Efficient Indifferentiable Hashing into Ordinary Elliptic Curves[⋆]

Eric Brier[1], Jean-Sébastien Coron[2], Thomas Icart[2⋆⋆], David Madore[3], Hugues Randriam[3], and Mehdi Tibouchi[2,4⋆⋆⋆]

[1] Ingenico
eric.brier@ingenico.com
[2] Université du Luxembourg
jean-sebastien.coron@uni.lu, thomas.icart@m4x.org
[3] TELECOM-ParisTech
{david.madore,randriam}@enst.fr
[4] École normale supérieure
mehdi.tibouchi@ens.fr

Abstract. We provide the first construction of a hash function into ordinary elliptic curves that is indifferentiable from a random oracle, based on Icart's deterministic encoding from Crypto 2009. While almost as efficient as Icart's encoding, this hash function can be plugged into any cryptosystem that requires hashing into elliptic curves, while not compromising proofs of security in the random oracle model.
We also describe a more general (but less efficient) construction that works for a large class of encodings into elliptic curves, for example the Shallue-Woestijne-Ulas (SWU) algorithm. Finally we describe the first deterministic encoding algorithm into elliptic curves in characteristic 3.

1 Introduction

Hashing into Elliptic Curves. Many elliptic curve cryptosystems require to hash into an elliptic curve. For example in the Boneh-Franklin IBE scheme [6], the public-key for identity $id \in \{0,1\}^*$ is a point $Q_{id} = H_1(id)$ on the curve. This is also the case in many other pairing-based cryptosystems including IBE and HIBE schemes [1,19,22], signature and identity-based signature schemes [5,7,8,13,35] and identity-based signcryption schemes [10,25].

Hashing into elliptic curves is also required for some passwords based authentication protocols, for instance the SPEKE (Simple Password Exponential Key Exchange) [24] and the PAK (Password Authenticated Key exchange) [11], and also for discrete-log based signature schemes such as [14] when instantiated over an elliptic curve. In all those previous cryptosystems, security is proven when the hash function is seen as a random oracle into the curve. However, it remains to determine which hashing algorithm should be used, and whether it is reasonable to see it as a random oracle.

In [6], Boneh and Franklin use a particular supersingular elliptic curve E for which, in addition to the pairing operation, there exists a one-to-one mapping f from the base field $\mathbb{F}_p$ to $E(\mathbb{F}_p)$. This enables to hash using $H_1(m) = f(h(m))$ where h is a classical hash function from $\{0,1\}^*$ to $\mathbb{F}_p$. The authors show that their IBE scheme remains secure when h is seen as a random oracle into $\mathbb{F}_p$ (instead of H_1 being seen as a random oracle into $E(\mathbb{F}_p)$). However, when no pairing operation is required (as in [11,14,24]), it is more efficient to use ordinary elliptic curves, since supersingular curves require much larger security parameters due to the MOV attack [27].

For hashing into an ordinary elliptic curve, the classical approach is inherently probabilistic: one can first compute an integer hash value $x = h(m)$ and then determine whether x is the abscissa of a

[⋆] An extended abstract of this paper will appear at CRYPTO 2009. This is the full version.

[⋆⋆] Research carried out while working at Sagem Sécurité.

[⋆⋆⋆] Research carried out while on a visit to the Okamoto Research Laboratory at the NTT Information Sharing Platform.

point on the elliptic curve:

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

otherwise one can try $x + 1$ and so on. Using this approach the number of operations required to hash a message m depends on m, which can lead to a timing attack (see [9]). To avoid this attack, one can determine whether $x + i$ is the abscissa of a point, for *all* i between $0 \leq i < k$, and use for example the smallest such i; here k is a security parameter that gives an error probability of roughly 2^{-k}. However, this leads to a very lengthy hash computation.

The first algorithm to generate elliptic curve points in *deterministic* polynomial time was published in ANTS 2006 by Shallue and Woestijne [31]. The algorithm has running time $\mathcal{O}(\log^4 p)$ for any p, and $\mathcal{O}(\log^3 p)$ when $p \equiv 3 \pmod 4$, using standard multiplications. The rational maps in [31] were later simplified and generalized to hyper-elliptic curves by Ulas in [34]; we refer to this algorithm as the Shallue-Woestijne-Ulas (SWU) algorithm. Letting $f : \mathbb{F}_p \to E(\mathbb{F}_p)$ be the function defined by SWU, one can then hash in deterministic polynomial time using $H(m) = f(h(m))$ where h is any hash function into $\mathbb{F}_p$.

Another deterministic hash algorithm for ordinary elliptic curves was recently published by Icart in [23]. The algorithm works for $p \equiv 2 \pmod 3$, with complexity $\mathcal{O}(\log^3 p)$. Given any elliptic curve E defined over $\mathbb{F}_p$, Icart defines a function f that is an algebraic function from $\mathbb{F}_p$ into the curve. As previously given any hash function h into $\mathbb{F}_p$, one can use $H(m) = f(h(m))$ to hash into $E(\mathbb{F}_p)$. As shown in [23], H is one-way if h is one-way.

The Random Oracle Model (ROM). Many cryptosystems based on elliptic curves have been proven secure in the random oracle model, see for example [1,5,6,7,8,10,11,13,19,22,24,25,35]. In the random oracle model [3], the hash function is replaced by a publicly accessible random function (the random oracle); the adversary cannot compute the hash function by himself but instead he must query the random oracle. Obviously, a proof in the random oracle model is not fully satisfactory, because such a proof does not imply that the scheme will remain secure when the random oracle is replaced by a concrete hash function. Numerous papers have shown artificial schemes that are provably secure in the ROM but completely insecure when the RO is instantiated with any function family (see [12]). Despite these separation results, a proof in the ROM is believed to indicate that there are no structural flaws in the design of the system, and that no flaw will suddenly appear when a "well designed" hash function is used instead.

For a cryptosystem that requires a hash function H into an ordinary elliptic curve (such as [11,24]), one possibility could be to use $H(m) = f(h(m))$ where f is either Icart or SWU's function and h is a hash function into $\mathbb{F}_p$. However we know that neither Icart nor SWU's function generate all the points of E; for example, Icart's function covers only about 5/8 of the points [17,18]; moreover it is easy to see that the distribution of $f(h(m))$ is not uniform in the image of f. Therefore the current proofs in the random oracle model for H do not guarantee the security of the resulting scheme when $H(m) = f(h(m))$ is used instead (even if h is assumed to be ideal). In other words, even if a proof in the random oracle for H can indicate that there are no structural flaws in the design of the cryptosystem, using $H(m) = f(h(m))$ could introduce a flaw that would make the resulting cryptosystem completely insecure (we give an example in Section 5.2).

Our Results. We provide the first construction of a hash function H into ordinary elliptic curves with the property that *any cryptosystem* proven secure assuming H is a random oracle remains secure when our construction is plugged instead (still assuming that the underlying h is a random oracle). For this we use the indifferentiability framework of Maurer *et al.* [26]. As shown in [15], when a construction

H is indifferentiable from a random oracle, such a construction can then replace a random oracle in any cryptosystem, and the resulting scheme remains secure in the random oracle model for h.

Since the output of Icart and SWU functions only covers a fraction of the elliptic curve points, we cannot use the construction $H(m) = f(h(m))$ for indifferentiable hashing. Our main result is to show that for Icart's function f, we can use the following alternative construction which is almost as efficient:

$$H(m) := f(h_1(m)) + f(h_2(m))$$

where h_1, h_2 are two hash functions into $\mathbb{F}_p$, and + denotes elliptic curve addition. Therefore $H(m)$ can be used in any cryptosystem provably secure with random oracle into elliptic curves, and the resulting cryptosystem remains secure in the random oracle model for h_1 and h_2.

However the proof involves somewhat technical tools from algebraic geometry, and it is not so simple to adapt to other encodings such as the SWU algorithm. Therefore we describe a more general (but less efficient) construction that applies to a large class of encoding functions satisfying a few simple axioms. Those encodings include Icart's function, the SWU algorithm, new deterministic encodings in characteristic 3, etc. More precisely, given an elliptic curve E defined over $\mathbb{F}_p$ whose group of points is cyclic of order N with generator G, our general construction is as follows:

$$H(m) := f(h_1(m)) + h_2(m)G$$

where $h_1 : \{0,1\}^* \to \mathbb{F}_p$ and $h_2 : \{0,1\}^* \to \mathbb{Z}_N$ are two hash functions, and f is SWU or Icart's function. We show that $H(m)$ is indifferentiable from a random oracle when h_1 and h_2 are seen as random oracles. Intuitively, the term $h_2(m)G$ plays the role of a one-time pad; this ensures that $H(m)$ can behave as a random oracle even though $f(h_1(m))$ does not reach all the points in E. Note that one could not use $H(m) = h_2(m)G$ only since in this case the discrete logarithm of $H(m)$ would be known, which would make most protocols insecure.[5]

We also show how to extend the two previous constructions to hashing into a subgroup of an elliptic curve (whether its group of points is cyclic or not) and to hash functions into strings (rather than $\mathbb{F}_p$). We also describe a slightly more efficient variant of the SWU algorithm when $p \equiv 3 \pmod 4$. Finally, we describe the first deterministic encoding algorithms into elliptic curves in characteristic 3.

2 Preliminaries

2.1 Deterministic Encodings to Elliptic Curves

The indifferentiable hash function constructions proposed in this paper can be based on various *deterministic encoding functions* to elliptic curves. The first example of such an encoding is the one introduced by Boneh and Franklin in [6] for *supersingular* elliptic curves (see below). For ordinary (*i.e.* non-supersingular) elliptic curves, the two main encoding functions introduced thus far are due to Shallue and Woestijne with later improvements by Ulas (SWU) [31,34] on the one hand, and to Icart [23] on the other. They are defined on ordinary curves over finite fields of characteristic > 3; we describe them succinctly below. They also admit variants over binary fields.

Supersingular Curves and Boneh-Franklin Admissible Encoding An elliptic curve E over $\mathbb{F}_p$ is called supersingular when it has exactly $p+1$ points over $\mathbb{F}_p$. When $p \equiv 2 \pmod 3$, the map $x \mapsto x^3$ is a bijection, therefore the curves $Y^2 = X^3 + b$ are supersingular. One can then define the encoding

$$f : u \mapsto \left((u^2 - b)^{1/3}, u\right)$$

[5] For example in Boneh-Franklin IBE one could then decrypt any ciphertext.

and the hash function $H(m) := f(h(m))$, where h is a classical hash function into $\mathbb{F}_p$.

In the Boneh-Franklin scheme [6], one actually works in a subgroup $\mathbb{G}$ of prime order q of $E_{a,b}(\mathbb{F}_p)$; one lets ℓ be the co-factor such that $p + 1 = \ell \cdot q$. One requires that q does not divide ℓ (*i.e.* that q^2 does not divide $p + 1$). In order to hash into $\mathbb{G}$, one can therefore use the encoding $f_{\mathbb{G}}(u) := \ell f(u)$ and the hash function into $\mathbb{G}$:

$$H_{\mathbb{G}}(u) := f_{\mathbb{G}}(h(m)) \tag{1}$$

In [6], Boneh and Franklin introduced the following notion of admissible encoding:

Definition 1 (Boneh-Franklin admissible encoding [6]). *A function $f : S \to R$ between finite sets is an admissible encoding if it satisfies the following properties:*

1. *Computable: f is computable in deterministic polynomial time;*
2. *ℓ-to-1: for any $r \in R$, $\#f^{-1}(r) = \ell$;*
3. *Samplable: there exists a probabilistic polynomial time algorithm that for any $r \in R$ returns a random element in $f^{-1}(r)$.*

The authors of [6] show that if $f : S \to \mathbb{G}$ is an admissible encoding in this sense, then the Boneh-Franklin scheme is secure with $H(m) = f(h(m))$, in the random oracle model for $h : \{0,1\}^* \to S$. Since the function $f_{\mathbb{G}}$ is easily seen to be an admissible encoding under the previous definition, this shows that with $H_{\mathbb{G}}$ Boneh-Franklin is provably secure in the random oracle model for h.

