is a bijection in $\mathbb{F}_q$ and cube roots are uniquely defined with $x^{\frac{1}{3}} = x^{\frac{2q-1}{3}}$.Icart defined an encoding as follows:

$$f_{a,b}\colon \mathbb{F}_{p^n} \longrightarrow E_{a,b}$$

$$u \longrightarrow (x, y)\,,$$

where

$$x=(v^2-b-\frac{u^6}{27})^{\frac{1}{3}}+\frac{u^2}{3} \qquad y=ux+v \qquad v=\frac{3a-u^4}{6u}$$

He fixed $f_{a,b}(0) = O$, the neutral element of the elliptic curve. Icart proved that for all $p \in E_{a,b}$ the set $f_{a,b}^{-1}(p)$ is computable in polynomial time and

$\left|f_{a,b}^{-1}(p)\right| \leqslant 4$, namely a point has at most 4 preimages. He also proved that his algorithm works with complexity $O\left(\log^3 q\right)$ and conjectured that the image of $f_{a,b}$ contains $\frac{5}{8}.\#E_{a,b}(\mathbb{F}_q) + O(q^{\frac{1}{2}})$. Icart's conjecture was proved by Farashahi, Shparlinski and Voloch[9].

Brier et al [5] proposed a further simplification of the Shallue-Woestijne-Ulas algorithm for elliptic curves over finite field $\mathbb{F}_q$ with $q \equiv 3(mod\ 4)$. They showed every point $p = (x, y)$ has at most 8 preimages.

For cryptographic purposes it is important to have an injective encoding into an elliptic curve. In 2011 Farashahi [8] described an injective encoding to Hessian curves with a point of order 3 over $\mathbb{F}_q$ where $q \equiv 2(mod\ 3)$.
Fouque, Jeux and Tibouchi [10] proposed an injective encoding to elliptic curves of the form

$$E_c^\delta : y^2 = x^3 - 4\delta x^2 + \delta(c + \delta/c)^2 x,$$

where $c \in \mathbb{F}_q \backslash \{-1, 0, 1\}, \delta = \pm 1$.
Bernstein, Hamburg, Krosnova and Lange [3] proposed an injective encoding for elliptic curves of the form

$$E_{a,b} : y^2 = x(x^2 + ax + b)$$

with $a, b \in \mathbb{F}_q$.

Foque and Tibouchi [11] proposed a deterministic encoding in to hyperelliptic curves of the form

$$y^2 = x^{2g+1} + a_1 x^{2g-1} + \cdots + a_g x,$$

where g is the genus of the curve.
We need to take some security considerations for choosing a hyperelliptic curves. In this context, we have two important sequences:
1. If g is large there exists a subexponential algorithm for solving the discrete logarithm problems in $J(\mathbb{F}_q)$.[1]
2. If g is small such that $g \geqslant 5$ the attack by gaudry can solve discrete logarithm problem in $J(\mathbb{F}_q)$.[12]
Therefore for cryptographic desires we must consider the hyperelliptic curves of genus 2,3,4.

3. Main result

In this section we first propose an algorithm for encoding to hyperelliptic curves of the form $y^2 = x^n + ax^{n-1} + bx$ over finite field $\mathbb{F}_q$. Then we show our proposed method defines an injective encding where $n = 5$ (genus is 2) and q is a power of 3.

Lemma 3.1. *Let $g(x) = x^n + ax^{n-1} + bx$. If λ is a quadratic non-residue such that for some $x \in \mathbb{F}_q$ we have*

$$g(\lambda.x) = \lambda g(x) \tag{3.1}$$

then either x or $\lambda.x$ is the abscissa of a point on the $y^2 = g(x)$. Moreover for each λ the value

$$x = \frac{a(1-\lambda^{n-2})}{(\lambda^{n-1}-1)} \tag{3.2}$$

satisfies (3.1).

