A COLLECTION OF PAPERS IN MATHEMATICS AND RELATED SCIENCES,
A FESTSCHRIFT IN HONOUR OF THE LATE GALAYE DIA
EDITORS : PROFS HAMET SEYDI, GANE SAMB LO, ABOUBAKARY DIAKHABY
SPAS-EDS (SAINT-LOUIS, CALGARY, 2018), WWW.STATPAS.ORG/SPASEDS/
IN EUCLID (WWW.PROJECTEUCLID.ORG)
DOI : 10.16929/SBS/2018.100

CHAPTER 29

A note on Encoding and Hashing into Elliptic and Hyperelliptic Curves, *by N. Diarra, M. Seck and D. Sow*

Nafissatou Diarra. Email : nafissatou.diarra@ucad.edu.sn
Michel Seck. Email : michel.seck@ucad.edu.sn
Djiby Sow. Email : sowdjibab@yahoo.fr
Cheikh Anta Diop University, Dakar, Senegal, Department of Mathematics and Computer Science

Abstract: We give an (up-to-date) overview of existing encoding and hash functions into (hyper)elliptic curves. We also design an indifferentiable hash function into the Jacobian of certain families of hyperelliptic curves of genus $g \leq 5$ using the unified formulas of Seck and Diarra (AFRICACRYPT-2018).

Keywords. Elliptic Curve Cryptography; Hyperelliptic curves; Encoding; Hashing .
AMS 2010 Mathematics Subject Classification. 11T71; 14G50

Cite the chapter as :
Diarra N., M. Seck M. and D. Sow D.(2018). A Note on Encoding and Hashing into Elliptic and Hyperelliptic Curves . In *A Collection of Papers in Mathematics and Related Sciences, a festschrift in honour of the late Galaye Dia* (Editors : Seydi H., Lo G.S. and Diakhaby A.). Spas Editions, Euclid Series Book, pp. 615 –648.
Doi : 10.16929/sbs/2018.100-06-01

1. Introduction

When hashing into the Jacobian of an (hyper)elliptic curve, we need a function that maps in a deterministic way an element of a finite field $\mathbb{F}_q$ to a point of the curve. Such function is called an encoding . We need encoding functions into (Hyper)Elliptic Curves for globally two reasons:

(1) **representation of points**: if an encoding f is almost-injective (it is injective when restricted to a certain subset of $\mathbb{F}_q$, see Bernstein *et al.* (2013)) and invertible (we can find a preimage for any element in the image set), it can be used to represent each point of the image set by a random uniform string of bits, like what Bernstein *et al.* (2013) did for Elligator(1 and 2).

(1) **indifferentiable hashing**: if an encoding f is well-distributed (that is, the sum of its images relatively to a fixed associated character, depending on the security parameter, is bounded), one can use f to design an indifferentiable hash function onto the targeted curve, following the general process proposed by Brier *et al.* (2010) and Farashahi *et al.* (2013).

The problem of encoding into elliptic and hyperelliptic curves was investigated by many authors. This problem can be formulated as follows: Given an element r in a finite field $\mathbb{F}_q$ and an (hyper)elliptic curve H, map this element r into H. The first proposal (to our knowledge) of encoding functions into elliptic curves was done by Boneh and Franklin (2001) for their IBE scheme. Since then, there have been many proposals of encoding into different forms of elliptic curves:

• for the Weierstrass model $E : y^2 = x^3 + a_2x^2 + a_4x + a_6$: Boneh and Franklin (2001), Shallue and van de Woestijne (2006), Icart (2009), Brier *et al.* (2010), Fouque and Tibouchi (2012), Fouque *et al.* (2013), Bernstein *et al.* (2013), Diarra *et al.* (2017), Seck and Diarra (2018);

• for the Hessian form $E : x^3 + y^3 + 1 = 3dxy$: Kammerer *et al.* (2010), Farashahi (2011);

• for Montgomery curves $E_{a,b} : by^2 = x^3 + ax^2 + x$: Yu *et al.* (2013);

• for the Edwards model $E_d : x^2 + y^2 = 1 + dx^2y^2$: Bernstein *et al.* (2013), Yu *et al.* (2016), Diarra *et al.* (2017);

• for the Jacobi Quartic model $E: y^2 = x^4 + 2bx^2 + 1$: Yu et al. (2015);

• for the Huff models $E: ax(y^2 - c) = by(x^2 - d)$ and $E_{a,b}x(ay^2 - 1) = y(bx^2 - 1)$: Yu *et al.* (2016) and Diarra *et al.* (2017);

• for the Jacobi Intersection model $au^2 + v^2 = 1, bv^2 + w^2 = 1$: He *et al* (2017).

For the hyperelliptic case also, many authors have proposed encoding for different families of curves, like the encoding of Ulas (2007), Kammerer *et al.* (2010), Fouque and Tibouchi (2010), Seck *et al.* (2017), Seck and Diarra (2018).

Using the existing encoding, some constructions of hash functions into the Jacobian of certain families of (hyper)elliptic curves have been proposed. The first construction was the *try-and-increment method* (Boneh *et al.* (2001)). Other constructions were proposed by Brier *et al.* (2010): constructions of the form $H = f \circ h$, $H = f \circ h_1 + h_2\mathbb{G}$ and $H = f \circ h_1 + f \circ h_2$ later generalized by Farashahi *et al.* (2013) ($H = f \circ h_1 + f \circ h_2 + \ldots + f \circ h_s$).

Our contributions:

(a) We first give a complete overview of all existing encoding into both elliptic and hyperelliptic curves and an overview of all techniques of hashing into the Jacobian of an (hyper)elliptic curve.

(b) Then we do a comparative analysis of the different methods of construction of deterministic encoding. We also compare existing methods for constructing indifferentiable hash functions.

(c) To finish, we propose new unified formulas for indifferentiable hashing into the Jacobian of certain family of hyperelliptic curves (with genus $g \leq 5$).

Organization of the paper: This paper is organised as follows. In Section 2, we give some preliminaries such as definition of the square root function and of a quadratic character in finite fields, (hyper)elliptic curve, encoding function and the Riemann-Hurwitz formula. In Section 3, we give an overview of all encoding into (hyper)elliptic curves. In Section 4, we give an overview of all methods of hashing into the Jacobian of an elliptic and hyperelliptic curve. In Section 5, we design a new indifferentiable hash function using the unified encoding of Seck and Diarra (2018). And we conclude in Sect. 6.

2. Preliminaries

We recall some definitions and properties on character sums, quadratic character, (hyper)elliptic curves, Riemann-Hurwitz formula and encoding functions into curves.

2.1. Characters and Square Root on Finite Abelian Groups.

Definition 10. Let G be a finite abelian group (written multiplicatively). A character on G is a group homomorphism $\chi : G \to \mathbb{C}^*$, where $\mathbb{C}^*$ is the multiplicative group of the nonzero complex numbers.

- For a character χ on G, we have $\chi(g_1g_2) = \chi(g_1)\chi_2(g_2)$ and $\chi(1) = 1$.

- The trivial character 1_G is the function on G where $1_G(g) = 1$ for all $g \in G$.

- The set of characters on G, together with the multiplication $(\chi_1\chi_2)(g) = \chi_1(g)\chi_2(g)$ is an abelian group called character group of G, and denoted by $\bar{G}$.

Quadratic Character and Square Root.

Let q be an odd prime power, and denote by $\mathbb{F}_q$ the finite field of q elements.

- The **quadratic character** (or generalized Legendre symbol) is the function χ_q defined by $\chi_q : \mathbb{F}_q \to \mathbb{F}_q : u \mapsto \chi_q(u) = u^{(q-1)/2}$ and verifying: $\chi_q(u) = 1$ if u is a non-zero square; $\chi_q(u) = -1$ if u is a non-square; and $\chi_q(u) = 0$ if $u = 0$. The following properties are also verified: $\chi_q(uv) = \chi_q(u) \cdot \chi_q(v)$ for any $u, v \in \mathbb{F}_q$; $\chi_q(a^2) = 1$ for any $a \in \mathbb{F}_q^*$; and if $q \equiv 3 \mod 4$, $\chi_q(-1) = -1$, $\chi_q(\chi_q(u)) = \chi_q(u)$, for any $u \in \mathbb{F}_q$. If $q \equiv 1 \mod 4$, then $\chi_q(-1) = 1$.

- **Square Root:** Define $\mathbb{F}_q^2 = \{a^2 : \ a \in \mathbb{F}_q\}$. A square root function $\sqrt{\ }$ for $\mathbb{F}_q$ is defined by:

$$\sqrt{\ } : \ \mathbb{F}_q^2 \to \mathbb{F}_q : a^2 \mapsto \sqrt{a^2} = \pm a.$$

For $q \equiv 3 \mod 4$, $a^{\frac{q+1}{4}}$ is called the principal square root of a; therefore one can take the principal square root as a square-root function. For odd prime q, one can take $\sqrt{\mathbb{F}_q^2} = \{0, 1, ..., \frac{q-1}{2}\}$.

2.2. Elliptic and Hyperelliptic Curves . Hyperelliptic curves are a special class of algebraic curves and can be viewed as generalizations of elliptic curves. Let $\mathbb{K}$ be a field and $\overline{\mathbb{K}}$ its algebraic closure.

Definition 11. *An Hyperelliptic curve $\mathcal{C}$ of genus $g \geq 1$ over $\mathbb{K}$ is given by an equation of the form $\mathcal{C}\ :\ \ y^2 + h(x)y = f(x)$, where:* ***(i)*** *$h \in \mathbb{K}[x]$ is a polynomial of degree at most g,* ***(ii)*** *$f \in \mathbb{K}[x]$ is a monic polynomial of degree $2g+1$, and* ***(iii)*** *there are no point in $\mathcal{C}$ over $\overline{\mathbb{K}}$ which simultaneously satisfy the partial derivative $2y + h = 0$ and $yh' - f' = 0$.*

The last point means that an Hyperelliptic curve does not have a *singular* point, which is equivalent to f has no multiple roots. Note that there exist Hyperelliptic curves for any genus $g \geq 1$, and some Hyperelliptic curves verify $\deg(f) = 2g + 2$. When $g = 1$, one obtains an *elliptic curve* in Weierstrass form $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$ (see Silverman (1986); Menezes *et al.* (1998); Koblitz (1989); Scholten and Vercauteren (2008) for more details about elliptic and Hyperelliptic curves).