In this paper, we introduce a new notion of admissible encoding that is more general than the previous notion. This enables us to define admissible encodings to ordinary elliptic curves as well as supersingular ones. Moreover, we show that the resulting hash functions are indifferentiable from a random oracle; therefore, they can be used in any cryptosystem, not only in Boneh-Franklin.

The Shallue-Woestijne-Ulas Algorithm. The first algorithm to generate elliptic curve points in *deterministic* polynomial time was published in ANTS 2006 by Shallue and Woestijne [31]. The maps were later simplified and generalized to hyperelliptic curves by Ulas in [34]; we recall those maps in the following result.

Lemma 1 (Ulas [34]). *Let $\mathbb{F}_q$ be a finite field and let $g(x) := x^3 + ax + b$, where $a, b \neq 0$. Let:*

$$X_1(t,u) = u \qquad X_2(t,u) = \frac{-b}{a}\left(1 + \frac{1}{t^4 g(u)^2 + t^2 g(u)}\right)$$
$$X_3(t,u) = t^2 g(u) X_2(t,u) \qquad U(t,u) = t^3 g(u)^2 g(X_2(t,u))$$

Then

$$U(t,u)^2 = g(X_1(t,u)) \cdot g(X_2(t,u)) \cdot g(X_3(t,u)) \tag{2}$$

From equation (2) at least one of $g(X_1(t,u))$, $g(X_2(t,u))$ and $g(X_3(t,u))$ must be a quadratic residue. Therefore, either $X_1(t,u)$, $X_2(t,u)$ or $X_3(t,u)$ is the abscissa of a point on the curve $y^2 = g(x)$. Computing the corresponding y requires to compute a square root; when $q \equiv 3 \pmod 4$, this is simply an exponentiation. However for $q \equiv 1 \pmod 4$, no deterministic algorithm is known for computing square roots. The Tonelli-Shanks algorithm [30] requires to use a non-quadratic residue, and it is unknown how to generate such non-quadratic residue deterministically. One of the main contributions of [31] is to show a deterministic variant of the Tonelli-Shanks algorithm when an equality of the form $a_0 a_1 a_2 = b^2$ holds, as with equation (2); this gives a deterministic encoding algorithm even for $q \equiv 1 \pmod 4$.

In Section 7, we provide a simplified version of the Ulas maps when $q \equiv 3 \pmod 4$. This enables to slightly improve the efficiency of the SWU algorithm.

Icart's Function. Consider an elliptic curve E over a finite field $\mathbb{F}_q$, with q odd and congruent to 2 mod 3, with equation:

$$Y^2 = X^3 + aX + b$$

Icart's function is defined in [23] as the map $f_{a,b} : \mathbb{F}_q \to E(\mathbb{F}_q)$ such that $f_{a,b}(u) = (x, y)$ where:

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

for $u \neq 0$, and $f_{a,b}(0) = O$, the neutral element of the elliptic curve. When $q \equiv 2 \pmod 3$ we have that $x \mapsto x^3$ is a bijection in $\mathbb{F}_q$ so cube roots are uniquely defined with $x^{1/3} = x^{(2q-1)/3}$. We recall the following properties of $f_{a,b}$:

Lemma 2 (Icart). *The function $f_{a,b}$ is computable in deterministic polynomial time. For any point $\varpi \in f_{a,b}(\mathbb{F}_q)$, the set $f_{a,b}^{-1}(\varpi)$ is computable in polynomial time and $\#f_{a,b}^{-1}(\varpi) \leq 4$. Moreover $q/4 < \#f_{a,b}(\mathbb{F}_q) < q$.*

We note that Icart's function can also be defined in a field of characteristic 2 (see Appendix D).

Summary. Table 1 lists currently published encoding algorithms into ordinary elliptic curves and their properties.

char(K)	normal form	discriminant $\varDelta$	encoding	condition
$\neq 2, 3$	$y^2 = x^3 + ax + b$	$-16(4a^3 + 27b^2)$	Icart [23]	$p \equiv 2 \pmod 3$
			SW [31]	–
			SWU [34]	–
			SWU, Sec. 7	$p \equiv 3 \pmod 4$
2	$y^2 + xy = x^3 + ax^2 + b$	b	Icart [23]	odd n
			SW [31]	–
3	$y^2 = x^3 + ax^2 + b$	$-a^3b$	Sec. 8.1	$\varDelta \in Q$
			Sec. 8.2	$\varDelta \notin Q$
			Sec. 8.3	–

Table 1. Known deterministic hashing algorithms into ordinary elliptic curves with discriminant $\varDelta \neq 0$. We denote by Q the set of quadratic residues. In char 2 we denote by n the extension degree.

2.2 Indifferentiability

We recall the notion of indifferentiability introduced by Maurer *et al.* in [26]. We define an *ideal primitive* as an algorithmic entity which receives inputs from one of the parties and delivers its output immediately to the querying party. A *random oracle* [3] into a finite set S is an ideal primitive which provides a random output in S for each new query; identical input queries are given the same answer.

Definition 2 (Indifferentiability [26]). *A Turing machine C with oracle access to an ideal primitive h is said to be $(t_D, t_S, q_D, \varepsilon)$-indifferentiable from an ideal primitive H if there exists a simulator S with oracle access to H and running in time at most t_S, such that for any distinguisher $\mathcal{D}$ running in time at most t_D and making at most q_D queries, it holds that:*

$$\left|\Pr\left[\mathcal{D}^{C^h, h} = 1\right] - \Pr\left[\mathcal{D}^{H, S^H} = 1\right]\right| < \varepsilon$$

C^h is said to be indifferentiable from H if ε is a negligible function of the security parameter k, for polynomially bounded q_D, t_D and t_S.

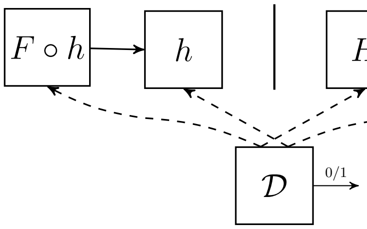

Fig. 1. The indifferentiability notion, illustrated with construction $C^h = F \circ h$ for some function F, and random oracles h and H.

It is shown in [26] that the indifferentiability notion is the "right" notion for substituting one ideal primitive by a construction based on another ideal primitive. That is, if the construction C^h is indifferentiable from an ideal primitive H, then C^h can replace H in any cryptosystem, and the resulting cryptosystem is at least as secure in the h model as in the H model; see [26] or [15] for a proof.

We also recall the definition of statistically indistinguishable distributions.

Definition 3. *Let X and Y be two random variables over a set S. The distributions of X and Y are ε-statistically indistinguishable if:*

$$\sum_{s \in S} \big| \Pr[X = s] - \Pr[Y = s] \big| \leq \varepsilon.$$

The two distributions are statistically indistinguishable if ε is a negligible function of the security parameter.

3 Admissible Encodings and Indifferentiability

Our goal is to construct a hash function into elliptic curves that is indifferentiable from a random oracle. First, we introduce our new notion of *admissible encoding.*

Definition 4 (Admissible Encoding). *A function $F : S \to R$ between finite sets is an ε-admissible encoding if it satisfies the following properties:*

1. *Computable: F is computable in deterministic polynomial time.*
2. *Regular: for s uniformly distributed in S, the distribution of $F(s)$ is ε-statistically indistinguishable from the uniform distribution in R.*
3. *Samplable: there is an efficient randomized algorithm $\mathcal{I}$ such that for any $r \in R$, $\mathcal{I}(r)$ induces a distribution that is ε-statistically indistinguishable from the uniform distribution in $F^{-1}(r)$.*

F is an admissible encoding if ε is a negligible function of the security parameter.

Our new definition can be seen as a generalization of Definition 1 recalled in Section 2.1. Namely Criterion 2 of our definition gives:

$$\sum_{r \in R} \left| \Pr_s[F(s) = r] - \frac{1}{\#R} \right| = \sum_{r \in R} \left| \frac{\#F^{-1}(r)}{\#S} - \frac{1}{\#R} \right| \leq \varepsilon \tag{3}$$

whereas Criterion 2 from Definition 1 requires $\#F^{-1}(r) = \#S/\#R$ for all $r \in R$; this is equivalent to $\varepsilon = 0$ in (3).

The following theorem shows that if $F : S \to R$ is an admissible encoding, then the hash function $H : \{0,1\}^* \to R$ with:

$$H(m) := F(h(m))$$

is indifferentiable from a random oracle into R when $h : \{0,1\}^* \to S$ is seen as a random oracle. This shows that the construction $H(m) = F(h(m))$ can replace a random oracle into R, and the resulting scheme remains secure in the random oracle model for h.

Theorem 1. *Let $F : S \to R$ be an ε-admissible encoding. The construction $H(m) = F(h(m))$ is $(t_D, t_S, q_D, \varepsilon')$-indifferentiable from a random oracle, in the random oracle model for $h : \{0,1\}^* \to S$, with $\varepsilon' = 4q_D\varepsilon$ and $t_S = 2q_D \cdot t_I$, where t_I is the maximum running time of F's sampling algorithm.*

Proof. We first describe our simulator; then we prove the indistinguishability property. As illustrated in Figure 1, the simulator must simulate random oracle h to the distinguisher $\mathcal{D}$, and the simulator has oracle access to random oracle H. It maintains a list L of previously answered queries. Our simulator is based on sampling algorithm $\mathcal{I}$ from F.

```
Simulator $\mathcal{S}$:
Input: $m \in \{0,1\}^*$
Output: $s \in S$
 1. If $(m, s) \in L$, then return $s$
 2. Query $H(m) = r$ and let $s \leftarrow \mathcal{I}(r)$
 3. Append $(m, s)$ to $L$ and return $s$.
```

We must show that the systems (C^h, h) and $(H, \mathcal{S}^H)$ are indistinguishable. We consider a distinguisher making at most q_D queries. Without loss of generality, we can assume that the distinguisher makes all queries to $h(m)$ (or $\mathcal{S}^H$) for which there was a query to $C^h(m)$ (or $H(m)$), and conversely; this gives a total of at most $2q_D$ queries. We can then describe the full interaction between the distinguisher and the system as a sequence of triples:

$$\mathsf{View} = (m_i, s_i, r_i)_{1 \le i \le 2q}$$

where $s_i = h(m_i)$ (or $\mathcal{S}^H(m_i)$) and $r_i = C^h(m_i)$ (or $H(m_i)$). Without loss of generality we assume that the m_i's are distinct.

In system (C^h, h) we have that $s_i = h(m_i)$. Therefore the s_i's are uniformly and independently distributed in S. Moreover we have $r_i = C^h(m_i) = F(s_i)$ for all i.

In system $(H, \mathcal{S}^H)$ we have that $r_i = H(m_i)$. Therefore the r_i's are uniformly and independently distributed in R. Moreover we have $s_i = \mathcal{I}(r_i)$ for all i.