Proof. Since λ is not a quadratic residue, if x satisfies (3.1) then either $g(\lambda.x)$ or $g(x)$ must be a square in $\mathbb{F}_q$. Therefor either x or $\lambda.x$ must be abscissa of a point on the curve $y^2 = g(x)$. Moreover we have:

$$\begin{aligned} g(\lambda x) &= \lambda g(x) \\ (\lambda x)^n + a(\lambda x)^{n-1} + b(\lambda x) &= \lambda(x^n + ax^{n-1} + bx) \\ \lambda^{n-1}x^n + a\lambda^{n-2}x^{n-1} + bx &= x^n + ax^{n-1} + bx \\ \lambda^{n-1}x + a\lambda^{n-2} &= x + a \\ x &= \frac{a(1-\lambda^{n-2})}{(\lambda^{n-1}-1)}. \end{aligned}$$

□

Theorem 3.1. *Let $q \equiv 3(mod\ 4)$ and for any $t \in \mathbb{F}_q$*

$$\begin{aligned} X_1(t) &= \frac{a(1-(-t)^{2n-4})}{((-t)^{2n-2}-1)} \\ X_2(t) &= -t^2 X_1(t) \\ U(t) &= tg(X_1(t)) \end{aligned}$$

Then

$$(U(t))^2 = -g(X_1(t))g(X_2(t))$$

Proof. since $q \equiv 3(mod\ 4)$, -1 is a quadratic non-residue and we can take $\lambda = -t^2$ in previous lemma. Therefore $X_1(t) = x$ and $X_2(t) = \lambda x$ and we have:

$$\begin{aligned} g(X_1(t))g(X_2(t)) &= g(x)g(\lambda.x) = \lambda g(x)^2 = -t^2 g(x)^2 \\ &= -(tg(x))^2 = -(U(t))^2 \end{aligned}$$

□

Remark 3.1. Let $P = (X_P, Y_P)$ be a point generated by this method. We solve the equations $X_1(t) = X_P$ and $X_2(t) = X_P$ to compute the pre-images of P. Since $degX_1(t) = 2n-2$ and $degX_2(t) = 2n-2$ each equation has at most $2n-2$ solutions. The minus sign in the final step of the algorithm makes that set of points obtained of form $(X_1, g_1^{\frac{q+1}{4}})$ and set of points obtained of form $(X_2, -g_2^{\frac{q+1}{4}})$ are separated. Hence a point has at most $2n-2$ pre-images.

Algorithm 2 Encoding Algorithm

Require: : $\mathbb{F}_q$ such that $q \equiv 3(mod\ 4)$, parameters $a, t \in \mathbb{F}_q$.
Ensure: : $(x, y) \in H_{n,a,b}(\mathbb{F}_q)$ where $H_{n,a,b} : y^2 = x^n + ax^{n-1} + bx$.
If $t = 0$ then return $(0, 0)$
If $t = \pm 1$ then return O
$\lambda \longleftarrow -t^2$
$X_1 \longleftarrow \frac{a(1-\lambda^{n-2})}{(\lambda^{n-1}-1)}$
$X_2 \longleftarrow \lambda.X_1$
$g_1 = X_1^n + aX_1^{n-1} + bX_1; g_2 = X_2^n + aX_2^{n-1} + bX_2$
If g_1 is a square, return $(X_1, g_1^{\frac{q+1}{4}})$, otherwise return $(X_2, -g_2^{\frac{q+1}{4}})$

3.1. Injective encoding

In this section we express an injective encoding for hyperelliptic curves of the form $H_{2,a,b} : y^2 = x^5 + ax^4 + bx$. If we want to use our proposed algorithm for $n = 5$ we have:

$$X_1(\lambda) = \frac{a(1-\lambda^3)}{(\lambda^4-1)} \qquad X_2(\lambda) = \frac{a(\lambda-\lambda^4)}{(\lambda^4-1)}$$

for every quadratic non-residue λ.
If $X_1(\lambda_1) = X_1(\lambda_2)$ we have:

$$\lambda_1^4 - \lambda_1^4\lambda_2^3 + \lambda_2^3 - \lambda_2^4 + \lambda_1^3\lambda_2^4 - \lambda_1^3 = 0.$$

We divide the sides of this equation by $1 - \lambda_1$ and $1 - \lambda_2$ and $\lambda_1 - \lambda_2$. Then we have:

$$(\lambda_2^2 + \lambda_2 + 1)\lambda_1^2 + (\lambda_2^2 + \lambda_2)\lambda_1 + \lambda_2^2 = 0. \tag{3.3}$$