Some elliptic curves are of interest in Cryptography (because of a faster addition law for example), like Edwards model, Huff model, Hessian model, etc. All these curves are birationnaly equivalent to the Weierstrass model.

2.3. Riemann Hurwitz Formulas.

We recall here the Riemann Hurwitz formulas which relates the ramification of a separable morphism to its degree, as well as the genus of the curves

Theorem 87. *Let $\phi : X \to Y$ be a separable morphism of curves, of degree d. Let g_X and g_Y be the genus of X and Y respectively. Then we have $2g_X - 2 \geq d(2g_Y - 2) + \sum_{P \in X}(e_P - 1)$ with equality if and only if ϕ is tamely ramified. e_P denote the ramification index at P.*

3. Overview of (Hyper)Elliptic Curves Encoding

3.1. Elliptic Curves Encoding.

We give in this subsection, an overview of main existing encoding into different forms of elliptic curves such that Weierstrass form, Montgomery form, Jacobi quartic form, Huff form and Edwards form etc.

3.1.1. *Encoding into Weierstrass Curves.*

• **Naive method** : For an elliptic curve $E_{a,b}: \ y^2 = x^3 + ax + b$, the simplest way to construct a point of $E_{a,b}$ is to use the naive method. The idea is to pick a x-coordinate and try to deduce the y-coordinate by computing a square root: choose a random element $u \in \mathbb{F}_q^*$ and compute $u^3 + au + b$; and then test whether $u^3 + au + b$ is a square in $\mathbb{F}_q$. If yes, then return $(x, y) = (u, \pm\sqrt{u^3 + au + b})$ as a point of the curve. Otherwise, one can choose another u in $\mathbb{F}_q$ and try again. But this method has at least one drawback, that is it cannot run in constant time: the number of operations depends on the input u. In practice the input u is the message m we want to hash; thus running this algorithm can allow the attacker to guess some information about m.

• **Encoding for supersingular curves** : A *supersingular* curve over $\mathbb{F}_q$ is an elliptic curve E such that $|E(\mathbb{F}_q)| = q+1$. For $q \equiv 2 \mod 3$, the curve defined by $E_b : y^2 = x^3 + b$ is a supersingular one. In their identity-based scheme Boneh and Franklin (2001), proposed the function $f : u \mapsto ((u^2 - b)^{\frac{1}{3}}, u)$ that constructs a point of E_b, given any $u \in \mathbb{F}_q$. This function allows them to construct the public key Q_{id}(a point on the supersingular curve) corresponding to the identity $\mathrm{id} \in \{0, 1\}^*$.

But it is well-known that supersingular curves are useless for cryptographic concerns because of the MOV attack Menezes *et al.* (1993): that is the DLP on E_b can be reduced to the DLP in $\mathbb{F}_q$. To avoid these attack, a large q should be used.

• **SWU's algorithm** *(2006)* : Shallue and van de Woestijne (2006) proposed an algorithm that generates, in polynomial time, a point of an elliptic curve E over $\mathbb{F}_q$ (of characteristic ≥ 5) given by its Weierstrass equation $y^2 = g(x) = x^3 + a_2x^2 + a_4x + a_6$, with $a \neq 0$. Their encoding is based on the Skalba (2005)'s equality: there exist four non-constant rational functions $X_1(t), X_2(t), X_3(t), X_4(t)$ such that:

$$g(X_1(t)) \cdot g(X_2(t)) \cdot g(X_3(t)) = (X_4(t))^2$$

It follows that at less one of the $g(X_i(u))$ must be a quadratic residue in the finite field $\mathbb{F}_q$, given a fixed $u \in \mathbb{F}_q$. One can then deduce an encoding function to the curve $E : \ y^2 = g(x)$. It suffices to set $(x, y) = (X_i(u), \sqrt{g(X_i(u))})$, where i is the smallest index such that $g(X_i(u))$ is a quadratic residue.

Even if this construction defines a constant-time encoding, it presents at least one drawback. In fact, the rational functions $X_i(t)$ are large and complex enough to make them difficult to implement. And there is no deterministic polynomial time for computing a square root in $\mathbb{F}_q$, unless making additional hypotheses on q. For example, when $q \equiv 3 \mod 4$, then computing a square root is simply an exponentiation.

Note that some authors used this general method to construct encoding for particular families of elliptic curves (like BN curves [3.1.1], generalized Huff curves [3.1.4], etc.) and Hyperelliptic curves ([3.2]).

• **Icart's encoding** (*2009*) : Let $q = 2 \mod 3$; so the map $x \mapsto x^3$ is a bijection and then computation of a cubic root can be done as an exponentiation. Icart (2009) defined a new encoding function, based on the following idea: intersect the line $y = ux + v$ with the Weierstrass curve $E_{a,b} : \; y^2 = x^3 + ax + b$, with $a, b \in \mathbb{F}_q$. He defined the encoding function: $f_{a,b} : \; \mathbb{F}_q \to E_{a,b} : u \mapsto f_{a,b}(u) = (x, ux + v)$, where $x = (v^2 - b - \frac{u^6}{27})^{1/3} + \frac{u^2}{3}$ and $v = (3a - u^4)/6u$. As shown in the paper, this function presents many interesting properties. In fact, it can be implemented in polynomial time with $O(\log^3 q)$ operations. The inverse function $f_{a,b}^{-1}$ is also computable in polynomial time. Icart also showed that $|f_{a,b}^{-1}(P)| \leq 4$, given a point P on the elliptic curve. This results from the fact that to compute $f_{a,b}^{-1}(P)$, it's sufficient to solve the degree 4 polynomial over $\mathbb{F}_q$:

$$u^4 - 6u^2x + 6uy - 3a = 0 \tag{3.1}$$

Moreover, Icart showed that the cardinal of the image set $\mathrm{Im}(f_{a,b})$ is greater than $q/4$ and made the following conjecture (proved later by Fouque *et al.* (2013)) : the size of $\mathrm{Im}(f_{a,b})$ is approximately $\dfrac{5}{8}$ of the size of the curve $E_{a,b}$.

• **Simplified Ulas's encoding** (*2010*) : Brier *et al.* (2010) gave a simplified version of Ulas's encoding when $q \equiv 3 \mod 4$ (Ulas generalized the SWU algorithm to Hyperelliptic curves of the form $y^2 = x^n + ax + b$ or $y^2 = x^n + ax^2 + bx$). In fact, the authors obtained formulas for the Weierstrass form $y^2 = x^3 + ax + b$, by expliciting the rational functions described in SWU algorithm. Their maps are defined by the following proposition:

PROPOSITION 29. *Let* $q \equiv 2 \mod 3$*, and* $g(x) = x^3 + ax + b$ *(with* $a, b \in \mathbb{F}_q^*$*). Let* $X_2(t) = -\frac{b}{a}\left(1 + \frac{1}{t^4 - t^2}\right)$, $X_3(t) = -t^2 X_2(t)$, $U(t) = t^3 g(X_2(t))$ *Then* $U(t)^2 = -g(X_2(t)) \cdot g(X_3(t))$*.*

This allows them to define an $\alpha-$weak encoding (with $\alpha = 8N/q$ and N is the order of the curve) into the Weierstrass model $y^2 = x^3 + ax + b$.

• **Encoding into BN Curves** (*2012*) : A Barreto-Nahrig (BN) curve is a pairing-friendly elliptic curve of the form $E_b: \ y^2 = g(x) = x^3 + b$ over a field $\mathbb{F}_q$, where $q \equiv 1 \mod 3$, $\#E(\mathbb{F}_q)$ is prime and b is the smallest interger > 0 such that $1 + b$ is a non-zero square in $\mathbb{F}_q$. A BN curve is then a particular case of curves taken in account by the SWU's algorithm, with $a_2 = a_4 = 0$ and $a_6 = b$.

Using this fact, Fouque and Tibouchi (2012) proposed an encoding for BN curves, by making explicit the encoding function of Shallue and Woestjine (see previous paragraph). This leads to the following encoding function:

$$f: \ \mathbb{F}_q \to E_b(\mathbb{F}_q), \ t \mapsto f(t) = \begin{cases} \left(\frac{-1+\sqrt{-3}}{2}, \sqrt{1+b}\right) & \text{if } t = 0 \\ (x_i, \chi(t) \cdot \sqrt{g(x_i)}) & t \neq 0 \end{cases}$$

where χ is the quadratic character on and i is the smallest index in $\{1, 2, 3\}$ such that $g(x_i)$ is a square (the values of x_1, x_2, x_3 are given explicitly in the same paper Fouque and Tibouchi (2012)).

The authors also gave an estimation of the size of the image set (roughly $9/16$ of the size of the curve E_b).

• **Injective encoding** (*2013*) : The problem of constructing injective encoding for a large family of elliptic curves (including all curves with a point of order 4 and only one point of order 2, and curves with points of order 2) was investigated in Fouque *et al.* (2013). In their paper, they proposed a way for constructing an injective encoding for an elliptic curve viewed as a quotient of an odd Hyperelliptic curve (e.g. an Hyperelliptic curve of the form $y^2 = f(x) = x^{2g+1} + a_1 x^{2g-1} + \ldots + a_g(x)$). The elliptic curves compatible with their encoding are of the form: $y^2 = x^3 \pm 4x^2 + Ax$ (see Fouque *et al.* (2013), pages 11-12). The authors proposed later a revisited version in which they gave formulas for computing the injective encoding for the short Weierstrass model (and also for the Edwards form by using a birational equivalence). They encode from a subset I of $\mathbb{F}_q$ of cardinality $(q{+}1)/2$ and such that $I \cap -I = \{0\}$.

Some authors used later their generic method to construct encoding for particular families of elliptic curves (like Edwards curves by Bernstein *et al.* [3.1.3], or generalized Huff curves by Diarra *et al.* [3.1.4]).