Lemma 3. *For r uniformly distributed in R, the distribution of $s = \mathcal{I}(r)$ is 2ε-statistically indistinguishable from the uniform distribution in S.*

Proof. We let $\omega \in \Omega$ be the random used by $\mathcal{I}$. We must show that $\delta \le 2\varepsilon$, where:

$$\delta := \sum_{s \in S} \left| \Pr_{\omega,r}[\mathcal{I}(r) = s] - \frac{1}{\#S} \right|$$

Given $s \in S$, we have $\mathcal{I}(r) = s$ only if $r = f(s)$. Therefore, $\Pr_{\omega,r}[\mathcal{I}(r) = s] = \Pr_{\omega}[\mathcal{I}(r) = s]/\#R$ with $r = f(s)$. This gives:

$$\delta = \sum_{r \in R} \sum_{s \in F^{-1}(r)} \frac{1}{\#R} \left| \Pr_{\omega}[\mathcal{I}(r) = s] - \frac{\#R}{\#S} \right| \tag{4}$$

Since F is an ε-admissible encoding, by definition we have $\delta_1 \leq \varepsilon$, where:

$$\delta_1 := \sum_{r\in R} \left| \Pr_s[f(s)=r] - \frac{1}{\#R} \right| = \sum_{r\in R} \left| \frac{\#F^{-1}(r)}{\#S} - \frac{1}{\#R} \right| = \sum_{r\in R} \sum_{s\in F^{-1}(r)} \frac{1}{\#R} \left| \frac{\#R}{\#S} - \frac{1}{\#F^{-1}(r)} \right| \tag{5}$$

Moreover by definition for all $r \in R$ the distribution of $\mathcal{I}(r)$ is ε-statistically indistinguishable from the uniform distribution in $F^{-1}(r)$; this gives for all $r \in R$:

$$\sum_{s\in F^{-1}(r)} \left| \Pr_\omega[\mathcal{I}(r)=s] - \frac{1}{\#F^{-1}(r)} \right| < \varepsilon$$

which gives by summation over $r \in R$:

$$\delta_2 := \sum_{r\in R} \sum_{s\in F^{-1}(r)} \frac{1}{\#R} \left| \Pr_\omega[\mathcal{I}(r)=s] - \frac{1}{\#F^{-1}(r)} \right| < \varepsilon \tag{6}$$

From (4), (5) and (6) we have $\delta \leq \delta_1 + \delta_2$, which gives $\delta \leq \varepsilon + \varepsilon = 2\varepsilon$. □

From Lemma 3 we obtain that in system $(H, \mathcal{S}^H)$ the distribution of $s_i = \mathcal{I}(r_i)$ is 2ε-indistinguishable from the uniform distribution in S. Moreover from the definition of algorithm $\mathcal{I}$ we have that $r_i = F(s_i)$ except if $s_i = \perp$. Therefore, the statistical distance between View in system (C^h, h) and View in system $(H, \mathcal{S}^H)$ is at most $4q_D\varepsilon$. This concludes the proof of Theorem 1. □

4 Our Main Construction

Let E be an elliptic curve over a finite field $\mathbb{F}_q$ with $q \equiv 2 \pmod 3$. Let $f : \mathbb{F}_q \to E(\mathbb{F}_q)$ denote Icart's function to E. It is easy to see that Icart's function f is *not* an admissible encoding into E since as mentioned previously, the image of f comprises only a fraction of the elliptic curve points. Therefore we cannot use the construction $H(m) = f(h(m))$ for indifferentiable hashing (not even on the image f since the distribution of $f(u)$ is not uniform in $f(\mathbb{F}_q)$ for uniform $u \in \mathbb{F}_q$).

In this section, we describe a different construction which is almost as efficient. We consider the following map $F : (\mathbb{F}_q)^2 \mapsto E(\mathbb{F}_q)$ introduced by Icart in [23]:

$$F(u_1, u_2) = f(u_1) + f(u_2) \tag{7}$$

and we let $H(m) := F(h_1(m), h_2(m))$. We prove the following theorem:

Theorem 2. *If $q > 2^{13}$ is any $2k$-bit prime power congruent to $2 \bmod 3$ (even or odd), and if the j-invariant of E is not in $\{0; 2592\}$, then the function*

$$H(m) := f(h_1(m)) + f(h_2(m))$$

is $(t_D, t_S, q_D, \varepsilon')$-indifferentiable from a random oracle, where $\varepsilon' = 2^{10} \cdot q_D \cdot 2^{-k}$, in the random oracle model for $h_1, h_2 : \{0,1\}^* \to \mathbb{F}_q$.

Theorem 2 implies that this construction $H(m)$ can be used in any cryptosystem provably secure with random oracles into elliptic curves, and the resulting cryptosystem remains secure in the random oracle model for h_1 and h_2. We note that to prevent timing attacks (as in [9]), our construction H can easily be implemented in constant time since Icart's function can be implemented in constant time.

To prove this result, it is enough, in view of Theorem 1, to show that the function $F : (\mathbb{F}_q)^2 \to E(\mathbb{F}_q)$ given by is an ε-admissible encoding with $\varepsilon = 2^8 \cdot q^{-1/2}$.

F is clearly computable in deterministic polynomial time, so Criterion 1 of admissible encodings is satisfied. To prove Criterion 2, we denote for any $\varpi \in E(\mathbb{F}_q)$:

$$N(\varpi) = \#\{(u,v) \in (\mathbb{F}_q)^2 \mid f(u) + f(v) = \varpi\} = \#F^{-1}(\varpi)$$

Proposition 1. *If q is an odd prime power congruent to $2 \bmod 3$, and if the j-invariant of E is not in $\{0; 2592\}$, then for every point $\varpi \in E(\mathbb{F}_q)$ except at most 144, we have*

$$\left|q - N(\varpi)\right| \leq 2^7 \cdot \sqrt{q}$$

and all the remaining points ϖ satisfy $N(\varpi) \leq 2^5 \cdot q$.

Sections A.1 and A.2 in Appendix are devoted to the proof of this proposition. Intuitively, the idea of the proof is to show that, for all points $\varpi \in E(\mathbb{F}_q)$ except a few exceptional ones, $F^{-1}(\varpi)$ is an irreducible algebraic curve of bounded genus in the affine plane $\mathbb{A}^2$ over $\mathbb{F}_q$. The estimate for the number of points then follows from the Hasse-Weil bound. We also show in Appendix A.3 that the result extends to Icart's function f in characteristic 2.

4.1 Admissibility of $F(u,v) = f(u) + f(v)$

We can now prove that ε-admissibility, and hence Theorem 2, easily follow from Proposition 1. Since F is clearly computable, it suffices to show that it is ε-regular and ε-samplable. Now, for any ϖ not among the exceptional points, we have

$$\begin{aligned} \left|\frac{\#F^{-1}(\varpi)}{\#(\mathbb{F}_q)^2} - \frac{1}{\#E(\mathbb{F}_q)}\right| &\leq \left|\frac{\#F^{-1}(\varpi)}{\#(\mathbb{F}_q)^2} - \frac{1}{q}\right| + \left|\frac{1}{q} - \frac{1}{\#E(\mathbb{F}_q)}\right| \\ &\leq \frac{2^7}{q^{3/2}} + \frac{5}{q^{3/2}} \leq \frac{2^7+5}{q^{3/2}} \end{aligned}$$

And on the other hand, for exceptional points ϖ:

$$\left|\frac{\#F^{-1}(\varpi)}{\#(\mathbb{F}_q)^2} - \frac{1}{\#E(\mathbb{F}_q)}\right| \leq \frac{2^5}{q}$$

Thus, the statistical distance between the distribution of $F(u,v)$ for uniform (u,v) and the uniform distribution on the curve can be bounded as

$$\sum_{\varpi \in E(\mathbb{F}_q)} \left|\frac{N(\varpi)}{q^2} - \frac{1}{\#E(\mathbb{F}_q)}\right| \leq (q + 2\sqrt{q} + 1) \cdot \frac{2^7+5}{q^{3/2}} + 144 \cdot \frac{2^5}{q} \leq \frac{2^8}{q^{1/2}}$$

for $q > 2^{13}$, as required. This proves ε-regularity, with $\varepsilon = 2^8 \cdot q^{-1/2}$.

To see that F is ε-samplable, one can consider the following randomized sampling algorithm, where c is some constant to be determined later. Note that computing the set S in step 3 amounts to solving univariate polynomial equations over $\mathbb{F}_q$, which is easily done in polynomial time using an algorithm such as Berlekamp's [4].

Algorithm 1 Sampling algorithm for F

```
 1: repeat $\lceil c \cdot \lg q \rceil$ times
 2:     pick $v \in \mathbb{F}_q$ uniformly at random
 3:     $S \leftarrow f^{-1}\big(\varpi - f(v)\big)$
 4:     if $S = \varnothing$ then
 5:         next iteration
 6:     else
 7:         pick $i$ uniformly at random in $\{1, 2, 3, 4\}$
 8:         if $i \leq \#S$ then
 9:             $u \leftarrow i$-th element of $S$
10:             return $(u, v)$
11:         else
12:             next iteration
13:         end if
14:     end if
15: end repeat
16: return $\perp$
```

Since the image of f contains roughly $5/8 \cdot \#E(\mathbb{F}_q)$ points (as proved in [17,18]), a pigeonhole argument shows that $S \neq \varnothing$ with probability at least $2 \cdot 5/8 - 1 = 1/4$. Furthermore, we always have $\#S \leq 4$. Thus, each **repeat** iteration succeeds with probability at least $1/16$. Therefore, if we set $c = \frac{1}{2 \lg(16/15)}$, we find that Algorithm 1 succeeds with probability greater than $1 - \varepsilon/2$, and steps 7–13 ensure that it yields the uniform distribution on $F^{-1}(\varpi)$. This concludes the proof that F is an admissible encoding.

5 A More General Construction

Our construction of Section 4 has the advantage of being simple and efficient as it only requires two evaluations of Icart's function. However, the proof involves somewhat technical tools from algebraic geometry, and it is not so simple to adapt to other encoding functions, such as the SWU algorithm.

At the cost of a small performance penalty, however, we describe a more general construction that applies to a large class of encoding functions satisfying a few simple axioms. Those encoding functions include Icart's function, a simpler variant of the SWU function, new deterministic encodings in characteristic 3, etc. We call them *weak encodings*. They are defined as follows.

Definition 5 (Weak Encoding). *A function $f : S \to R$ between finite sets is said to be an α-weak encoding if it satisfies the following properties:*

1. *Computable: f is computable in deterministic polynomial time.*
2. *α-bounded: for s uniformly distributed in S, the distribution of $f(s)$ is α-bounded in R,* i.e. *the inequality* $\Pr_s[f(s) = r] \leq \alpha/\#R$ *holds for any $r \in R$.*
3. *Samplable: there is an efficient randomized algorithm $\mathcal{I}$ such that $\mathcal{I}(r)$ induces the uniform distribution in $f^{-1}(r)$ for any $r \in R$. Additionally $\mathcal{I}(r)$ returns $N_r = \#f^{-1}(r)$ for all $r \in R$.*

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

The main difference with an admissible encoding is that in Criterion 2, the distribution of $f(s)$ is only required to be α-bounded instead of being ε-indistinguishable from the uniform distribution. More precisely Criterion 2 for a weak encoding requires:

$$\forall r \in R, \quad \Pr_s[f(s) = r] = \frac{\#f^{-1}(r)}{\#S} \leq \frac{\alpha}{\#R} \tag{8}$$

instead of inequality (3).

From inequality (8) we have that any invertible function with bounded pre-image and bounded $\#R/\#S$ is a weak encoding; in particular, this is the case for Icart's function (the proof is given in Appendix B).

Lemma 4. *Icart's function $f_{a,b}$ is an α-weak encoding from $\mathbb{F}_q$ to $E_{a,b}(\mathbb{F}_q)$, with $\alpha = 4N/q$, where N is the order of $E_{a,b}(\mathbb{F}_q)$.*

When the output set is a group (such as the group of points on an elliptic curve), we demonstrate how to construct an admissible encoding from any weak encoding.