The discremnant of equation 3.3 is $\Delta_1 = \lambda_2^2(-3\lambda_2^2 - 2\lambda_2 - 3)$. Therefore if Δ_1 is a quadratic non-residue, this equation has no solution.
It also follows from $X_2(\lambda_1) = X_2(\lambda_2)$ that:

$$\lambda_1^4 - \lambda_2^4 + \lambda_1\lambda_2^4 - \lambda_1^4\lambda_2 - \lambda_1 + \lambda_2 = 0.$$

Similarly if we divide the sides of this equation by $1 - \lambda_1$ and $1 - \lambda_2$ and $\lambda_1 - \lambda_2$, we have:

$$\lambda_1^2 + \lambda_1(\lambda_2 + 1) + \lambda_2^2 + \lambda_2 + 1 = 0 \tag{3.4}$$

The discremnant of equation 3.4 is $\Delta_2 = -3\lambda_2^2 - 2\lambda_2 - 3$. Therefore if Δ_2 is a quadratic non-residue, this equation has no solution. By looking at equations $\Delta_1 = \lambda_2^2(-3\lambda_2^2 - 2\lambda_2 - 3)$ and $\Delta_2 = -3\lambda_2^2 - 2\lambda_2 - 3$, it can be concluded that they are quadratic non-residues if for any λ as quadratic non-residue the value $\Delta = -3\lambda^2 - 2\lambda - 3$ is a quadratic non-residue.

Definition 3.1. Let p be a prime number and $q = p^n$ for $n \in \mathbb{N}$. If $\{\beta_1, \ldots, \beta_{n-1}\}$ be a basis for $\mathbb{F}_q$ over $\mathbb{F}_p$, for every element $a \in \mathbb{F}_q$ we have:

$$a = a_0 + a_1\beta + \cdots + a_{n-1}\beta^{n-1} \qquad a_i \in \mathbb{F}_p$$

We define set S which consists of half the field elements as follows:

$$S = S_0 \cup S_1 \cup \cdots \cup S_{n-1}$$

Such that

$$\begin{aligned} S_0 &= \{(a_0, a_1, \ldots, a_n) : 0 < a_0 \leq \frac{p-1}{2}, \forall 1 \leq i \leq n \quad a_i = 0\} \\ S_1 &= \{(a_0, a_1, \ldots, a_n) : 0 < a_1 \leq \frac{p-1}{2}, \forall 2 \leq i \leq n \quad a_i = 0\} \\ &\vdots \\ S_{n-1} &= \{(a_0, a_1, \ldots, a_n) : 0 < a_n \leq \frac{p-1}{2}\}. \end{aligned}$$

It is easy to see that S has cardinality $\frac{p^n-1}{2}$ and for each $x \in \mathbb{F}_q$ exactly one of x or $-x$ is in $\mathbb{F}_q$.

Corollary 3.1. *If we consider $H_{2,a,b}$ over finite fields of characteristic 3, the algorithm 2 defines an injective encoding from S into points $H_{2,a,b}$.*

Proof. Since $Char(\mathbb{F}_q) = 3$ we have $\Delta = \lambda$. Therefore the Δ value is always a quadratic non residue. Since each of λ comes by two values $\pm t$, every point (x, y) in the outpot of this algorithm has exactly 2 preimages in $\mathbb{F}_q$. Therefore for the elements of S we have an injective encoding. □

Remark 3.2. We know that the set of points on $H_{n,a,b}$ is not a group. Therefore if for cryptographic purposes we need to be in a group, we can map $H_{n,a,b}$ to the jacobian J of $H_{n,a,b}$ which is an abelian group. If we use the jacobian of a hyperelliptic curve instead of an elliptic curve over a finite field $\mathbb{F}_q$ we can reduce key size by having the same level of security. In our case by using a hyperelliptic curve of genus 2 over a finite field $q \simeq 3^{80}$ we have the same level of security when we use an elliptic curve group where $q \simeq 3^{160}$.