• **Elligator 2** (*2013*) : In the same paper than Elligator 1 (for Edwards curves), Bernstein *et al.* (2013) introduced a second encoding function that they called *Elligator 2*, for the Weierstrass model $E_{A,B}: \ y^2 = x^3 + Ax^2 + Bx$ with $AB(A^2 - 4B) \neq 0$ over any finite field $\mathbb{F}_q$ of odd characteristic. Elligator 2 maps any element of a particular subset R of $\mathbb{F}_q$ to a point on the Weierstrass curve. And, for a suitable choice of q (namely $q \equiv 1 \mod 4$), one obtains $R = \mathbb{F}_q$, e.g. the encoding cover all elements of $\mathbb{F}_q$. In Elligator 2, to encode a nonzero $r \in R = \{r \in \mathbb{F}_q : \ 1 + ur^2 \neq 0, \ A^2ur^2 \neq B(1 + ur^2)^2\}$, one should do the following: first compute $v = -A(1 + ur^2)$, and then return (x, y) as a point on the curve $E_{A,B}$ where $x = v$, $y = -\sqrt{x^3 + Ax^2 + Bx}$, if $v^3{+}Av^2{+}Bv$ is a quadratic residue in $\mathbb{F}_q$; and $x = -v{-}A$, $y = \sqrt{x^3 + Ax^2 + Bx}$ otherwise.

Elligator 2 is almost-injective; hence it can be used for representing points on the curve as uniform random strings of bits, as showed by the authors.

• **Encoding for Weiertrass model in characteristic two** : An elliptic curve over a binary field $\mathbb{F}_{2^n}$ is a curve of the form $E_{a,b}: \ y^2 + xy = x^3 + ax^2 + b$, where $a, b \in \mathbb{F}_{2^n}$. Icart (2009) applied his method for short Weierstrass model (with the computation of a cubic root) to obtain a simple and efficient encoding for binary elliptic curves (assuming that n is odd, and thus $x \mapsto x^3$ is a bijection). This encoding in characteristic two satisfies the same properties as in even characteristic.

Brier *et al.* (2010) also proposed a variant of their Weierstrass encoding over binary fields, using the Shalue-Woestjine algorithm.

• **Encoding for Weierstrass model in characteristic three** : Ordinary elliptic curves over $\mathbb{F}_3^n$ are represented by a Weierstrass equation $E_{a,c}: \ y^2 = f(x) = x^3 + ax^2 + c$, with discriminant $\Delta = -a^3b \neq 0$. Brier *et al.* (2010) were the first to propose a deterministic encoding for elliptic curves in characteristic three. Using the Elligator 2 method, Diarra *et al.* (2017) also proposed a direct encoding from $\mathbb{F}_3^n$ into such curves (where $-ac$ is a square). Given a non-zero element r in $\mathbb{F}_3^n$, their encoding works as follows:

(i) since $-ac$ is a square, set $s^2 = -c/a$, and compute $v = \dfrac{ur^2 - s^2}{s}$;
(ii) return (x, y) as a point on $E_{a,c}$, where

$$\begin{cases} x = v \text{ and } y = -\sqrt{f(x)} & \text{if } f(v) \text{ is a square;} \\ x = \dfrac{sv}{s+v} \text{ and } y = \sqrt{f(x)} & \text{otherwise.} \end{cases}$$

This encoding shares all properties of Elligator 2, like pseudo-injectivity and characterizable image set.

3.1.2. *Encoding into Hessian Curves.*

- Kammerer *et al.* (2010) : They proposed encoding for various models of (hyper)elliptic curves, including the Hessian model $E_d: \ x^3 + y^3 + 1 = 3dxy$ $(d \neq 1)$ over $\mathbb{F}_q$, with $q \equiv 2 \mod 3$. To define this encoding, they solved curve equations in radicals (like in Icart (2009)'s method) for the Weierstrass model $E_a \ : \ Y^2 + XY + aY = X^3$, which is birationnaly equivalent to the curve E_d. Their encoding is a weak one in the sense of definition (15) , and is $2:1$ (compared to Icart's encoding which is $4:1$).Their encoding can be extended to $\mathbb{F}_q$ and can be defined when $d = 2$ (see Kammerer *et al.* (2010)).

- **Farashahi** *(2011)* : Let $d \in \mathbb{F}_q$ with $d^3 \neq 1$. A **Hessian curve**(a curve with a point of order 3) H_d over $\mathbb{F}_q$ is given by the equation $x^3 + y^3 + 1 = 3dxy$. For $q \equiv 2 \mod 3$, Farashahi (2011) applied Icart's method to the set $H_d(\mathbb{F}_q)$ of $\mathbb{F}_q$-rational points of H_d. He obtained the function: $h_d: \ \mathbb{F}_q to H_d(\mathbb{F}_q)$ defined by $h_d(u)$

$= (x, y)$ if $u \neq -1$, where $x = -u\left(\dfrac{d^3u^3+1}{u^3+1}\right)^{1/3}, \ v = -\left(\dfrac{d^3u^3+1}{u^3+1}\right)^{1/3} + du;$

or

$= \mathcal{O}$ if $u = -1$, where $\mathcal{O}$ is the point at infinity.

The map h_d is well-defined since $h_d(u)$ is a point of $H_d(\mathbb{F}_q)$, for $u \in \mathbb{F}_q$. This encoding function for Hessian curves is less general than Icart's one, but has many other interesting properties. In fact, Farashahi (2011) showed that the size of the image set $h_d(\mathbb{F}_q)$ is at least $q/2$ an that the inverse function h_d^{-1} can be easily described. He also studied the possibility to extract random bits from the image $h_d(u)$ of an element $u \in \mathbb{F}_q$. .

3.1.3. *Encoding into Edwards Curves.*

- **Elligator 1: An injective encoding for Edwards curves** *(2013)* : Bernstein *et al.* (2013) defined an encoding that maps an element of $\mathbb{F}_q$ to a point of the Edwards curve $E_d: \ x^2+y^2 = 1+dx^2y^2$, with $d \notin \{0,1\}$ and d is not a square. Their work used the general method proposed by Fouque *et al.* (2013). Note that, in their updated paper, Fouque *et al.* (2013) have also proposed an explicit encoding for Edwards curves, based on their previous work of Fouque *et al.* (2013). The encoding process of Bernstein *et al.* (2013) is described as follows:

Let q be a prime power congruent to 3 modulo 4. Let s be a nonzero element of $\mathbb{F}_q$ with $(s^2-2)(s^2+2) \neq 0$. Define $c = 2/s^2$. Then $c(c-1)(c+1) \neq 0$. Define $r = c+1/c$ and $d = -(c+1)^2/(c-1)^2$. Then $r \neq 0$, and d is not a square. The following elements of $\mathbb{F}_q$ are defined for each $t \in \mathbb{F}_q \setminus \{0,1\}$:

$$\begin{aligned} u &= (1-t)/(1+t), \qquad v = u^5+(r^2-2)u^3+u;\\ X &= \chi(v)u, \qquad Y = (\chi(v)v)^{(q+1)/4}\chi(v)\chi(u^2+1/c^2);\\ x &= (c-1)sX(1+X)/Y, \qquad y = (rX-(1+X)^2)/(rX+(1+X)^2). \end{aligned}$$

Furthermore $x^2+y^2 = 1+dx^2y^2$; $uvXYx(y+1) \neq 0$ and $Y^2 = X^5+(r^2-2)X^3+X$. Hence, this algorithm describes an encoding from E_d to $\mathbb{F}_q$. The authors also showed that the image set of their encoding can be easily described. Furthermore, it becomes an injective encoding into E_d if one considers only half of the points of $\mathbb{F}_q$.

- **AIIE-For-Edwards** (*2017*) : Diarra *et al.* (2017) proposed an algorithm to encode directly an element of $\mathbb{F}_q$ (with $\mathrm{char}(\mathbb{F}_q) \neq 2$) to a point of an Edwards curves $E_d \ : \ x^2+y^2 = 1+dx^2y^2$ ($d \neq \pm 1, -2$ and d is not a square), by using Elligator-2's method of Bernstein *et al.* (2013) (and without any birational equivalence). To encode into E_d, one must first choose an element u which is not as square in $\mathbb{F}_q$, and define the set $\mathcal{R} = \{r \in \mathbb{F}_q^* : ur^2+1 \neq 0,\ ur^2(1-d)-(1+3d) \neq 0,\ ur^2(1+3d)-(1-d) \neq 0\}$.

The encoding is then given by the following algorithm:

```
Input: $r \in \mathcal{R}$;
Output: A point $(x, y) \in E_d : \; x^2 + y^2 = 1 + dx^2y^2$;
    1. $v = \frac{(d-1)ur^2 - 3 - d}{ur^2(d-1) + 1 + 3d}$        $\varepsilon = \chi[(1 - v^2)(1 - dv^2)]$
    2. $x = \frac{\varepsilon(v+1)(dv+1) + dv^2 - 1}{2d(v+1) + (1-d)}$        $y = -\varepsilon\sqrt{\frac{1-x^2}{1-dx^2}}$
    3. Return $(x, y) \in E_d$.
```

Algorithm 1: AIIE-For-Edwards

The authors showed that this encoding is almost-injective and invertible. They also gave some examples of SafeCurves (Bernstein and Lange (2014)) that are suitable for this encoding.

3.1.4. *Encoding into Huff curves.*

• **Encoding into generalized Huff curves: Brief SWU Encoding** (*2015*) : He *et al.* (2015) proposed two methods for encoding into generalized Huff curves given by $E : \; ax(y^2 - c) = by(x^2 - d)$, with $abcd(a^2c - b^2d) \neq 0$. The first one, that we describe here, is called Brief SWU encoding. This encoding is a mapping of the form $f = \psi \circ \phi$, where ϕ is an encoding from $\mathbb{F}_q$ to the Weierstrass curve $E' : \; y^2 = g(x) = x^3 + (b^2d + a^2c)x^2 + (a^2b^2cd)x$. As mentioned by the authors, the encoding ϕ is an adaptation of Ulas's encoding (based on SWU method [3.1.1]) for elliptic curves defined by $y^2 = x^n + Ax^2 + Bx$ over $\mathbb{F}_q$, with $q \equiv 3 \mod 4$. It works as follows, given $u \in \mathbb{F}_q$:

- if $u = 0$, return $(0, 0)$;
- else, compute $X(u) = \frac{a^2b^2cd(u^2 - 1)}{a^2c + b^2d}$ and $g(X(u))$;
- compute $Y(u) = -\frac{a^2b^2cd}{a^2c + b^2d}\left(1 - \frac{1}{u^2}\right)$ and $g(Y(u))$;
- return $(s, t) = (X(u), -\sqrt{g(X(u))})$ or $(s, t) = (Y(u), \sqrt{g(Y(u))})$ (from the Skalba's equality, it follows that either $g(X(u))$ or $g(Y(u))$ is a quadratic residue in $\mathbb{F}_q$).