Theorem 3 (Weak → Admissible Encoding). *Let $\mathbb{G}$ be cyclic group of order N noted additively, and let G be a generator of $\mathbb{G}$. Let $f : S \to \mathbb{G}$ be an α-weak encoding. Then the function $F : S \times \mathbb{Z}_N \to \mathbb{G}$ with $F(s, x) := f(s) + xG$ is an ε-admissible encoding into $\mathbb{G}$, with $\varepsilon = (1-1/\alpha)^t$ for any t polynomial in the security parameter k, and $\varepsilon = 2^{-k}$ for $t = \alpha \cdot k$.*

We prove this theorem in the next section. As a consequence, we get that if $f : S \to \mathbb{G}$ is any weak encoding to a cyclic group with generator G, then the hash function $H : \{0,1\}^* \to \mathbb{G}$ defined by:

$$H(m) := f(h_1(m)) + h_2(m)G$$

where $h_1 : \{0,1\}^* \to \mathbb{F}_p$ and $h_2 : \{0,1\}^* \to \mathbb{Z}_N$ are two hash functions, is indifferentiable from a random oracle in the random oracle model for h_1 and h_2. In particular, we obtain the following corollary.

Proposition 2. *Let q be a k-bit prime power such that $q \equiv 2 \pmod 3$ and $E : y^2 = x^3 + ax + b$ an elliptic curve over $\mathbb{F}_q$ whose group of $\mathbb{F}_q$-points $E(\mathbb{F}_q)$ is cyclic with generator G. Let further $f_{a,b} : \mathbb{F}_q \to E(\mathbb{F}_q)$ be Icart's function. The construction $H(m) := f_{a,b}(h_1(m)) + h_2(m)G$ is $(t_D, t_S, q_D, \varepsilon)$-indifferentiable from a random oracle when the hash functions $h_1 : \{0,1\}^* \to \mathbb{F}_p$ and $h_2 : \{0,1\}^* \to \mathbb{Z}_N$ are seen as random oracles, with $\varepsilon = 4 \cdot q_D \cdot 2^{-k}$.*

We note that for elliptic curves with non-cyclic group, we can easily adapt the previous construction with $H(m) = f(h_1(m)) + h_2(m)G_1 + h_3(m)G_2$ where (G_1, G_2) are the generators of the group.

5.1 Proof of Theorem 3

We must show that F is an admissible encoding. Criterion 1 is trivially satisfied. Criterion 2 is also satisfied since for uniform $(s, x) \in S \times \mathbb{Z}_N$ the distribution of $F(s, x) = f(s) + xG$ is uniform in $\mathbb{G}$.

It remains to prove that Criterion 3 is satisfied for F. We consider an α-weak encoding $f : S \to R$ with sampling algorithm $\mathcal{I}_f$. We observe that when r is uniformly distributed in R, the distribution of $\mathcal{I}_f(r)$ is not necessarily close to uniform. Therefore from $\mathcal{I}_f$ we first construct a new sampling algorithm $\mathcal{I}'_f$ which artificially aborts with some well chosen probability (dependent on the input); this is to ensure that $\mathcal{I}'_f(r)$ is uniformly distributed in S (when $\mathcal{I}'_f$ does not abort).

```
Algorithm $\mathcal{I}'_f(r)$:
Input: $r \in R$
Output: $s \in S$ such that $f(s) = r$ or $s = \perp$
1. Let $N_r = \#f^{-1}(r)$ returned by $\mathcal{I}_f(r)$, and let $\delta_r = \#R \cdot N_r / (\alpha \cdot \#S)$.
2. With probability $1 - \delta_r$ return $\perp$; otherwise return $\mathcal{I}_f(r)$.
```

Lemma 5. *For uniformly distributed $r \in R$, the algorithm $\mathcal{I}'_f(r)$ aborts with probability at most $1 - 1/\alpha$, and the distribution of $\mathcal{I}'_f(r)$ conditioned on $\mathcal{I}'_f(r) \neq \perp$ is uniform in S.*

Proof. Given $r \in R$, we have that $\mathcal{I}'(r) \neq \perp$ with probability δ_r, where:

$$\delta_r = \frac{\#R \cdot \#f^{-1}(r)}{\alpha \cdot \#S}$$

This gives:

$$\Pr_r[\mathcal{I}'(r) \neq \perp] = \sum_{r \in R} \frac{1}{\#R} \cdot \delta_r = \frac{1}{\alpha \cdot \#S} \sum_{r \in R} \#f^{-1}(r) = \frac{1}{\alpha}$$

By definition we have that $\mathcal{I}(r)$ is uniformly distributed in the set $f^{-1}(r)$; this gives for any $r \in R$ and any $s \in f^{-1}(r)$:

$$\Pr[\mathcal{I}'(r) = s] = \delta_r \cdot \frac{1}{\#f^{-1}(r)} = \frac{\#R}{\alpha \cdot \#S}$$

Since $\mathcal{I}'(r) = s$ only if $r = f(s)$, this gives for any $s \in S$:

$$\Pr_r[\mathcal{I}'(r) = s] = \frac{1}{\alpha \cdot \#S}$$

and finally for any $s \in S$:

$$\Pr_r[\mathcal{I}'(r) = s \mid \mathcal{I}'(r) \neq \perp] = \frac{\Pr_r[\mathcal{I}'(r) = s]}{\Pr_r[\mathcal{I}'(r) \neq \perp]} = \frac{1}{\#S}$$

which shows that the distribution of $\mathcal{I}'(r)$ conditioned on $\mathcal{I}'(r) \neq \perp$ is uniform in S. This concludes the proof of Lemma 5. □

Finally, our sampling algorithm $\mathcal{I}_F(P)$ is constructed as follows, given algorithm $\mathcal{I}'_f$ from f:

```
Algorithm $\mathcal{I}_F$:
Input: $P \in \mathbb{G}$.
Output: $(s, x) \in S \times \mathbb{Z}_N$ such that $P = F(s, x) = f(s) + xG$, or $\perp$
 1. For $i = 1$ to $t$:
    (a) Randomly choose $x \in \mathbb{Z}_N$.
    (b) Let $s \leftarrow \mathcal{I}'_f(P - xG)$. If $s \neq \perp$, return $(s, x)$
 2. Return $\perp$.
```

We must show that for any $P \in \mathbb{G}$, the distribution of (s, x) is statistically close to uniform in $F^{-1}(P)$. From Lemma 5 and the uniform distribution of $P - xG \in \mathbb{G}$, we have that $s = \perp$ at step i with probability at most $1 - 1/\alpha$. Therefore algorithm $\mathcal{I}_F$ eventually outputs $s = \perp$ with probability at most $(1 - 1/\alpha)^t$. Moreover from Lemma 5 conditioned on $s \neq \perp$ the distribution of s in (s, x) is uniform in S. From:

$$F^{-1}(P) = \big\{(s, x) \in S \times \mathbb{Z}_N \mid f(s) + xG = P\big\} = \big\{\, (s, \log_G(P - f(s))) \mid s \in S\big\} \tag{9}$$

this implies that the distribution of (s, x) conditioned on $s \neq \perp$ is also uniform in $F^{-1}(P)$. Therefore, for any $P \in \mathbb{G}$ the distribution of $\mathcal{I}_F(P)$ is ε-statistically close to uniform in $F^{-1}(P)$, with $\varepsilon = (1 - 1/\alpha)^t$. Taking $t = \alpha \cdot k$, we can take $\varepsilon = 2^{-k}$; this concludes the proof of Theorem 3.

5.2 Discussion

We see that the construction $H(m) = f_{a,b}(h_1(m)) + f_{a,b}(h_2(m))$ of Section 4 requires two evaluations of Icart's function $f_{a,b}$ but no scalar multiplication. Since $f_{a,b}$ is essentially a field exponentiation, and in practice field exponentiation is roughly 10 times faster than scalar multiplication, the construction of Section 4 is approximately 5 times faster than the general construction of this section.

We note that for a number of existing schemes that are proven secure in the random oracle model into an elliptic curve, it would actually be sufficient to use $H(m) = f_{a,b}(h(m))$ only. This is because for many existing schemes the underlying complexity assumption (such as CDH or DDH) has the random self-reducibility property. So in the security proof one "programs" the RO using a random instance generated from the original problem instance. Then instead of letting $H(m) = P$ where P is from the random instance, one can adapt the proof by letting $f(h(m)) = P$. To make sure that $h(m)$ is uniformly distributed, one can "replay" the random instance generation depending on the number of solutions to the equation $f(u) = P$, as we do in the proof of Theorem 3.

However it is easy to construct a cryptosystem that is secure in the ROM but insecure with $H(m) = f(h(m))$. Consider for example the following symmetric-key encryption scheme: to encrypt with symmetric key k, generate a random r and compute $c = m + H(k, r)$ where the message m is a point on the curve and H hashes into the curve; the ciphertext is (c, r). This scheme is semantically secure in the ROM for H, since this is a one-time pad. But the scheme is insecure with $H(k, r) = f(h(k, r))$ because in this case $H(k, r)$ is not uniformly distributed, and for two messages m_0 and m_1 the attacker has a good advantage in distinguishing between the encryption of m_0 and m_1.[6]

The advantage of the two constructions of Sections 4 and 5 is that we have a simple criterion to plug them into existing schemes: it suffices that the scheme has a proof in the random oracle model. Whereas with $H(m) = f(h(m))$ it seems difficult to derive a formal criterion from the previous observations.

6 Extensions

In this section we consider four extensions that apply to both hash functions of Sections 4 and 5. We show how to hash into any prime order subgroup of an elliptic curve (with cyclic or non-cyclic group), how to use hash functions mapping into strings (rather than $\mathbb{F}_q$), how to take advantage of primes $p = 2^\ell - \omega$, and how to hash in characteristic 2.

6.1 Extension to a Prime Order Subgroup

In many applications only a prime order subgroup of E is used, so we show how to adapt the constructions of Sections 4 and 5 into a subgroup. Let E be an elliptic curve over $\mathbb{F}_q$ with N points, and let $\mathbb{G}$ be a subgroup of prime order N' and generator G. Let ℓ be the co-factor, *i.e.* $N = \ell \cdot N'$. We require that N' does not divide ℓ (*i.e.* that $(N')^2$ does not divide N), which is satisfied in practice for key size and efficiency reasons.

We show that it suffices to scalar multiply by co-factor ℓ the constructions of Sections 4 and 5 and the resulting constructions are still indifferentiable hash functions. More precisely, we consider the construction $H : \{0,1\}^* \to \mathbb{G}$ with:

$$H(m) := \ell\big(f_{a,b}(h_1(m)) + f_{a,b}(h_2(m))\big) \tag{10}$$

with $h_1, h_2 : \{0,1\}^* \to \mathbb{F}_q$ and $f_{a,b}$ is Icart's function.

[6] If we take Icart's function for f, this is even worse: given c the attacker can easily determine whether there exists u and v such that $c - m_0 = f_{a,b}(u)$ or $c - m_1 = f_{a,b}(v)$ and if one of the two equations has no solution then the attacker recovers the plaintext without uncertainty (this happens with good probability over r).

Proposition 3. *H is $(t_D, t_S, q_D, \varepsilon)$-indifferentiable from a random oracle, in the random oracle model for h_1 and h_2, with $\varepsilon = 2^{10} \cdot q_D \cdot 2^{-k}$.*

Informally, we show that the composition of two admissible encodings remains an (almost) admissible encoding, and that multiplication by a co-factor is an ε-admissible encoding, with $\varepsilon = 0$. This proves that H is an indifferentiable hash function. See Appendix C.2 for the proof.

The same result holds for the construction of Section 5. In this case for both cyclic and non-cyclic elliptic curves we simply use $H(m) = \ell f(h_1(m)) + h_2(m)G$ where G is a generator of the subgroup.