REFERENCES

1. L. ADLEMAN, J. DEMARRAIS and M. HUANG: A subexponential algorithm for discrete logarithms over the rational subgroup of the jacobians of large genus hyperelliptic curves finite fields. Algorithmic Number Theory, LNCS 877 (**1994**), 28-40.

2. D. J. BERNSTEIN, P. BIRKNER, M. JOYE, T. LANGE and C. PETERS: Twisted edwards curves. In: International Conference on Cryptology in Africa, Springer, Berlin, Heidelberg, 2008, p. 389-405.
3. D. J. BERNSTEIN, M. HAMBURG, A.KROSNOVA and T. LANGE: Elliptic-curve points indistinguishable from uniform random strings. In: Proceedings of the 2013 ACM SIGSAC conference on Computer and communications security, ACM, 2013, p. 967-980.
4. D. BONEH, M. FRANKLIN: Identity-based encryption from the Weil pairing. In: Annual international cryptology conference, Springer, Berlin, Heidelberg, 2001, p. 213-229.
5. E. BRIER, J. -S. CORON, T. ICART, D. MADORE, H. RANDRIAM and M. TIBOUCHI: Efficient indifferentiable hashing into ordinary elliptic curves. In: Annual Cryptology Conference, Springer, Berlin, Heidelberg, 2010, p. 237-254.
6. O. CHEVASSUT, P. A. FOUQUE, P. GAUDRY and D. POINTCHEVAL: The twist-augmented technique for key exchange. In: International Workshop on Public Key Cryptography, Springer, Berlin, Heidelberg, 2006, p. 410-426.
7. H. COHEN: A course in computational algebraic number theory. volume 138 of Graduate Texts in Mathematics, Springer-Verlag, Berlin, 1993.
8. R. R. FARASHAHI: Hashing into Hessian curves. In: International Conference on Cryptology in Africa, Springer, Berlin, Heidelberg, 2011, p. 278-289.
9. R. R. FARASHAHI, I. E. SHPARLINSKI and J. F. VOLOCH: On hashing into elliptic curves. Journal of Mathematical Cryptology, 2009, 3.4: 353-360.
10. P. A. FOUQUE, A. JOUX and M. TIBOUCHI: Injective encodings to elliptic curves. In: Australasian Conference on Information Security and Privacy, Springer, Berlin, Heidelberg, 2013, p. 203-218.
11. P.A. FOUQUE, M. TIBOUCHI: Deterministic encoding and hashing to odd hyperelliptic curves. In: International Conference on Pairing-Based Cryptography, Springer, Berlin, Heidelberg, 2010, p. 265-277.
12. P. GAUDRY: An algorithm for solving the discrete log problem on hyperelliptic curves. In: International Conference on the Theory and Applications of Cryptographic Techniques, Springer, Berlin, Heidelberg, 2000, p. 19-34.
13. T. ICART: How to hash into elliptic curves. In: Advances in Cryptology-CRYPTO 2009, Springer, Berlin, Heidelberg, 2009, p. 303-316.
14. N. KOBLITZ : Elliptic curve cryptosystems. Mathematics of computation, 1987, 48.177: 203-209.
15. A. J. MENEZES, T. OKAMOTO and S. A. VANSTONE: Reducing elliptic curve logarithms to logarithms in a finite field. iEEE Transactions on information Theory, 1993, 39.5: 1639-1646.
16. A. SHALLUE, C. E. VAN DE WOESTIJNE: Construction of rational points on elliptic curves over finite fields. In: International Algorithmic Number Theory Symposium, Springer, Berlin, Heidelberg, 2006, p. 510-524.
17. J.H. SILVERMAN: Advanced Topics in the Arithmetic of Elliptic Curves. Grad. Texts in Math., vol. 151, Springer, New York, 1994.
18. M. ULAS: Rational points on certain hyperelliptic curves over finite fields. Bull. Polish Acad. Sci. Math., 55(2), 2007: 97-104.

Amirmehdi Yazdani Kashani
Faculty of Mathematical Sciences
Department of Pure Mathematics
University of Kashan, Iran
yazdani@grad.kashanu.ac.ir

Hassan Daghigh
Faculty of Mathematical Sciences
Department of Pure Mathematics
University of Kashan, Iran
hassan@kashanu.ac.ir