The map ψ defines the birational equivalence from the above Weierstrass form to the generalized Huff curve :

$$\psi : \; E'(\mathbb{F}_q) \to E(\mathbb{F}_q), \; (s, t) \mapsto (x, y) = \left(\frac{bd(s + a^2c)}{t}, \frac{ac(s + b^2d)}{t}\right).$$

The encoding f covers all $\mathbb{F}_q$ and is well-distributed (in the sense of Farashahi *et al.* (2013)). The authors also gave an estimation size of the image set, using the Chebotarev density theorem (see He *et al.* (2015)) like Fouque and Tibouchi did in order to estimate the size of Icart's encoding.

• **Encoding into generalized Huff curves: Cube Root Encoding** (*2015*) : The second encoding for generalized Huff curves proposed by He *et al.* (2015) is the Cube root Encoding. As the name suggests, this encoding is based on computing cube roots in $\mathbb{F}_q$, where $q \equiv 2 \mod 3$. This condition on q allows them to efficiently compute cube roots, by $a^{1/3} = a^{(2q-1)/3}$. The encoding works as follows, given $u \in \mathbb{F}_q$:

- compute $t = u^2 - a^2c - b^2d$;
- compute $r = \frac{1}{2}(a^2b^2cd - \frac{1}{3}t^2)$.

This encoding is a weak encoding (in the sense of Brier *et al.* (2010)), which will allow to construct indifferentiable hash function into this Huff curve.

• **AIIE-For-Gen-Huff: Encoding into generalized Huff curves** (*2017*) :

Following the idea of Fouque *et al.* (2013) for constructing injective encoding, Diarra *et al.* (2017) proposed an explicit encoding for the generalized Huff model $E_{a,b}$: $x(ay^2 - 1) = y(bx^2 - 1)$ over $\mathbb{F}_q$, with $q \equiv 3 \mod 4$ and $ab(a - b) \neq 0$ (note that Bernstein *et al.* (2013) did a same construction for the Edwards model). We summarize here their encoding. To encode elements of $\mathbb{F}_q$ into points of $E_{a,b}$, one needs to choose an element $c \in \mathbb{F}_q$ such that $c(c-1)(c+1) \neq 0$, and compute $A = -(c - \frac{1}{c})^2$. Let $s = \dfrac{c+1}{c-1}$ and $\lambda \in \mathbb{F}_q^*$; define $b = \dfrac{\lambda^2}{1 - s^2}$ and $a = -bs^2$. Then the following algorithm shows how to encode any $z \in \mathbb{F}_q \setminus \{-1, 1\}$ into a point of $E_{a,b} \setminus (0,0)$ (and $1, -1$ are sent on the point $(0,0)$).

```
Input: $z \in \mathbb{F}_q \setminus \{-1,1\}$,
Output: A point $(x,y) \in \mathbf{E_{a,b}} : \mathbf{x(ay^2-1) = y(bx^2-1)}$;
    1. $u = (1-z)/(1+z)$,    $v = -u^5 + (c^2 + 1/c^2)u^3 - u$
    2. $X = \chi(v)u$,    $Y = \chi(v \cdot (c^2u^2 - 1))\sqrt{\chi(v)v}$
    3. $\alpha = \frac{\lambda}{2} = \frac{\sqrt{a+b}}{2}$,    $x = \frac{(1+X)(bX-\alpha^2(1+X)^2)}{b\alpha Y}$,    $y = \frac{(1+X)(aX-\alpha^2(1+X)^2)}{a\alpha Y}$
    4. Return $(x,y) \in E_{a,b}$
```

Algorithm 2: AIIE-For-Gen-Huff

This algorithm, in addition to encode into the Huff curve, also provides a way to encode on the Hyperelliptic curve $Y^2 = -X^5 + (2-A)X^3 - X$ (one can verify that (X,Y) is indeed a point of this curve).

• **AIIE-For-Class-Huff: Encoding into classical Huff curves** (*2017*) ; Diarra *et al.* (2017) also proposed an encoding for classical Huff curves, which are curves of the form $\alpha x(y^2-1) = \beta y(x^2-1)$ with $\alpha\beta(\alpha^2-\beta^2)(\alpha^2+\beta^2) \neq 0$. One can remark that such curves are a particular class of the generalized model $x(ay^2-1) = y(bx^2-1)$ (it suffices to set $a = \alpha^2$, $b = \beta^2$ and use the change of variables $(x,y) \to (x' = \beta x, y' = \alpha y)$). But, as the authors noticed, their method for encoding into the generalized model could not work directly for the classical model (since the product ab has to be a non-quadratic residue in $\mathbb{F}_q$). Hence, they suggested another method, which is based on Elligator-2 by Bernstein *et al.*, to encode directly (that is without any birational equivalence) into the classical model. Recall that Elligator-2 defines an injective encoding for Weierstrass curves of the form $y^2 = x^3 + Ax^2 + Bx$. With this method, to encode into another model of curves, it would be necessary to use a birational equivalence.

3.1.5. *Encoding into Montgomery curves (2013) and Jacobi Quartic Curves.* **(2015)**

• **Montgomery curves :** Yu *et al.* (2013) have proposed four deterministic encoding in the families of Montgomery curves $E_{a,b} : by^2 = x^3 + ax^2 + x$ defined over $\mathbb{F}_q$ where q is an odd power of prime. One is based on finding a cube root, whereas the other three are based on finding square roots.

• **Jacobi Quartic Curves :** Yu et al. (2015) have proposed two deterministic encoding from a finite field $\mathbb{F}_q$ into Jacobi quartic curves $E_b : y^2 = x^4 + 2bx^2 + 1$. When $q \equiv 3(\mod 4)$, their first deterministic encoding based on Skalba's equality saves two field squaring compared with birational equivalence composed with Fouque and Tibouchi's brief version of Ulas' function. When $q \equiv 2(\mod 3)$, their second deterministic encoding based on computing cube root costs one field inversion less than birational equivalence composed with Icart's function at the cost of four field multiplications and one field squaring.

3.1.6. *Encoding into Jacobi Intersection Curves.*

He *et al* (2017) proposed two encoding into families of Twisted Jacobi Intersection Curves $E_{a,b} : au^2 + v^2 = bv^2 + w^2 = 1$. Their first encoding is based to the SWU encoding and the second is the cube root encoding. Moreover, they have estimates the density of images of both encoding by Chebotarev theorem He *et al* (2017). We recall their cube root encoding.

Suppose $q = p^n \equiv 2 \mod 3$ is a power of odd prime number. Their cube root encoding $f_2 : \mathbb{F}_q \to E_{a,b} : t \mapsto (u, v, w)$ is defined as follows.

```
**Input :** $a, b$ and $t \in: \mathbb{F}_q$ with $ab(a-b) \neq 0$.
**Output :** A point $(u, v, w) \in E_{a,b}(\mathbb{F}_q)$
    (1) If $t = 0$ return $(u, v, w) = (0, 1, 1)$ directly;
    (2) Else put $\alpha = \frac{a+b+t^2}{3}$, $\beta = \frac{ab-3\alpha^2}{2}$, $\gamma = \alpha t + (t\beta^2 - (\alpha t)^3)^{1/3}$ and return
        $(u, v, w) = \frac{2t}{\gamma^2 - abt^2}(t\gamma + \beta, a(\gamma - bt), b(\gamma - at)$
            Note that since $q \equiv 2 \mod 3$, one can efficiently compute the cube root by
        $x^{1/3} = x^{\frac{2q-1}{3}}$.
```

Algorithm 3: Encoding Jacobi Intersection Curves (2018)

3.2. Hyperelliptic Curves encoding.

Hyperelliptic curves are generalizations of elliptic curves in higher genus (an elliptic curve has genus one).

• **Ulas encoding** (*2007*) : Ulas (2007) was the first to propose encoding for some hyperellitpic curves. He simplified the results (namely the rational maps) of Shalue and Woestjine (see section 3.1.1) and generalized them to obtain encoding for the two families of Hyperelliptic curves $y^2 = x^n + ax + b$ and $y^2 = x^n + ax^2 + bx,\ n \geq 3, q \equiv 2 \mod 3$, where $n \geq 5$ is a fixed positive integer.

• **KLR-Genus-Two: Kammerer-Lercier-Renault's encoding** (*2010*) : Kammerer *et al.* (2010) proposed encoding for various models of (hyper)elliptic curves, including for example the Hessian model $x^3 + y^3 + 1 = 3d$ (see previous section on elliptic curves encoding), the genus two curves $y^2 = (x^3 + 3ax + 2)^2 + 8b^3$ (type A) and $y^2 = \lambda((x^3 + 3\mu x + 2a)^2 + 4b)$ (type B), over $\mathbb{F}_q$ (with $q \equiv 2 \mod 3$). We refer the reader to Kammerer *et al.* (2010) for the precise description of these encoding, since the formulas are rather complicated.

•**Unified encoding of Seck *et al.***(*2018*) : Following the technique of Elligator-2 in Bernstein *et al.* (2013), Seck *et al.* (2017) proposed three deterministic *almost*-injective encoding for three families of Hyperelliptic curves given by the equations $\mathbb{H}^i : \;\; y^2 = F_i(x) \quad (i = 1, 2, 3)$, where $F_1(x) = x^5 + ax^4 + cx^2 + dx, \; F_2(x) = x^5 + bx^3 + dx + e, \; F_3(x) = x^5 + ax^4 + e$ over $\mathbb{F}_q$. Like many existing encondigs (Elligator-2, Encoding into Classical Huff, etc.), their encoding $\psi_i : \; \mathcal{R}_i \to \mathbb{H}^i$ ($\mathcal{R}$ is a certain subset of $\mathbb{F}_q$) have characterizable image set an can be extended to $\mathbb{F}_q$. These works were recently unified and generalized by Seck and Diarra (2018). In fact, the authors found unified formulas that work given any non-binary (hyper)elliptic curve of the form $y^2 = h_g(x) = x^{2g+1} + a_{2g-1}x^{2g-1} + a_{2g-}x^{2g-3} + \ldots + a_1x + a_0$, where the genus g of the curve verifies $g \leq 5$. We recall here their main result.