6.2 Extension to Random Oracles into Strings

The constructions in the previous sections are based on hash functions into $\mathbb{F}_{p^n}$ or $\mathbb{Z}_N$. However in practice a hash function outputs a fixed length string in $\{0,1\}^\ell$. We can modify our construction as follows. We consider an elliptic curve $E_{a,b}$ over $\mathbb{F}_p$, with p a $2k$-bit prime. We define the hash function $H : \{0,1\}^* \to E_{a,b}(\mathbb{F}_p)$ with:

$$H(m) := f_{a,b}\big(h_1(m) \bmod p\big) + f_{a,b}\big(h_2(m) \bmod p\big)$$

where h_1 and h_2 are two hash functions from $\{0,1\}^*$ to $\{0,1\}^{3k}$ and $f_{a,b}$ is Icart's function.

Proposition 4. *The previous hash function H is $(t_D, t_S, q_D, \varepsilon)$-indifferentiable from a random oracle, in the random oracle model for h_1 and h_2, with $\varepsilon = 2^{11} \cdot q_D \cdot 2^{-k}$.*

Informally, we first show that reduction modulo p is an admissible encoding from $\{0,1\}^\ell$ to $\mathbb{F}_p$ if $2^\ell \gg p$. Since the composition of two admissible encodings remains an (almost) admissible encoding, this shows that $F(u,v) = f(u \bmod p) + f(v \bmod p)$ is also an admissible encoding into $E(\mathbb{F}_p)$ and therefore H is an indifferentiable hash function. The same result holds for the general construction of Section 5. See Appendix C.3 for the proof.

6.3 Extension to Primes $p = 2^\ell - \omega$

We show a slightly more efficient construction for primes p of the form $p = 2^\ell - \omega$ for small ω, as used for example in the NIST curves [29]. Let $E_{a,b}(\mathbb{F}_p)$ be an elliptic curve of order N and generator G. Our construction $H : \{0,1\}^* \to E_{a,b}(\mathbb{F}_p)$ is as follows:

$$H(m) := f_{a,b}\big(h_1(m) \bmod p\big) + f_{a,b}\big(h_2(m) \bmod p\big)$$

where h_1 and h_2 are two hash functions from $\{0,1\}^*$ to $\{0,1\}^\ell$ and $f_{a,b}$ is Icart's function. Note that the output size ℓ of h_1 and h_2 is the same as the bit-size of p, as opposed to the previous section in which we took $\ell = 3k$ for a $2k$-bit prime p.

Proposition 5. *H is $(t_D, t_S, q_D, \varepsilon)$-indifferentiable from a random oracle, in the random oracle model for h_1 and h_2, with $\varepsilon = q_D \cdot \big(2^{10} \cdot 2^{-\ell/2} + 4\omega \cdot 2^{-\ell}\big)$.*

The proof is similar to the proof of Proposition 4, except that since $p \simeq 2^\ell$, reduction modulo p is now a generalized admissible encoding from $\{0,1\}^\ell$ to $\mathbb{Z}_p$. See Appendix C.4 for the full proof. The same result holds for the general construction of Section 5, where we can take $H(m) := f(h_1(m) \bmod p) + h_2(m)G$ with $h_1,\ h_2 : \{0,1\}^* \to \{0,1\}^\ell$.

Remark 1. We only need a single hash function $h : \{0,1\}^* \to \{0,1\}^\ell$ since we can obtain h_1 and h_2 by prepending a bit as input of h.

6.4 Extension to Elliptic Curves in Characteristic 2

Let E be an elliptic curve defined over $\mathbb{F}_{2^\ell}$ of order N and generator G. We refer to Appendix D for a description of Icart's function in characteristic 2. When working in $\mathbb{F}_{2^\ell}$ we can use a hash function into $\{0,1\}^\ell$ directly. More precisely, our construction is $H : \{0,1\}^* \to E_{a,b}(\mathbb{F}_{2^\ell})$ is defined as follows:

$$H(m) := f_{a,b}\big(h_1(m)\big) + f_{a,b}\big(h_2(m)\big)$$

Proposition 6. *H is $(t_D, t_S, q_D, \varepsilon)$-indifferentiable from a random oracle, in the oracle model for h_1 and h_2, with $\varepsilon = 2^{11} \cdot q_D \cdot 2^{-\ell/2}$.*

Proof. The proof is the same as the proof of Theorem 2. □

The same result applies to the general construction of Section 5, for which we can use:

$$H(m) := f_{a,b}\big(h_1(m)\big) + h_2(m)G \tag{11}$$

In Appendix E, we also recall the Shallue-Woestijne algorithm in characteristic 2 and show that it is a weak encoding into the curve; therefore Shallue-Woestijne can also be used in the hash function given by equation (11).

7 A Simpler Variant of the SWU Algorithm

In this section, we describe a slightly simpler variant of the Shallue-Woestijne-Ulas (SWU) algorithm over $\mathbb{F}_q$ recalled in Section 2.1, for $q \equiv 3 \pmod 4$. Note that this condition is usually satisfied in practice, since it enables to compute square roots efficiently.

Proposition 7 (Simplified Ulas maps). *Let $\mathbb{F}_q$ be a field and let $g(x) := x^3 + ax + b$, where $a, b \neq 0$. Let:*

$$X_2(t) = \frac{-b}{a}\left(1 + \frac{1}{t^4 - t^2}\right), \qquad X_3(t) = -t^2 X_2(t), \qquad U(t) = t^3 g(X_2(t))$$

Then $U(t)^2 = -g\big(X_2(t)\big) \cdot g\big(X_3(t)\big)$.

Proof. Let $g(x) = x^3 + ax + b$. Let u be a non-quadratic residue and consider the equation in x:[7]

$$g(u \cdot x) = u^3 \cdot g(x) \tag{12}$$

The first observation is that we can solve this equation for x because the terms of degree 3 cancel:

$$\begin{aligned} g(u \cdot x) = u^3 \cdot g(x) &\Leftrightarrow (ux)^3 + a(ux) + b = u^3(x^3 + ax + b) \\ &\Leftrightarrow aux + b = u^3 ax + u^3 b \\ &\Leftrightarrow x = \frac{b(u^3 - 1)}{a(u - u^3)} = \frac{-b}{a} \cdot \left(1 + \frac{1}{u + u^2}\right) \end{aligned}$$

The second observation is that since u is not a square, either $g(u \cdot x)$ or $g(x)$ must be a square. Therefore either x or $u \cdot x$ must be the abscissa of a point on the curve. Moreover when $q \equiv 3 \pmod 4$ we have that -1 is a quadratic non-residue and we can take $u = -t^2$. Finally from (12) we get:

$$g(u \cdot x) \cdot g(x) = u^3 \cdot g^2(x) = -t^6 \cdot g^2(x) = -(t^3 \cdot g(x))^2$$

which gives the maps of Proposition 7. □

[7] A similar equation was used in [28] to show that there exists infinitely many elliptic-curves with j-invariant equal to given $j \neq 0, 1728$ and with Mordell-Weil rank ≥ 2.

```
Simplified SWU algorithm:
Input: $\mathbb{F}_q$ such that $q \equiv 3 \pmod 4$, parameters $a, b$ and input $t \in \mathbb{F}_q$
Output: $(x, y) \in E_{a,b}(\mathbb{F}_q)$ where $E_{a,b} : y^2 = x^3 + ax + b$

1. $\alpha \leftarrow -t^2$
2. $X_2 \leftarrow \frac{-b}{a}\left(1 + \frac{1}{\alpha^2+\alpha}\right)$
3. $X_3 \leftarrow \alpha \cdot X_2$
4. $h_2 \leftarrow (X_2)^3 + a \cdot X_2 + b$;  $h_3 \leftarrow (X_3)^3 + a \cdot X_3 + b$
5. If $h_2$ is a square, return $(X_2, h_2^{(q+1)/4})$, otherwise return $(X_3, h_3^{(q+1)/4})$
```

Our simplified SWU algorithm from Section 7 defines a function $f'_{a,b} : \mathbb{F}_q \to E(\mathbb{F}_q)$ which is also a weak encoding into the curve.

Lemma 6. *The function $f'_{a,b}$ has pre-image size at most 8 and can be inverted on its image in polynomial time. Then $f'_{a,b}$ is an α-weak encoding with $\alpha = 8N/q$, where N is the elliptic curve order.*

Proof. To compute the pre-images of a point $P = (X_P, Y_P)$, the equations $X_2(t) = X_P$ and $X_3(t) = X_P$ must be solved. Since $X_2(t)$ and $X_3(t)$ are rational functions over the finite field $\mathbb{F}_q$, efficient algorithms such as the Berlekamp algorithm [4] can be used to compute the roots of the corresponding polynomial equations. The Berlekamp algorithm has complexity $\mathcal{O}(d^2 \log^3 q)$, where d is the degree of the equation. Since $\deg X_2(t) = 4$ and $\deg X_3(t) = 4$, we have that each equation has at most 4 solutions; therefore a point has at most 8 pre-images which can be efficiently computed. This proves Lemma 6. □

From Lemma 6 we can therefore use SWU's function $f'_{a,b}$ instead of Icart's function $f_{a,b}$ in the general construction of Section 5. We provide an implementation of our simplified SWU algorithm in Appendix G.

8 Hashing in Characteristic 3

In characteristic 3 the normal form of an elliptic curve with j-invariant $j \neq 0$ and discriminant $\Delta \neq 0$ is:

$$Y^2 = X^3 + aX^2 + b$$

with $\Delta = -a^3 b$. It is easy to see that Icart's technique cannot work in characteristic 3, and the SWU algorithm does not work in characteristic 3 because the elliptic curve has a different equation. In this section we show the first deterministic[8] encoding algorithms for elliptic curves in characteristic 3. We denote by Q the set of quadratic residues in the field. An implementation of the three algorithms is provided in Appendix H.

8.1 Algorithm for $\Delta \in Q$

Proposition 8. *Let $\mathbb{F}$ be a field of characteristic 3 and $g(x) = x^3 + ax^2 + b$ with $a \neq 0$ and $\Delta = -a^3 b \in Q$. Let $\eta \notin Q$ and let c such that $c^2 = -b/a$. Let*

$$X(t) = c \cdot \left(1 - \frac{1}{\eta \cdot t^2}\right)$$

Then either $g(X(t))$ or $g(\eta \cdot t^2 \cdot X(t))$ is a quadratic residue.

[8] We allow for a probabilistic pre-computation phase given the elliptic curve parameters.

Proof. As previously we choose $u \notin Q$ and we consider the equation in x:

$$g(u \cdot x) = u^3 \cdot g(x) \tag{13}$$

As previously the terms of degree 3 cancel, and using $u^3 - 1 = (u-1)^3$ in char 3, we get:

$$\begin{aligned} g(u \cdot x) = u^3 \cdot g(x) &\Leftrightarrow au^2x^2 + b = au^3x^2 + bu^3 \\ &\Leftrightarrow x^2 = \frac{b(u^3-1)}{a(u^2-u^3)} = \frac{b(u-1)^3}{au^2(1-u)} = \frac{-b}{a} \cdot \left(\frac{u-1}{u}\right)^2 \end{aligned}$$

Since $\Delta = -a^3b \in Q$, we have $-b/a \in Q$ so we can compute c such that $c^2 = -b/a$. Therefore we can take the following solution for equation (13):

$$x = c \cdot \left(1 - \frac{1}{u}\right)$$

For u we can take $u = \eta \cdot t^2$ where $\eta \notin Q$ is pre-computed. We recover the map $X(t)$ of Proposition 8. Moreover from equation (13) since $u^3 \notin Q$ either $g(x)$ or $g(u \cdot x)$ must be a quadratic residue. □

From Proposition 8 we easily deduce a deterministic encoding algorithm. Let denote by $f_1 : \mathbb{F}_q \to E(\mathbb{F}_q)$ the resulting encoding function.