Consider the Hyperelliptic curve $\mathbb{H}^g$ of genus g, given by

$$y^2 = F_g(x) = x^{(2g+1)} + a_{(2g-1)}x^{(2g-1)} + a_{(2g-3)}x^{(2g-3)} + \ldots + a_1x + a_0 \tag{3.2}$$

over $\mathbb{F}_q$ ($q = p^n, \; p \neq 2$ and $p \nmid 2g^2 + g$).
We refer to Seck and Diarra (2018) for the explicit values of the coefficients a_i. Let $\alpha_g = 2^{2g-1} - 1, \, \beta_g = 4g^2 + 2g$ and define m_g and n_g as follows:

$$\begin{cases} m_g = \dfrac{\alpha_g \beta_g}{2}, \;\; n_g = (2g^2 + g)^2 & \text{if } g \text{ is odd,} \\ m_g = \dfrac{\alpha_g \beta_g}{4}, \;\; n_g = \dfrac{(2g^2 + g)^2}{2} & \text{if } g \text{ is even.} \end{cases}$$

Then let $\mathcal{R}_g = \{r \in \mathbb{F}_q^* : \ F_g(w(ur^2(-m_g s - n_g) - 1)) \neq 0\}$. The encoding into the Hyperelliptic curve $\mathbb{H}^g$ is given by the following algorithm.

```
**Input:** The Hyperelliptic curve $\mathbb{H}_g$, and $r \in \mathcal{R}_g$;
**Output:** A point $(x, y)$ on $\mathbb{H}_g$;
    **1.** $v := v(g) = w[ur^2(-m_g s - n_g) - 1]$;
    **2.** $\varepsilon := \chi_q(v^{(2g+1)} + a_{(2g-1)}v^{(2g-1)} + a_{(2g-3)}v^{(2g-3)} + \ldots + a_1 v + a_0)$;
    **3.** $x := \frac{1+\varepsilon}{2}v + \frac{1-\varepsilon}{2}\left(\frac{w(-v+w)}{v+w}\right)$;
    **4.** $y := -\varepsilon\sqrt{x^{(2g+1)} + a_{(2g-1)}x^{(2g-1)} + a_{(2g-3)}x^{(2g-3)} + \ldots + a_1 x + a_0}$;
    **5.** **Return** $(x, y) \in \mathbb{H}_g$.
```

Algorithm 4: Genus-g-Encoding

Like all existing Elligator-like functions, this generalized encoding is almost-injective and efficienty invertible. We will use it in the last section to define a (generalized) indifferentiable hash function for any Hyperelliptic curve $\mathbb{H}^g$ given by equation (3.2).

3.3. Summary and Comparative Analyses.

From the above sections, it appears that one can classify almost all existing encoding into 4 types (following the construction used to defined the encoding): Icart-like encoding, Elligator 2-like encoding, SWU-like encoding and injective-like encoding. We then obtain the following table.

Table 1. Classification of encoding into Elliptic and Hyperelliptic Curves

Encoding	**Curve**	$\mathbb{F}_q$	**SWU**	**Icart**	**Injective**	**Elligator-2**
			-like	**-like**	**-like**	**-like**
Brier *et al.* (2010)	$y^2 = x^3 + a + b$	$q \equiv 2 \mod 3$	✓			
Fouque and Tibouchi (2012)	$y^2 = x^3 + b$	$q \equiv 1 \mod 3$	✓			
Farashahi (2011)	$x^3 + y^3 + 1 = 3dxy$	$q \equiv 2 \mod 3$		✓		
Kammerer *et al.* (2010)	$x^3 + y^3 + 1 = 3dxy$	$q \equiv 2 \mod 3$		✓		
Bernstein *et al.* (2013)	$x^2 + y^2 = 1 + d^2y^2$	$q \equiv 3 \mod 4$			✓	
Diarra *et al.* (2017)	$x^2 + y^2 = 1 + dx^2y^2$	$q \neq 2^n$				✓
He *et al.* (2015)	$ax(y^2 - c) = by(x^2 - d)$	$q \equiv 3 \mod 4$	✓			
	$ax(y^2 - c) = by(x^2 - d)$	$q \equiv 2 \mod 3$		✓		
Diarra *et al.* (2017)	$x(ay^2 - 1) = y(bx^2 - 1)$	$q \equiv 3 \mod 4$			✓	
	$\alpha x(y^2 - 1) = \beta y(x^2 - 1)$	$q \neq 2^n$				✓
Yu *et al.* (2013)	$by^2 = x^3 + ax^2 + x$	$q \equiv 2 \mod 3$		✓		
		$q \equiv 3 \mod 4$	✓			
Yu et al. (2015)	$y^2 = x^4 + 2bx^2 + 1$	$q \equiv 2 \mod 3$		✓		
		$q \equiv 3 \mod 4$	✓			
He *et al* (2017)	$au^2 + v^2 = bv^2 + w^2 = 1$	$q \equiv 2 \mod 3$		✓		
		$q \equiv 3 \mod 4$	✓			
Ulas (2007)	$y^2 = x^n + ax + b$		✓			
	$y^2 = x^n + ax^2 + bx$					
Kammerer *et al.* (2010)	$y^2 = (x^3 + 3ax + 2)^2 + 8b^3$	$q \equiv 2 \mod 3$		✓		
	$y^2 = \lambda((x^3 + 3\mu x + 2a)^2 + 4b)$	$q \equiv 2 \mod 3$		✓		
Seck and Diarra (2018)	$y^2 = F_g(x),\ g \leq 5$	$q \equiv 7 \mod 8$				✓

3.4. encoding into (Hyper)Elliptic Curves.

DEFINITION 12. *Given an Hyperelliptic curve $\mathbb{H}$ over $\mathbb{F}_q$, an encoding into $\mathbb{H}$ is a function $f : \mathbb{F}_q \to \mathbb{H}$.*

encoding used in (Hyper)Elliptic Curve Cryptography can verify some interesting properties, like *admissibility* or *almost-injectivity*. We give here more formal definitions (that can be found in Farashahi *et al.* (2013); Bernstein *et al.* (2013); Diarra *et al.* (2017); Seck *et al.* (2017)) of such properties.

DEFINITION 13 (Almost-injective encoding).
An encoding $f : \mathbb{F}_q \to \mathbb{H}$ is *almost-injective* if there exists $S \subset \mathbb{F}_q$ such that: **(i)** $S \cap (-S) = \{0\}$; **(ii)** $\phi_i(r) = \phi_i(-r), \ \forall r \in \mathbb{F}_q$; **(iii)** $\#\phi_i^{-1}(\phi_i(r)) = 2, \ \forall r \in \mathbb{F}_q^*$. Thus f is almost-injective if its restriction to a certain subset S of $\mathbb{F}_q$ is injective.

This notion was first used by Bernstein *et al.* (2013) for their encoding (Elligator 1 and Elligator 2).

DEFINITION 14 (Admissible encoding, Brier *et al.* (2010), Farashahi *et al.* (2013)).

An encoding $f : \ \mathbb{F}_q \to \mathbb{H}$ is ϵ-*admissible* if it satisfies the following properties:

(i) computable: f is computable in deterministic polynomial time,

(ii) regular: for r uniformly distributed in $\mathbb{F}_q$, the distribution of $f(r)$ is ϵ-statistically indistinguishable from the uniform distribution in $\mathbb{H}$

and

(iii) samplable: there is an efficient randomized algorithm $I : \ \mathbb{H} \to \mathbb{F}_q$ such that for any $P \in \mathbb{H}$, $I(P)$ induces a distribution that is ϵ-statistically indistinguishable from the uniform distribution in $f^{-1}(P)$.

And f is *admissible* if ϵ is a negligible function of the security parameter.

DEFINITION 15 (Weak encoding, Brier *et al.* (2010), Farashahi *et al.* (2013)).
An encoding $f : \ \mathbb{F}_q \to \mathbb{H}$ is said to be an α-*weak encoding* if it satisfies:
(i) computable: f is computable in deterministic polynomial time;

(ii) α-bounded: for r uniformly distributed in $\mathbb{F}_q$, the distribution of $f(r)$ is α-bounded in $\mathbb{H}$, i.e. the inequality $Pr[f(r) = P] \leq \frac{\alpha}{\#R}$ holds for any $P \in \mathbb{H}$;
(iii) samplable: there is an efficient randomized algorithm I such that $I(P)$ induces the uniform distribution in $f^{-1}(P)$ for any $P \in \mathbb{H}$. Additionally $I(P)$ returns $N_P = \#f^{-1}(P)$.

And f is a *weak encoding* if α is a polynomial function of the security parameter

Weak encoding are essentially those whose image set size is a positive constant fraction of the size of the curve, which includes all known (hyper)elliptic curve encoding .

Definition 16 (Well-distributed encoding, Brier *et al.* (2010), Farashahi *et al.* (2013)).

Let C be a smooth projective curve over a finite field $\mathbb{F}_q$, Jac its Jacobian, f a function $\mathbb{F}_q \to C(\mathbb{F}_q)$ and B a positive constant. We say that f is B-well-distributed if for any nontrivial character χ of $Jac(\mathbb{F}_q)$, the following holds:

$$|S_f(\chi)| \leq B\sqrt{q},\ where\ S_f(\chi) = \sum_{u \in \mathbb{F}_q} \chi(f(u))$$

Then f is well-distributed if it is B-well-distributed for some B bounded independently of the security parameter.

Consider an encoding f into a curve C, and let Jac denote the Jacobian of C. We define the function $f^{\otimes s} : (\mathbb{F}_q)^s \to Jac(\mathbb{F}_q)$ as follows:

$$f^{\otimes s}(u_1, u_2, \ldots, u_s) = f(u_1) + f(u_2) + \ldots + f(u_s), \forall (u_1, u_2, \ldots, u_s) \in (\mathbb{F}_q)^s$$

Theorem 88. *Farashahi et al. (2013)* Let $h : \tilde{C} \to C$ be a non constant morphism of curves, and χ be any nontrivial character of $Jac(\mathbb{F}_q)$, where Jac is the Jacobian of C. Assume that h does not factor through a nontrivial unramified morphism $Z \to X$. Then:

$$\left| \sum_{P \in \tilde{C}(\mathbb{F}_q)} \chi(h(P)) \right| \leq (2\tilde{g} - 2)\sqrt{q},$$

where $\tilde{g}$ is the genus of $\tilde{C}$.

Furthermore, if q is odd and ϕ is a non constant rational function on $\tilde{C}$:

$$\left| \sum_{P \in \tilde{C}(\mathbb{F}_q)} \chi(h(P)) \left(\frac{\phi(P)}{q} \right) \right| \leq (2\tilde{g} - 2 + \deg(\phi))\sqrt{q} \tag{3.3}$$

where $\left(\frac{\div}{q}\right)$ denotes Legendre symbol.

4. Overview of Hashing into (Hyper)Elliptic Curves

By a hash function into an (Hyper)elliptic Curve, we mean a function $H : \{0,1\}^* \to \mathbb{J}(\mathbb{F}_q)$, that sends a string of bits to an element of the Jacobian $\mathbb{J}(\mathbb{F}_q)$ of the curve. But why to define such functions? Simply because many elliptic curve-based schemes require to hash into the group of points of an elliptic curve. For example, Boneh and Franklin (2001) were one of the first who required hashing into elliptic curves; in fact, the public key of their identity-based encryption scheme is a point on the curve $y^2 = x^3 + b$ (which is known to be supersingular). There are other examples of schemes using hash functions into elliptic curves, such as BLS signatures Boneh *et al.* (2001), passwords-based authentication protocols (SPEKE Jablon (1996), PAK Boyko *et al.* (2000)), etc.

In this section, we will give an overview of all existing methods to construct hash functions into elliptic curves, going from the simplest one (which of course does not meet all ”cryptographic” requirements of hash functions) to some more complex constructions. And we will se how these hash functions are closely related to the encoding functions that we have seen in the previous sections.

4.1. Try-and-increment Method (*2001*).

This method of hashing requires both trivial encoding approach (3.1.1) and the use of a ”classical” cryptographic hash function. It was proposed by Boneh *et al.* (2001). In fact, their signature scheme (GDH) required a hash function $H : \{0,1\}^* \to G$ that sends a message $M \in \{0,1\}^*$ to a point of an elliptic curve defined by $y^2 = f(x)$. They first supposed the existence of $h : \{0,1\}^* \to \mathbb{F}_q \times \{0,1\}$, a classical cryptographic hash function. And to hash a message $m \in \{0,1\}^*$, they proceeded as follows:

1. set a counter $c = 0$;
2. compute $h(c||m) = (x, b)$;
3. compute $f(x)$
4. if $f(x)$ is a square, then return $H(m) = (x, (f(x))^{1/2})$ (the bit b allows to choose one of the two square roots of $f(x)$); otherwise, increment c and go to step 2.

This method of hashing has the drawback to not running in constant time: in fact, the running time depends on the message m to hash. This could lead to side-channel attacks.

4.2. The Construction $H = f \circ h$.

Brier *et al.* (2010) generalized and formalized the notion of *admissible encoding* by Boneh and Franklin (2001) and showed that the construction $H = f \circ h$ is indifferentiable from a random oracle, where f is an admissible encoding and h is a (classical) hash function modeled as random oracle. But most of the known deterministic encoding are not admissible (being admissible includes to be a $r : 1$ function) since their image set is only a fraction of the curve (about $5/8$ for Icart's encoding for example). Hence, this construction can not replace a random oracle in the group of points of the curve, unless for some particular cases (like the encoding to a supersingular curve used by Boneh and Franklin (2001) in their IBE scheme, which is a bijection from $\mathbb{F}_q$ to the group of rational points of the curve,).

4.3. The Construction $H = f \circ h_1 + h_2\mathbb{G}$ [Brier *et al.* (*2010*)].

The previous construction of indifferentaible hash function (using admissible encoding) does not work for most of existing deterministic encoding, since they can not be admissible. Hence the authors proposed another method of hashing, wich uses the notion of weak encoding (which is lighter than admissibility notion, and thus covers more encoding). Weak encoding can be seen as an encoding such that...

Consider an elliptic curve E over $\mathbb{F}_q$ such that $E(\mathbb{F}_q)$ is cyclic of order n, and a weak encoding $f : \mathbb{F}_q \to E(\mathbb{F}_q)$ into the curve. Let $h_1 : \{0,1\}^* \to \mathbb{F}_q$ and $h_2 : \{0,1\}^* \to \mathbb{Z}/n\mathbb{Z}$ two hash functions seen as random oracles. Then the construction $H(m) = f(h_1(m)) + h_2(m)G$ is indifferientable from a random oracle into the curve. This construction is more general than the previous one but it is less efficient since it requires scalar multiplication on the curve.

4.4. The Construction $H = f \circ h_1 + f \circ h_2$.

Brier *et al.* (2010), using the notion of *admissible encoding*, showed that the construction $H(m) = f(h_1(m)) + f(h_2(m))$ is indifferentiable from a random oracle, where f is the Icart's encoding 3.1.1 and h_1, h_2 are two (classical) hash functions modeled as random oracles. The proof involves well-distributed encoding and character sums.

4.5. Generalization of $H = f \circ h_1 + f \circ h_2$.

Farashahi *et al.* (2013) generalized the construction of Brier *et al.* to all known deterministic (and well-distributed) encoding on elliptic and Hyperelliptic curves. In fact, they proposed the construction $H(m) = f(h_1(m)) + \ldots + f(h_s(m))$, which preserves the indifferentiability if the h_i are modeled as random oracles, f is a well-distributed encoding on the curve and s is an integer strictly greater than the genus of the curve. This allows to hash into any Hyperelliptic curve where a well-distributed encoding is known. Many works used later this construction to construct hash functions into elliptic curves, like in Diarra *et al.* (2017); Seck *et al.* (2017); He *et al* (2017); Yu et al. (2015).

After listing all existing methods of hashing, we are now able to give the following classification of hash functions into (hyper)Elliptic curves, as we did to classify encoding into (Hyper)elliptic curves.

TABLE 2.EXISTING INDIFFERENTIABLE HASH FUNCTIONS INTO (HYPER)ELLIPTIC CURVES. Existing indifferentiable hash functions into (hyper)elliptic curves

Hashing into	**Method of Hashing**	**Encoding used**
$E_b: \ y^2 = x^3 + b$	$H = f \circ h$	Boneh and Franklin (2001)
$E_b: \ y^2 = x^3 + b$	$H = f \circ h_1 + \ldots + f \circ h_s$	Fouque and Tibouchi (2012)
$E_d: \ x^2 + y^2 = 1 + dx^2y^2$	$H = f \circ h_1 + \ldots + f \circ h_s$	Diarra *et al.* (2017)
$E_{a,b} : y^2 = x^3 + ax + b$	$H = f \circ h_1 + h_2\mathbb{G}$ $H = f \circ h_1 + h_2\mathbb{G}$	Brier *et al.* (2010)
$E_{a,b} = y^2 = x^3 + ax + b$	$H = f \circ h_1 + \ldots + f \circ h_s$	Farashahi *et al.* (2013)
$H = x^3 + (y + c)(3x + 2a + \frac{2b}{y}) = 0$	$H = f \circ h_1 + \ldots + f \circ h_s$	Farashahi *et al.* (2013)
$E_1 : \ ax(y^2 - c) = by(x^2 - d)$	$H = f \circ h_1 + h_2\mathbb{G}$	He *et al.* (2015)
$E_b = y^2 = x^4 + 2bx^2 + 1$	$H = f \circ h_1 + \ldots + f \circ h_s$	Yu et al. (2015)
$E_{a,b} : by^2 = x^3 + ax^2 + x$	$H = f \circ h_1 + h_2\mathbb{G}$ $H = f \circ h_1 + \ldots + f \circ h_s$	Yu *et al.* (2013)
$E_{a,b} : au^2 + v^2 = bv^2 + w^2 = 1$ $\mathbb{H}^1 : y^2 = x^5 + ax^4 + cx^2 + dx$ $\mathbb{H}^2 : y^2 = x^5 + a_3x^3 + a_1x + a_0$	$H = f \circ h_1 + \ldots + f \circ h_s$ $H = f \circ h_1 + \ldots + f \circ h_s$	He *et al* (2017) Seck *et al.* (2017)
$\mathbb{H}_g : y^2 = x^{2g+1} + a_{2g-1}x^{2g-1} + \ldots + a_1x + a_0$	$H = f \circ h_1 + \ldots + f \circ h_s$	Seck and Diarra (2018)

5. Unified Formulas for Hashing into (Hyper)Elliptic Curves

This part uses the generalized encoding of Seck and Diarra in AFRICACRYPT 2018. Our main goal, in this section is to show that one can design an indifferentiable hash function into the Jacobian of the hyperelliptic curve $\mathbb{H}_g : y^2 = F_g(x) = x^{2g+1} + a_{2g-1}x^{2g-1} + a_{2g-}x^{2g-3} + \ldots + a_1x + a_0$, $g \leq 5$ using the unified encoding ψ_g of Seck and Diarra (2018) and the framework of Farashahi *et al.* (2013).

Let as recall some results of Seck and Diarra. Let us define

- $\alpha_g = 2^{2g-1} - 1$, $\beta_g = 4g^2 + 2g$;
- $m_g = \dfrac{\alpha_g \beta_g}{2}$, $n_g = (2g^2 + g)^2$ if g is odd;

  and $m_g = \dfrac{\alpha_g \beta_g}{4}$, $n_g = \dfrac{(2g^2 + g)^2}{2}$ if g is even;
- $\mathcal{R}_g = \{r \in \mathbb{F}_q^* : \ F_g(w(ur^2(-m_g s - n_g) - 1)) \neq 0\}$ and $S_g = \mathbb{F}_q \setminus \mathcal{R}_g$ where $w \in \mathbb{F}_q^\star$ is an arbitrary parameter, $u \in \mathbb{F}_q$ is a nonzero non-square

parameter and $s \in \mathbb{F}_q$ satisfies $\alpha_g s^2 + \beta_g s - \gamma_g = 0$. The set S_g contains at most $2(2g+1)+1$ elements.