Proposition 9. *The function f_1 has pre-image size at most 4 and can be inverted on its image in polynomial time. Then f_1 is an α-weak encoding with $\alpha = 4N/q$, where N is the elliptic curve order.*

Proof. Given $P = (X_P, Y_p) \in E$, the equation $X(t) = X_P$ gives a polynomial equation of degree 2 and therefore at most 2 solutions. The same holds for $\eta \cdot t^2 \cdot X(t) = X_p$. Therefore, the pre-image size of f_1 is at most 4. □

From Proposition 9 we can therefore use our function f_1 in the general construction of Section 5 and obtain an indifferentiable hash function. We provide an implementation in Appendix H.

8.2 Algorithm for $\Delta \notin Q$

Proposition 10. *Let $\mathbb{F}$ be a field of characteristic 3 and $g(x) = x^3 + ax^2 + b$ with $\Delta = -a^3b \notin Q$. Let $x_0 \in \mathbb{F}$ such that $g(x_0) = 0$. Let $\eta \notin Q$. Let :*

$$X(t) = -2 \cdot x_0 \cdot \left(1 + \frac{1}{\eta \cdot t^2}\right)$$

Let $X_1(t) = X(t) + x_0$ and $X_2(t) = \eta \cdot t^2 \cdot X(t) + x_0$. Then either $g(X_1(t))$ or $g(X_2(t))$ is a quadratic residue.

Proof. When $\Delta \notin Q$ we have that $g(x) = x^3 + ax^2 + b$ has a (unique) root $x_0 \in \mathbb{F}$. Therefore we can let:

$$f(x) = g(x + x_0) = x^3 + ax^2 + b'x$$

where $b' = 2 \cdot a \cdot x_0$. A deterministic encoding for elliptic curves of equation $y^2 = x^3 + ax^2 + b'x$ is already described in [34]. Given $u \notin Q$ one considers the equation in x:

$$\begin{aligned} f(u \cdot x) = u^3 \cdot f(x) &\Leftrightarrow au^2x^2 + b'ux = au^3x^2 + b'u^3x \\ &\Leftrightarrow ax(u^2 - u^3) = b'(u^3 - u) \\ &\Leftrightarrow axu^2(1-u) = b'u(u-1)(u+1) \\ &\Leftrightarrow x = \frac{-b'}{a} \cdot \left(\frac{u+1}{u}\right) = -2 \cdot x_0 \cdot \left(1 + \frac{1}{u}\right) \end{aligned}$$

Then either $f(x)$ or $f(u \cdot x)$ is a square, which implies that either $g(x + x_0)$ or $g(u \cdot x + x_0)$ is a square. Letting $u = \eta \cdot t^2$ where $\eta \notin Q$ one recovers the maps $X(t)$, $X_1(t)$ and $X_2(t)$. □

From Proposition 8 we easily deduce a deterministic encoding algorithm. Let denote by $f_2 : \mathbb{F}_q \to E(\mathbb{F}_q)$ the resulting encoding function.

Proposition 11. *The function f_2 has pre-image size at most 4 and can be inverted on its image in polynomial time. Then f_2 is an α-weak encoding with $\alpha = 4N/q$, where N is the elliptic curve order.*

Proof. The proof is the same as the proof of Proposition 9. □

8.3 Algorithm for any Δ

In this section we describe a different encoding algorithm that works for any discriminant Δ. We pre-compute $\eta \notin Q$ and z_0, y_0 such that $a\eta \cdot z_0^2 - y_0^2 + b = 0$.

```
Deterministic Encoding Algorithm in char 3:
Input: $t \in \mathbb{F}$
Output: $(x, y) \in E(\mathbb{F})$
1. Let $z = (-z_0 t^2 + 2y_0 t - a\eta z_0)/(a\eta - t^2)$
2. Let $y = y_0 + t \cdot (z - z_0)$
3. Let $k = a/(b - y^2)$
4. Find the unique solution $\alpha$ of the linear system $\alpha^3 + k \cdot \alpha = -k/a$
5. Let $x = 1/\alpha$ and output $(x, y)$
```

We show in Appendix F that this also defines a deterministic encoding into elliptic curves. Let denote by $f_3 : \mathbb{F}_q \to E(\mathbb{F}_q)$ the resulting encoding function.

Proposition 12. *The function f_3 has pre-image size at most 2 and can be inverted on its image in polynomial time. Then f_3 is an α-weak encoding with $\alpha = 2N/q$, where N is the elliptic curve order.*

Proof. Given $P = (X_P, Y_P) \in E$, the equation $y(t) = Y_P$ gives a degree 2 polynomial equation in t, which gives at most 2 solutions. Therefore the pre-image size of f_3 is at most 2. □

Acknowledgments

We would like to thank Pierre-Alain Fouque and the anonymous referees of Eurocrypt 2010 and Crypto 2010 for useful comments on this paper.

References

1. Joonsang Baek and Yuliang Zheng. Identity-based threshold decryption. In Bao et al. [2], pages 262–276.
2. Feng Bao, Robert H. Deng, and Jianying Zhou, editors. *Public Key Cryptography - PKC 2004, 7th International Workshop on Theory and Practice in Public Key Cryptography, Singapore, March 1-4, 2004*, volume 2947 of *Lecture Notes in Computer Science*. Springer, 2004.
3. Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.
4. Elwyn R. Berlekamp. *Algebraic Coding Theory*. McGraw-Hill, 1968.
5. Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In Desmedt [16], pages 31–46.

6. Dan Boneh and Matthew K. Franklin. Identity-based encryption from the weil pairing. In Joe Kilian, editor, *CRYPTO*, volume 2139 of *Lecture Notes in Computer Science*, pages 213–229. Springer, 2001.
7. Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *EUROCRYPT*, pages 416–432, 2003.
8. Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. In Colin Boyd, editor, *ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 514–532. Springer, 2001.
9. Colin Boyd, Paul Montague, and Khanh Quoc Nguyen. Elliptic curve based password authenticated key exchange protocols. In Vijay Varadharajan and Yi Mu, editors, *ACISP*, volume 2119 of *Lecture Notes in Computer Science*, pages 487–501. Springer, 2001.
10. Xavier Boyen. Multipurpose identity-based signcryption (a swiss army knife for identity-based cryptography). In Dan Boneh, editor, *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 383–399. Springer, 2003.
11. Victor Boyko, Philip D. MacKenzie, and Sarvar Patel. Provably secure password-authenticated key exchange using diffie-hellman. In Bart Preneel, editor, *EUROCRYPT*, volume 1807 of *Lecture Notes in Computer Science*, pages 156–171. Springer, 2000.
12. Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004.
13. Jae Choon Cha and Jung Hee Cheon. An identity-based signature from gap diffie-hellman groups. In Desmedt [16], pages 18–30.
14. Benoît Chevallier-Mames. An efficient CDH-based signature scheme with a tight security reduction. In Shoup [32], pages 511–526.
15. Jean-Sébastien Coron, Yevgeniy Dodis, Cécile Malinaud, and Prashant Puniya. Merkle-damgård revisited: How to construct a hash function. In Shoup [32], pages 430–448.
16. Yvo Desmedt, editor. *Public Key Cryptography - PKC 2003, 6th International Workshop on Theory and Practice in Public Key Cryptography, Miami, FL, USA, January 6-8, 2003, Proceedings*, volume 2567 of *Lecture Notes in Computer Science*. Springer, 2002.
17. Reza R. Farashahi, Igor E. Shparlinski, and José F. Voloch. On hashing into elliptic curves, 2010. preprint available from `http://www.ma.utexas.edu/users/voloch/preprint.html`.
18. Pierre-Alain Fouque and Mehdi Tibouchi. Estimating the size of the image of deterministic hash functions to elliptic curves. Cryptology ePrint Archive, Report 2010/037, 2010. `http://eprint.iacr.org/`.
19. Craig Gentry and Alice Silverberg. Hierarchical id-based cryptography. In Zheng [36], pages 548–566.
20. Alexander Grothendieck and Jean Dieudonné. Éléments de géométrie algébrique III. Étude cohomologique des faisceaux cohérents, première partie. *Publ. Math. IHES*, 11:5–167, 1961.
21. Robin Hartshorne. *Algebraic Geometry*, volume 52 of *Graduate Texts in Mathematics*. Springer, 1977.
22. Jeremy Horwitz and Ben Lynn. Toward hierarchical identity-based encryption. In Lars R. Knudsen, editor, *EUROCRYPT*, volume 2332 of *Lecture Notes in Computer Science*, pages 466–481. Springer, 2002.
23. Thomas Icart. How to hash into elliptic curves. In Shai Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 303–316. Springer, 2009.
24. David P. Jablon. Strong password-only authenticated key exchange. *SIGCOMM Comput. Commun. Rev.*, 26(5):5–26, 1996.
25. Benoît Libert and Jean-Jacques Quisquater. Efficient signcryption with key privacy from gap diffie-hellman groups. In Bao et al. [2], pages 187–200.
26. Ueli M. Maurer, Renato Renner, and Clemens Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In Moni Naor, editor, *TCC*, volume 2951 of *Lecture Notes in Computer Science*, pages 21–39. Springer, 2004.
27. Alfred Menezes, Tatsuaki Okamoto, and Scott A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, 39(5):1639–1646, 1993.
28. Jean-François Mestre. Rang de courbe elliptiques d'invariant donné. *Comptes rendus de l'Académie des sciences. Série 1, Mathématique*, 314(12):297–319, 1992.
29. NIST. *FIPS PUB 186-3: Digital Signature Standard (DSS).* June 2009.
30. Ivan Niven, Herbert S. Zuckerman, and Hugh L. Montgomery. *An Introduction to the Theory of Numbers.* Wiley, fifth edition, 1991.
31. Andrew Shallue and Christiaan van de Woestijne. Construction of rational points on elliptic curves over finite fields. In Florian Hess, Sebastian Pauli, and Michael E. Pohst, editors, *ANTS*, volume 4076 of *Lecture Notes in Computer Science*, pages 510–524. Springer, 2006.
32. Victor Shoup, editor. *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*. Springer, 2005.
33. W.A. Stein et al. *Sage Mathematics Software (Version 4.1).* The Sage Development Team, 2009. `http://www.sagemath.org`.
34. Maciej Ulas. Rational points on certain hyperelliptic curves over finite fields. *Bull. Polish Acad. Sci. Math.*, 55(2):97–104, 2007.

35. Fangguo Zhang and Kwangjo Kim. Id-based blind signature and ring signature from pairings. In Zheng [36], pages 533–547.
36. Yuliang Zheng, editor. *Advances in Cryptology - ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, December 1-5, 2002, Proceedings*, volume 2501 of *Lecture Notes in Computer Science*. Springer, 2002.

A Proof of Proposition 1

A.1 Geometric Interpretation of Icart's Function

Icart's function f admits a natural extension to the projective line over $\mathbb{F}_q$ by setting $f(\infty) = O$, the neutral element of the elliptic curve. Then, consider the graph of f:

$$C = \{(u, \varpi) \in \mathbb{P}^1 \times E \mid f(u) = \varpi\}$$

As shown in [23, Lemma 3], C is the closed subscheme of $\mathbb{P}^1 \times E$ defined by

$$u^4 - 6xu^2 + 6yu - 3a = 0 \tag{14}$$

In other words, Icart's function is the algebraic correspondence between $\mathbb{P}^1$ and E given by (14).

Let j be the j-invariant of E:

$$j = 1728 \cdot \frac{4a^3}{4a^3 + 27b^2} \in \mathbb{F}_q$$

Save for a few exceptional values of j, we can precisely describe the geometry of C.