The unified encoding is defined as follows : $\psi_g : \mathcal{R}_g \to \mathbb{H}_g : r \mapsto \psi_g(r) = (x, y)$ where

$$x = \frac{1+\varepsilon}{2}v + \frac{1-\varepsilon}{2}\left(\frac{w(-v+w)}{v+w}\right) \textit{ and } y = -\varepsilon\sqrt{F_g(x)},$$

with

$v = w[ur^2(-m_g s - n_g) - 1]$ *and* $\varepsilon = \chi_q(v^{(2g+1)} + a_{(2g-1)}v^{(2g-1)} + a_{(2g-3)}v^{(2g-3)} + \ldots + a_1 v + a_0)$.

Let us extend the encoding ψ_g to $\mathbb{F}_q$ as follows. We choose $w = z^2, z \in \mathbb{F}_q^*$, then $F_g(w)$ is a nonzero square and $(-w, -\sqrt{F_g(w)}) \in \mathbb{H}_g \setminus \mathrm{Im}(\psi_g)$. We put for all $t \in S_g$, $\psi_g(\pm t) = (-w, -\sqrt{F_g(w)})$. We have

THEOREM 89 (Seck and Diarra (2018)). *Suppose that*

(1) Let (x, y) *be a point of the hyperelliptic curve* $\mathbb{H}_g$*, then* $(x, y) \in \mathrm{Im}(\psi_g)$ *if and only if* $uw(x+w)(-n_g - m_g s)$ *is a nonzero square in* $\mathbb{F}_q$

and

(2) Let $(x, y) \in \mathrm{Im}(\psi_g)$ *and define* $\bar{r}$ *as follows:*

$$\bar{r} = \begin{cases} \sqrt{\frac{x+w}{uw(-n_g - m_g s)}} & \textit{if } y \notin \sqrt{\mathbb{F}_q^2} \\ \sqrt{\frac{x+w}{uw(-n_g - m_g s)}} & \textit{Otherwise.} \end{cases} .$$

Then $\bar{r} \in \mathcal{R}_g$ *and* $\psi_g(\bar{r}) = (x, y)$*.*

By the previous theorem, we know that

$$\chi_q(y) = -1 \Longleftrightarrow y \notin \sqrt{\mathbb{F}_q^2} \Longleftrightarrow f_1(x, r) = uw(-n_g - m_g s)r^2 - (x+w) = 0,$$

and

$$\chi_q(y) = 1 \Leftrightarrow y \in \sqrt{\mathbb{F}_q^2} \Leftrightarrow f_2(x,r) = u(x+w)(-n_g - m_g s)r^2 - 2w = 0$$

Let us define the coverings $h_j : C_j \to \mathbb{H}_g, j = 1,2$, by the smooth projective curves whose function fields are the extensions (denoted by $\mathbb{F}_q(x,y,r)$) of $\mathbb{F}_q(x,y)$ defined by $uw(-n_g - m_g s)r^2 - (x+w) = 0$ and $u(x+w)(-n_g - m_g s)r^2 - 2w = 0$ respectively. In other words, a rational point in $C_j(\mathbb{F}_q)$ is a tuple (x,y,r) such that $(x,y) \in \mathbb{H}_g$ and $f_j(x,r) = 0$. In particular, for any $r \in \mathcal{R}_g$, there are two rational points of $\mathbb{H}_g$ whose third coordinate is r which are (x,y,r) and $(x,-y,r)$. Using this result, one can define morphisms $g_j : C_j \to \mathbb{P}^1$, such that any point in $\mathbb{A}^1(\mathbb{F}_q) \setminus S_g$ has exactly two preimages in $C_j(\mathbb{F}_q)$ for one of $j = 1,2$, and none in the other. These two preimages are conjugate under $y \mapsto -y$, so that exactly one of them satisfies $\chi_q(y) = \left(\frac{y}{q}\right) = (-1)^q$ since $q \equiv 7 \mod 8$ and then $q \equiv 3 \mod 4$. If we denote by $Q \in C_j(\mathbb{F}_q)$ that preimage, then $\psi_g(r) = h_j(Q)$. We have

THEOREM 90. *For any nontrivial character χ of $\mathbb{H}_g(\mathbb{F}_q)$, the character sum $S_{\psi_g}(\chi)$ satisfies:*

$$\left|S_{\psi_g}(\chi)\right| \leq (16g)\sqrt{q} + (44g + 31)$$

where g is the genus of $\mathbb{H}_g$.

Proof. We know that

$$\begin{aligned}\left|S_{\psi_g}(\chi)\right| = \left|\sum_{r\in\mathbb{F}_q} \chi(\psi_g(r))\right| &= \left|\sum_{r\in\mathcal{R}_g} \chi(\psi_g(r)) + \sum_{r\in S_g} \chi(\psi_g(r))\right| \\ &\leq \left|\sum_{r\in\mathcal{R}_g} \chi(\psi_g(r))\right| + \#S_g,\end{aligned}$$

where $\#A$ denote the cardinal of A. We have

$$\sum_{r\in\mathcal{R}_g} \chi(\psi_g(r)) = \sum_{\substack{Q\in C_1(\mathbb{F}_q)\setminus S_{g,1} \\ \chi_q(y)=-1}} \chi(h_1(Q)) + \sum_{\substack{Q\in C_2(\mathbb{F}_q)\setminus S_{g,2} \\ \chi_q(y)=1}} \chi(h_2(Q)),$$

here $S_{g,j} = g_j^{-1}(S_g \cup \{\infty\})$. Thus

$$\left|\sum_{r\in\mathcal{R}_g}\chi(\psi_g(r))\right| \le \left|\sum_{\substack{Q\in C_1(\mathbb{F}_q)\\ \chi_q(y)=-1}}\chi(h_1(Q))\right| + \left|\sum_{\substack{Q\in C_2(\mathbb{F}_q)\\ \chi_q(y)=1}}\chi(h_2(Q))\right| + \#S_{g,1} + \#S_{g,2}.$$

To estimate each sum in the right-hand side of the previous inequality, we will use the following equality

$$\sum_{Q\in C_j(\mathbb{F}_q)}\chi(h_j(Q)).\left(\frac{(1+(-1)^j\chi_q(y)}{2}\right) = \sum_{\substack{Q\in C_j(\mathbb{F}_q)\\ \chi_q(y)=(-1)^j}}\chi(h_j(Q)) + \frac{1}{2}\sum_{\substack{Q\in C_j(\mathbb{F}_q)\\ \chi_q(y)=0}}\chi(h_j(Q)).$$

We deduce that

$$\sum_{\substack{Q\in C_j(\mathbb{F}_q)\\ \chi_q(y)=(-1)^j}}\chi(h_j(Q)) = \sum_{Q\in C_j(\mathbb{F}_q)}\chi(h_j(Q)).\left(\frac{(1+(-1)^j\chi_q(y)}{2}\right) - \frac{1}{2}\sum_{\substack{Q\in C_j(\mathbb{F}_q)\\ \chi_q(y)=0}}\chi(h_j(Q))$$

On the one hand, the sum $\sum_{\substack{Q\in C_j(\mathbb{F}_q)\\ \chi_q(y)=0}}\chi(h_j(Q))$ is bounded by $2\times(2g+1) = 4g+2$ where g is the genus of the hyperelliptic curve $\mathbb{H}_g$.

In the other hand, by the Eisenstein criterion $h_j, j = 1, 2$ are totally ramified over points in $\mathbb{H}_g$ such that $x = -w$. So they cannot factor through any unramified covering of $\mathbb{H}_g$. Applying the Inequality 3.3 in the Theorem 88, we have

$$\left|\sum_{Q\in C_j(\mathbb{F}_q)}\chi(h_j(Q)).\left(\frac{(1+(-1)^j\chi_q(y)}{2}\right)\right| \le (2g_{C_j} - 2 - \deg y)\sqrt{q}$$

where g_{C_j} is the genus of C_j and $\deg y$ is the degree of y as a rational function on C_j. Since $[\mathbb{F}_q(x,y,r) : \mathbb{F}_q(x,y)] = 2$ and $[\mathbb{F}_q(x,y) : \mathbb{F}_q(y)] = 2g+1$, then $\deg y = [\mathbb{F}_q(x,y,r) : \mathbb{F}_q(y)] = 2\times(2g+1) = 4g+2$.

Now, we are going to compute the genus g_{C_j} of the curve C_j. The covering $h_j : C_j \to \mathbb{H}_g$ is only ramified at points with $x = -w$. Therefore by the Riemann-Hurwitz formula, we get

$$2g_{C_j} - 2 = 2(2g-2) + 2(2-1) = 4g-2 \Longleftrightarrow g_{C_j} = 2g.$$

Finally

$$\left| \sum_{Q \in C_j(\mathbb{F}_q)} \chi(h_j(Q)). \left(\frac{(1+(-1)^j \chi_q(y)}{2} \right) \right| \leq (8g)\sqrt{q}.$$

And then

$$\left| \sum_{\substack{Q \in C_j(\mathbb{F}_q) \\ \chi_q(y)=(-1)^j}} \chi(h_j(Q)) \right| \leq (8g)\sqrt{q} + (2g+1).$$

We deduce that

$$\begin{aligned} \left| \sum_{r \in \mathcal{R}_g} \chi(\psi_g(r)) \right| \leq \quad & \left((8g)\sqrt{q} + (2g+1) + \#S_{g,1} + \frac{1}{2}\#S_g \right) + \\ & \left((8g)\sqrt{q} + (2g+1) + \#S_{g,2} + \frac{1}{2}\#S_g \right) \\ & \leq (16g)\sqrt{q} + (4g+2) + \#S_{g,1} + \#S_{g,2} + \#S_g \end{aligned}$$

We know that $\#S_g \leq 8g+5$ and $S_{g,j} \leq 2(\#S_g + 1) \leq 16g+12$ since g_j is a map of degree 2. Thus

$$\left| \sum_{r \in \mathcal{R}_g} \chi(\psi_g(r)) \right| \leq (16g)\sqrt{q} + (4g+2) + 32g + 24 + 8g + 5 = (8g+16)\sqrt{q} + (44g+29)$$

We get finally $\left|S_{\psi_g}(\chi)\right| \leq 16g\sqrt{q} + (44g+31)$. In particular, we have

- $|S_{\psi_2}(\chi)| \leq 32\sqrt{q}+119$ if $g=2$ (it is the bound obtained by Seck *et al.* (2017))

and

- if $g = 3$ then $|S_{\psi_3}(\chi)| \leq 48\sqrt{q} + 163$.