Lemma 7. *If $j \notin \{0; 2592\}$, the subscheme C is a geometrically integral curve on $\mathbb{P}^1 \times E$ with one triple point at infinity and no other singularity. Its normalization $\widetilde{C}$ is a smooth, geometrically integral curve of genus 7. The natural map $h \colon \widetilde{C} \to E$ is a morphism of degree 4 ramified at 12 distinct finite points of $E(\bar{\mathbb{F}}_q)$, with ramification index 2.*

Proof. As usual with Icart's function, we assume that E is not supersingular, i.e. a and j are non-zero. The subscheme C is then clearly reduced, and was shown in [17,18] to be geometrically connected. It is thus a geometrically integral curve on the surface $\mathbb{P}^1 \times E$.

Let us determine its singular locus. First consider the affine patch of C given by $u \neq \infty$ and $\varpi \neq O$. It can be represented as the algebraic set of points (u, x, y) in affine 3-space satisfying $y^2 = x^3 + ax + b$ as well as equation (14). It is smooth at all points where the map

$$(u, x, y) \mapsto \big(y^2 - (x^3 + ax + b), u^4 - 6xu^2 + 6yu - 3a\big)$$

is of rank 2. The gradient of this map is

$$(u, x, y) \mapsto \begin{pmatrix} 0 & -3x^2 - a & 2y \\ 4u^3 - 12xu + 6y & -6u^2 & 6u \end{pmatrix}$$

Since E is smooth, this is of rank 2 at a point of the curve unless:

$$\begin{cases} y^2 = x^3 + ax + b \\ u^4 - 6xu^2 + 6yu - 3a = 0 \\ 4u^3 - 12xu + 6y = 0 \\ \begin{vmatrix} -3x^2 - a & 2y \\ -6u^2 & 6u \end{vmatrix} = 6u(2uy - 3x^2 - a) = 0 \end{cases}$$

Eliminating u, x, y between those four equations, we find $-24a^4 - 162ab^3 = -6a(4a^3 + 27b^2) = 0$, which is impossible. Therefore, there is no singular point in this affine patch.

Turning now to points at infinity, we denote by v the local coordinate $1/u$ on $\mathbb{P}^1$ in a neighborhood of ∞, and let $(z, w) = (1/y, x/y)$ be local coordinates on E in a neighborhood of O. First note that there is no point $(\infty, \varpi) \in C$ with $\varpi \neq O$. Indeed, writing equation (14) in terms of (v, x, y):

$$3av^4 - 6yv^3 + 6xv^2 - 1 = 0$$

we see that $v = 0$ is never a root.

Similarly, let us find points of the form (u, O) with $u \neq \infty$. In terms of (u, z, w), equation (14) becomes

$$zu^4 - 6wu^2 + 6u - 3az = 0$$

For $z = w = 0$, we get only one root $u = 0$. Thus, the only point on C of the form (u, O) with $u \neq \infty$ is $(0, O)$, and it is easily seen to be regular: the elliptic curve equation becomes $z = bz^3 + awz^2 + w^3$, and hence the map $\mathbb{A}^3 \to \mathbb{A}^2$ defining C in this patch has the following Jacobian matrix:

$$\begin{pmatrix} 0 & -3bz^2 - 2aw + 1 & -a - 3w^2 \\ 4zu^3 - 12wu + 6 & u^4 - 3a & -6u^2 \end{pmatrix}$$

which is of rank 2 for $u = z = w = 0$ (recall that the characteristic does not divide 6).

Finally, the point (∞, O) lies on C and is a triple point. Indeed, let $\mathscr{O} = \mathbb{F}_q[v, z, w]/(bz^3 + awz^2 - z + w^3)_{(v,z,w)}$ be the local ring of $\mathbb{P}^1 \times E$ at (∞, O), and write the local equation of C at this point:

$$3azv^4 - 6v^3 + 6wv^2 - z \in \mathscr{O}$$

Consider $\mathfrak{m} = (v, z, w)$ the maximal ideal of $\mathscr{O}$. Since $z = bz^3 + awz^2 + w^3 \in \mathfrak{m}^3$, we see that the curve equation belongs to $\mathfrak{m}^3$. It isn't in $\mathfrak{m}^4$, however, so the multiplicity of (∞, O) is exactly 3 (see [21, Ex. V.3.4]).

Thus, the normalization $\widetilde{C}$ of C is a smooth geometrically integral curve, and $\widetilde{C} \to C$ is an isomorphism outside (∞, O), whereas the fiber over (∞, O) consists of three points.

Consider now the map $h : \widetilde{C} \to E$ deduced from the second projection $C \to E$. Since equation (14) is of degree 4, h is a morphism of degree 4 as well. The fiber at the origin O of E contains 4 points, namely $(0, \infty)$ and the 3 points of $\widetilde{C}$ over the singular point of C. In particular, h is unramified at infinity. The ramification points are thus the finite points (x, y) of E where (14) has a multiple root, i.e. where the discriminant $\mathit{\Delta}$ vanishes.

Recall that $\mathit{\Delta}$ is a polynomial of degree 6 in x. It has 6 simple roots over the algebraic closure of $\mathbb{F}_q$ provided that its own discriminant:

$$\operatorname{disc}(\mathit{\Delta}) = 2^{58} \cdot 3^{42} \cdot (4a^3 + 27b^2)^2 \cdot (-4a^3 + 81b^2)^3$$

is nonzero, i.e. $4a^3 \neq 81b^2$, or $j \neq 2592$. We will assume this in what follows.

None of these roots x corresponds to a point (x, y) on E such that $y = 0$. Indeed, eliminating x between $\mathit{\Delta} = 0$ and $x^3 + ax + b = 0$, we find $a(4a^3 + 27b^2) = 0$, which is impossible. Thus, each root of $\mathit{\Delta}$ corresponds to exactly two points of E where h is ramified. Hence the 12 distinct ramification points on $E(\bar{\mathbb{F}}_q)$.

Eliminating u, x, y between equation (14) and its first and second derivatives as well as $y^2 = x^3 + ax + b$, we find $12(-4a^3 + 81b^2) = 0$, which contradicts our assumption that $j \neq 2592$. It follows that equation (14) cannot have a triple root. Thus, all 12 ramification points of h have ramification index 2. This allows us to compute the genus $g_{\widetilde{C}}$ of $\widetilde{C}$ using the Riemann-Hurwitz formula:

$$2g_{\widetilde{C}} - 2 = 4(2 \cdot 1 - 2) + 12 \quad \text{hence} \quad g_{\widetilde{C}} = 7$$

as required. $\square$

A.2 The Square Correspondence

In this context, the function $(u,v) \mapsto f(u) + f(v)$ occurring in our hash function construction admits the following description. A point (u,v) in the affine plane $\mathbb{A}^2$, or more generally in $\mathbb{P}^1 \times \mathbb{P}^1$, corresponds to ϖ on the elliptic curve E if and only if there is some point $(\alpha,\beta) \in \widetilde{C} \times \widetilde{C}$ over (u,v) such that $h(\alpha) + h(\beta) = \varpi$.

Consider the surface $S = \widetilde{C} \times \widetilde{C}$, and define the following two morphisms. The map $p : S \to \mathbb{P}^1 \times \mathbb{P}^1$ is the square of the first projection, and $s : S \to E$ is obtained by composing $h \times h : S \to E \times E$ with the group law $E \times E \to E$. Then the set of points $(u,v) \in \mathbb{P}^1 \times \mathbb{P}^1$ corresponding to a given $\varpi \in E$ is exactly $p(s^{-1}(\varpi))$ (and we can take the intersection with $\mathbb{A}^2$ if we are only interested in affine points). This allows us to give a geometric proof of Proposition 1.

Let us first describe the geometry of the fibers $s^{-1}(\varpi)$. Denote by $\rho_1, \dots, \rho_{12}$ the 12 geometric points of E over which h is ramified, and let $R = \{\rho_i + \rho_j\}_{1 \leq i,j \leq 12} \subset E$. The map s is of rank 1 at (α,β) if and only if h is of rank 1 at at least one of α or β, which is certainly the case when $h(\alpha)$ or $h(\beta)$ is not one the ρ_i. Therefore, s is smooth of relative dimension 1 over the open subscheme $E_0 = E - R$, and all points in E_0 have smooth curves on S as fibers. The following lemma makes this more precise.

Lemma 8. *The fibers of s at all geometric points of E_0 are smooth connected curves on $S_{\overline{\mathbb{F}}_q}$ of genus 49.*

Proof. The morphism $s : S \to E$ is projective, and thus proper, so it admits a Stein factorization $S \to E_1 \to E$, where $s_1 : S \to E_1$ has connected geometric fibers, and $E_1 \to E$ is finite [20, 4.3.3]. We will show that, in fact, $E_1 = E$.

Consider $\alpha_i \in \widetilde{C}$, $i = 0, \dots, 3$, the four points such that $h(\alpha_i) = O$. Then h factors as $\widetilde{C} \to S \to E_1 \to E$ where the first arrow is given by $\beta \mapsto (\alpha_i, \beta)$ for some fixed $i \in \{0, \dots, 3\}$. In particular, we have surjective morphisms of curves $\widetilde{C} \to E_1 \to E$. The function field $K(E_1)$ of E_1 is thus an intermediate field of the quartic extension $K(\widetilde{C})/K(E)$. But we know by [17,18] that this quartic extension has a normal closure of Galois group S_4. Thus, it doesn't have any non trivial subextension: we must either have $E_1 = \widetilde{C}$ or $E_1 = E$. In the latter case, we are done. Otherwise, $\widetilde{C} \to E_1$ is the identity map, since $K(\widetilde{C})$ doesn't have any non trivial automorphism over $K(E)$. This implies $s_1(\alpha_i, \alpha_j) = \alpha_j$ for all i,j. But by symmetry, we also have $s_1(\alpha_i, \alpha_j) = \alpha_i$, a contradiction.

Hence, $E_1 = E$ and s has connected geometric fibers. In particular, the fiber $Z = s^{-1}(\varpi)$ at any geometric point ϖ of E_0 is a smooth connected curve. Let us compute its genus g_Z. To do so, observe that the image of Z under $h \times h : S_{\overline{\mathbb{F}}_q} \to (E \times E)_{\overline{\mathbb{F}}_q}$ is the curve E' of points (ς, τ) such that $\varsigma + \tau = \varpi$. E' is clearly isomorphic to $E_{\overline{\mathbb{F}}_q}$, and is thus of genus 1. Since $h : \widetilde{C} \to E$ is of degree 4, any given point (ς, τ) on E' has $4^2 = 16$ pre-images by $Z \to E'$, except when either ς or τ is one of the 12 ramification points $\rho_1, \dots, \rho_{12}$ of h (note that ς and τ cannot be both ramification points since ϖ is outside R). In this latter case, (ς, τ) has $3 \cdot 4 = 12$ pre-images. Thus, $Z \to E'$ is a morphism of degree 16 with $2 \cdot 12 = 24$ ramification points in E', each of ramification type $(2,1,1,2,1,1,2,1,1,2,1,1)$. Applying the Riemann-Hurwitz formula, we get

$$2g_Z - 2 = 16(2 \cdot 1 - 2) + 24 \cdot 4 \quad \text{hence} \quad g_Z = 49$$

as stated. □

Consider now a fiber Z of s at some $\mathbb{F}_q$-point ϖ of E not in R. The previous description says that Z is a smooth geometrically integral curve of genus 49 on S. This gives a precise estimate of the number of $\mathbb{F}_q$-points on Z in view of the Hasse-Weil bound:

$$\left| q + 1 - \#Z(\mathbb{F}_q) \right| \leq 98\sqrt{q}$$

What we are interested in, however, is the number of points in $p(Z)$, or more precisely even, in $p(Z) \cap \mathbb{A}^2$. But those numbers are related in a simple way when Icart's function is well-defined, i.e. $q \equiv 2 \pmod 3$.