We can conclude that the encoding ψ_g is well-distributed using the Theorem 16. And consequently the hash function $H(m) = \psi_g(h_1(m)) + \ldots + \psi_g(h_s(m)), s > g$ is indifferentiable from a random oracle when $h_1, \ldots, h_s$ are seen as random oracles to $\mathbb{F}_q$ by Farashahi *et al.* (2013). ■

6. Conclusion

We gave a complete overview of all existing encoding and indifferentiable hash functions into elliptic and hyperelliptic curves. We have also constructed a generic indifferentiable hash function for certain families of hyperelliptic curves, by using the unified formulas of Seck and Diarra (2018) and the framework of Farashahi *et al.* (2013).

Bibliography

Aranha D. F, Fouque P.A, Qian C., Tibouchi M and Zapalowicz J. C. (2014). Binary Elligator Squared. IACR Cryptology ePrint Archive. pp. 486.

Bernstein, D.J., Hamburg, M., Krasnova, A. and Lange T.(2013) Elligator: elliptic-curve points indistinguishable from uniform random strings. In : *Proceedings of the 2013 ACM SIGSAC conference on Computer &#38; communications security : CS '13*, pp. 967-980. http://doi.acm.org/10.1145/2508859.2516734, doi : 10.1145/2508859.2516734 978-1-4503-2477-9. Berlin, Germany. Publisher : ACM, New York, NY, USA.

Boneh D., Lynn B. and Shacham H.(2001) Short Signatures from the Weil Pairing. In : *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology, ASIACRYPT '01*, pp. 514-532. ISBN : 3-540-42987-5, http://dl.acm.org/citation.cfm?id=647097.717005, Springer-Verlag.

Boneh D. and Franklin M. K.(2001). Identity-Based Encryption from the Weil Pairing. In : *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology, CRYPTO '01*, ISBN : 3-540-42456-3, year = 2001, pp. 213–229, http://dl.acm.org/citation.cfm?id=646766.704155, Springer-Verlag, London, UK, UK

Boyko V., MacKenzie P. and Patel S.(2000). Provably Secure Password-authenticated Key Exchange Using Diffie-Hellman. In *Proceedings of the 19th International Conference on Theory and Application of Cryptographic Techniques, EUROCRYPT'00*, ISBN : 3-540-67517-5. (Bruges, Belgium). pp 156–171. http://dl.acm.org/citation.cfm?id=1756169.1756186, Springer-Verlag, Berlin, Heidelberg

Seck M. Boudjou H., Diarra N. and Cheikh Khlil O/ A. Y.(2017). On Indifferentiable Hashing into the Jacobian of Hyperelliptic Curves of Genus 2. In : *Progress in Cryptology - AFRICACRYPT 2017 - 9th International Conference on Cryptology in Africa, Dakar, Senegal, May 24-26, 2017, Proceedings*. pp. 205–222. Doi : 10.1007/978-3-319-57339-7_12

Seck M. and Diarra N.(2018). Unified Formulas for Some Deterministic Almost-Injective Encodings into Hyperelliptic Curves. In *Progress in Cryptology - AFRICACRYPT 2018 - 10th International Conference on Cryptology in Africa, Marrakesh, Morocco, May 7-9, 2018, Proceedings*. pp. 183–202. Doi : 10.1007/978-3-319-89339-6_11

Cha, J. C. and Cheon J. H.(2013) An Identity-Based Signature from Gap Diffie-Hellman Groups. In : *Proceedings of the 6th International Workshop on Theory and Practice in Public Key Cryptography: Public Key Cryptography, PKC '03*. IBSN : 3-540-00324-X. pp 18–30, http://dl.acm.org/citation.cfm?id=648120.746918, Springer-Verlag, London, UK, UK.

Brier E., Coron J.S, Icart T., Madore D., Randriam H. and Tibouchi M.(2010). Efficient Indifferentiable Hashing into Ordinary Elliptic Curves. In : *Advances in Cryptology - CRYPTO 2010, 30th Annual Cryptology Conference, Santa Barbara, CA, USA, August 15-19, 2010. Proceedings*. pp. 237–254. Doi : 10.1007/978-3-642-14623-7_13.

Icart T.(2009) How to Hash into Elliptic Curves. *Cryptology ePrint Archive*. http://eprint.iacr.org/2009/226.

Pierre-Alain Fouque P.A, Joux A. and Tibouchi M.(2013). Injective Encodings to Elliptic Curves. *Information Security and Privacy - 18th Australasian Conference, ACISP (2013, Brisbane, Australia, July 1-3, 2013. Proceedings)*. pp. 203–218. Doi : 10.1007/978-3-642-39059-3_14.

Pierre-Alain Fouque P.A, Joux A. and Tibouchi M.(2013). Injective Encodings to Elliptic Curves. *Information Security and Privacy - 18th Australasian Conference, ACISP (2013, Brisbane, Australia, July 1-3, 2013. Proceedings)*. pp. 203–218. Doi : 10.1007/978-3-642-39059-3_14.

Yu W., Wang K., Li B., He X. and Tian S.(2015). Hashing into Jacobi Quartic Curves. In *Proceedings of the 18th International Conference on Information Security - Volume 9290, ISC 2015*. ISBN : 978-3-319-23317-8. (Trondheim, Norway). pp. 355–375. Doi : 10.1007/978-3-319-23318-5_20. Springer-Verlag, New York, Inc.

Yu W., Wang K., Li B. and Tian S.(2013) About Hash into Montgomery Form Elliptic Curves. In : *Information Security Practice and Experience - 9th International Conference, ISPEC 2013* (Lanzhou, China, May 12-14, 2013). pp. 147–159. Doi : 10.1007/978-3-642-38033-4_11.

Yu W., Wang K., Li B., He X. and Tian S.(2016) Deterministic Encoding into Twisted Edwards Curves. In : *Information Security and Privacy - 21st Australasian Conference, ACISP 2016, Proceedings, Part II*, (Melbourne, VIC, Australia, July 4-6, 2016). pp. 285–297. Doi : 10.1007/978-3-319-40367-0_18

Shallue A. and van de Woestijne C. (2006) Construction of Rational Points on Elliptic Curves over Finite Fields. In : *Algorithmic Number Theory, 7th International Symposium, ANTS-VII, Berlin, Germany, July 23-28, 2006, Proceedings*. pp. 510–524. Doi : 10.1007/11792086_36.

He X., Yu W. and Wang K. (2015) Hashing into Generalized Huff Curves. In *Information Security and Cryptology - 11th International Conference, Inscrypt 2015, Beijing, China, November 1-3, 2015, Revised Selected Papers*. pp.22–44.

He X. and Yu W. and Wang k. (2017). Hashing into Twisted Jacobi Intersection Curves. In *Information Security and Cryptology - 13th International Conference, Inscrypt 2017, Xi'an, China, November 3-5, 2017, Revised Selected Papers*. pp.117–138.

Farashahi R. R. (2011). Hashing into Hessian Curves. In *Progress in Cryptology - AFRICACRYPT 2011 - 4th International Conference on Cryptology in Africa, Dakar, Senegal, July 5-7, 2011. Proceedings*. pp.278–289.

Kammerer J.G., Lercier R. and Renault G.(2010). Encoding Points on Hyperelliptic Curves over Finite Fields in Deterministic Polynomial Time. In *Pairing-Based Cryptography - Pairing 2010 - 4th International Conference, Yamanaka Hot Spring, Japan, December 2010. Proceedings*. pp:278–297.

Jablon D. P. (1996). Strong password-only authenticated key exchange.*Computer Communication Review*.26(5). pp.5–26.

Fouque P. A. and Tibouchi M. (2010). Deterministic Encoding and Hashing to Odd Hyperelliptic Curves. In *Pairing-Based Cryptography - Pairing 2010 - 4th International Conference, Yamanaka Hot Spring, Japan, December 2010. Proceedings*. pp.265–277.

Fouque P.A. and Tibouchi M. (2010).Estimating the Size of the Image of Deterministic Hash Functions to Elliptic Curves. In *Progress in Cryptology - LATINCRYPT 2010, First International Conference on Cryptology and Information Security in Latin America, Puebla, Mexico, August 8-11, 2010, Proceedings*.pp.81–91.

Fouque P. A. and Tibouchi M. (2012).Indifferentiable Hashing to Barreto-Naehrig Curves. *Progress in Cryptology - LATINCRYPT 2012 - 2nd International Conference on Cryptology and Information Security in Latin America, Santiago, Chile, October 7-10, 2012. Proceedings*.pp.1–17.

Koblitz N. (1989). Hyperelliptic Cryptosystems.*J. Cryptology*.1(3), pp:139-150.

Menezes A., Okamoto T. and Vanstone S. A. (1993).Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Trans. Information Theory*.39(5). pp. 1639–1646.

Diarra N., Sow D. and Cheikh Khlil A. Y. O. (2017). Indifferentiable Deterministic Hashing into Elliptic Curves. *Indifferentiable Deterministic Hashing into Elliptic Curves.*10(2). pp. 363–391.

Farashahi R. R., Fouque P. A., Shparlinski I. E., Tibouchi M. and Voloch J. F. (2013). Indifferentiable Deterministic Hashing To Elliptic And Hyperelliptic Curves.*Mathematics of Computation*82(281). pp. 491-512.

Skalba M. (2005). Points on elliptic curves over finite fields.*Acta Arith.*117. pp. 293–301.

Ulas M. (2007). Rational points on certain hyperelliptic curves over finite fields. *Bull. Pol. Acad. Sci. Math.* 55(2). pp. 97–104

Menezes A.J., Wu Y.h. and Zuccherato R.J. (1998). An elementary introduction to hyperelliptic curves. In *Algebraic Aspects of Cryptography, vol 3 of Algorithms and Computation in Mathematics*. Koblitz N. (Eds). pp.155–178.

Bernstein D. J. and Lange T. (2014).SafeCurves.

Silverman J.H. (1986).*The Arithmetic of Elliptic Curves*. Graduate Texts in Mathematics. Springer-Verlag.

Scholten J. and Vercauteren F.(2008). An Introduction to Elliptic and Hyperelliptic Curve Cryptography and the NTRU Cryptosystem