Lemma 9. *Suppose that* $q \equiv 2 \pmod 3$*, and let* N *be the number of* $\mathbb{F}_q$*-points in* $p(Z) \cap \mathbb{A}^2$*. Then we have*

$$q - 98\sqrt{q} - 23 \leq N \leq q + 98\sqrt{q} + 1$$

Proof. Let D be the closed subscheme of points (α, β) on S such that $p(\alpha, \beta)$ has at least one component at infinity in $\mathbb{P}^1 \times \mathbb{P}^1$. Denote further by U be the complementary open subscheme: $U = p^{-1}(\mathbb{A}^2)$. Then p induces a bijection $U(\mathbb{F}_q) \to \mathbb{A}^2(\mathbb{F}_q) \subset (\mathbb{P}^1 \times \mathbb{P}^1)(\mathbb{F}_q)$. This is a direct consequence of the fact that $t \mapsto t^3$ is a bijection in $\mathbb{F}_q$, as explained in the proof of [23, Lemma 3].

In particular, the $\mathbb{F}_q$-points of $p(Z) \cap \mathbb{A}^2$ are in bijection with $(Z \cap U)(\mathbb{F}_q)$. This immediately yields the upper bound:

$$N \leq \#Z(\mathbb{F}_q) \leq q + 1 + 98\sqrt{q}$$

To obtain the lower bound, it suffices to estimate the number of $\mathbb{F}_q$-points on Z outside of U, i.e. on $Z \cap D$. This is certainly bounded above by the number of geometric points on $Z \cap D$, which is itself not greater than the intersection number $Z \cdot D$ when Z and D are regarded as 1-cycles on S.

Let l and m denote the divisor classes on $S = \widetilde{C} \times \widetilde{C}$ of $\widetilde{C} \times \{\alpha\}$ and $\{\alpha\} \times \widetilde{C}$ for an arbitrary α. Then $D \equiv n_\infty(l + m)$, where $\equiv$ denotes numerical equivalence and n_∞ is the number of points on $\widetilde{C}$ mapping to ∞ in $\mathbb{P}^1$. As seen in the proof of Lemma 7, we have $n_\infty = 3$ (the three points of $\widetilde{C}$ lying over the singular point of C). Hence

$$Z \cdot D = 3Z \cdot (l + m)$$

On the other hand, the canonical divisor K of S satisfies $K \equiv (2g_{\widetilde{C}} - 2)(l + m) = 12(l + m)$ (see e.g. [21, Ex. V.1.5(b) and V.1.9(b)]). Now K appears in the adjunction formula of the intersection theory of surfaces [21, Proposition V.1.5]:

$$2g_Z - 2 = Z \cdot K + Z^2 = 4Z \cdot D + Z^2$$

Since Z is a fiber of s_0, its self-intersection number is zero: any two fibers of s_0 are algebraically equivalent and disjoint. Thus $Z^2 = 0$ and we get

$$Z \cdot D = \frac{1}{4} \cdot (2g_Z - 2) = 24$$

In particular, $\#(Z \cap D)(\mathbb{F}_q) \leq 24$ and hence

$$N \geq \#Z(\mathbb{F}_q) - 24 \geq q - 98\sqrt{q} - 23$$

□

The first part of Proposition 1 now follows from the previous propositions: under the hypotheses of that theorem, if $\varpi \in E(\mathbb{F}_q)$ does not belong to R, then $N(\varpi) = \#\{(u, v) \in (\mathbb{F}_q)^2 \mid f(u) + f(v) = \varpi\}$ satisfies

$$\left|q - N(\varpi)\right| \leq 98\sqrt{q} + 23 \leq 2^7 \cdot \sqrt{q}$$

as required. And obviously, there are at most $12^2 = 144$ points in R.

It remains to bound $N(\varpi)$ for an $\mathbb{F}_q$-point $\varpi \in R \cap E(\mathbb{F}_q)$. To do so, consider again $Z = s^{-1}(\varpi)$ the fiber at such a point, and $E' \subset E \times E$ the image of Z under $h \times h$ (or equivalently, the fiber of the group law of E at ϖ). The morphism $Z \to E'$ is of degree 16, so each point has at most 16 pre-images. Hence

$$N(\varpi) \leq 16 \cdot \#E'(\mathbb{F}_q) \leq 16\big(q + 1 + 2\sqrt{q}\big) \leq 2^5 \cdot q$$

since $q \geq 5$. This concludes the proof.

A.3 Generalization to Even Characteristic

The previous technique carries over to Icart's function in characteristic 2 easily (see Appendix D for the definition of Icart's function in characteristic 2). In this case, if E is the elliptic curve $y^2 + xy = x^3 + ax^2 + b$ over a field $\mathbb{F}_q$ of characteristic 2, the curve $C \subset \mathbb{P}^2 \times E$ defining Icart's correspondence has the equation

$$u^4 + u^2 + xu + y + a^2 = 0$$

It is smooth except at the point (∞, O) which blows up into 4 regular points in the normalization $\widetilde{C}$. The second projection $h : \widetilde{C} \to E$ is then of degree 4 and only ramified over the single point ρ of E such that $x = 0$, with ramification type $(2, 2)$. In particular, $\widetilde{C}$ is a smooth curve of genus 2.

We can then consider the map $s : S = \widetilde{C} \times \widetilde{C} \to E$ again, and find that the fiber $Z = s^{-1}(\varpi)$ at any point $\varpi \in E - \{2\rho\}$ is a smooth geometrically integral curve. The morphism of Z to its image E' in $E \times E$ is of degree 16 and only ramified over $(\rho, \varpi - \rho)$ and $(\varpi - \rho, \rho)$, with ramification type $(2, 2, 2, 2, 2, 2, 2, 2)$. Thus, Z is of genus 9.

Using the notations from the proof of Lemma 9, the first projection $p : S \to \mathbb{P}^1 \times \mathbb{P}^1$ still induces a bijection $U(\mathbb{F}_q) \to \mathbb{A}^2(\mathbb{F}_q)$ on $\mathbb{F}_q$-points when $q \equiv 2 \pmod 3$ and $U = p^{-1}(\mathbb{A}^2)$. Moreover, if D denotes the complementary divisor on S, we can compute $Z \cdot D = n_\infty Z \cdot (l + m) = 4Z \cdot (l + m)$, while $K \equiv (2g_{\widetilde{C}} - 2)(l + m) = 2(l + m)$. Thus, the adjunction formula gives $2g_Z - 2 = Z \cdot K + Z^2 = \frac{1}{2} Z \cdot D$. Hence:

$$\#(Z \cap D)(\mathbb{F}_q) \le Z \cdot D = 2 \cdot (2g_Z - 2) = 32$$

Therefore, for any point $\varpi \in E(\mathbb{F}_q) - \{2\rho\}$, we get

$$q - 4\sqrt{q} - 31 \le N(\varpi) \le q + 4\sqrt{q} + 1$$

Furthermore, we still have $N(2\rho) \le 2^5 \cdot q$ as in the previous section.

B Proof of Lemma 4

We prove a slightly more general lemma:

Lemma 10. *Let $f : S \to R$ be a polynomially computable function. Let $B = \max\{\#f^{-1}(r) \mid r \in R\}$. Assume that there exists a polynomial-time algorithm Inv that for any $r \in R$ outputs the set $f^{-1}(r)$. Then f is an α-weak encoding, with $\alpha = B \cdot \#R/\#S$. In particular, Icart's function $f_{a,b}$ is an α-weak encoding from $\mathbb{F}_q$ to $E_{a,b}(\mathbb{F}_q)$ with $\alpha = 4 \cdot \#E_{a,b}(\mathbb{F}_q)/q$.*

Proof. We have $\Pr_s[f(s) = r] = \#f^{-1}(r)/\#S \le B/\#S = \alpha/\#R$ by taking $\alpha := B \cdot \#R/\#S$; therefore, the distribution of $f(s)$ for uniform $s \in S$ is α-bounded in R. Given $\mathsf{Inv}(r) = f^{-1}(r)$, the algorithm $\mathcal{I}(r)$ simply generates a random element in the set $f^{-1}(r)$, and lets $N_r = \#f^{-1}(r)$. The result for Icart's function follows from Lemma 2. □

C Composition Lemmas

C.1 Generalized Admissible Encodings

The Propositions of Sections 6.1, 6.2, 6.3 and 6.4 fit in a common framework: they assert that some function is admissible, and that we still get indifferentiable hashing when composing them with one of our constructions.

It is not quite correct that composing two admissible encodings yields another admissible encoding. To circumvent this problem, we introduce the slightly more general notion of *generalized admissible encoding*. We show that admissible encodings are generalized admissible encoding; that generalized admissible encodings are sufficient for indifferentiability; and that the composition of two admissible encodings is again a generalized admissible encoding.

Definition 6 (Generalized Admissible Encoding). *A function $F : S \to R$ is said to be an ε-generalized admissible encoding if it satisfies the following properties:*

1. *Computable: F is computable in deterministic polynomial time;*
2. *Invertible: there exists a probabilistic polynomial time algorithm $\mathcal{I}_F$ such that $\mathcal{I}_F(r) \in F^{-1}(r) \cup \{\perp\}$ for all $r \in R$, and the distribution of $\mathcal{I}_F(r)$ is ε-statistically indistinguishable from the uniform distribution in S when r is uniformly distributed in R.*

F is an generalized admissible encoding if ε is a negligible function of the security parameter.

From Lemma 3 we have that an ε-admissible encoding is also a 2ε-generalized admissible encoding. The next lemma says that Definition 6 is sufficient for obtaining the indifferentiability property; it follows immediately from our proof of Theorem 1.

Lemma 11. *Let $F : S \to R$ be an ε-generalized admissible encoding. The construction $H(m) = F(h(m))$ is $(t_D, t_S, q, \varepsilon')$-indifferentiable from a random oracle, in the random oracle model for $h : \{0,1\}^* \to S$, with $\varepsilon' = 2q\varepsilon$ and $t_S = 2q \cdot t_I$, where t_I is the maximum running time of F's sampling algorithm.*

Finally we prove that the composition of two generalized admissible encoding remains a generalized admissible encoding.

Lemma 12. *Let $F : S \to R$ be an ε_1-generalized admissible encoding and $G : R \to T$ be an ε_2-generalized admissible encoding. Then $G \circ F$ is an $(\varepsilon_1 + \varepsilon_2)$-generalized admissible encoding from S to T.*

Proof. Firstly, $G \circ F$ is computable in polynomial time. Secondly, given t uniformly distributed in T, the random variable $r = \mathcal{I}_G(t)$ is ε_2-statistically indistinguishable from the uniform distribution in R. Then $s = \mathcal{I}_F(r)$ is $(\varepsilon_1 + \varepsilon_2)$-statistically indistinguishable from the uniform distribution in S. □

C.2 Proof of Proposition 3

Proposition 3 is then an immediate consequence of Lemma 12 and of the following result.

Lemma 13. *The map $M_\ell : E(\mathbb{F}_q) \to \mathbb{G}$, $P \mapsto \ell P$ is an ε-admissible encoding, with $\varepsilon = 0$.*

Proof. The proof is the same as the proof of Lemma 5.1 in [6]. Since M_ℓ is a group homomorphism, the distribution of $M_\ell(P)$ is uniform in $\mathbb{G}$ for uniform $P \in E$.

The group $E(\mathbb{F}_q)$ is isomorphic to $\mathbb{Z}_{\ell_1 N'} \times \mathbb{Z}_{\ell_2}$ for some ℓ_1, ℓ_2 such that $\ell = \ell_1\ell_2$. Let G_1, G_2 be the points corresponding to $(1,0)$ and $(0,1)$ under this isomorphism. Given $Q \in \mathbb{G}$, the sampling algorithm picks $u \in \mathbb{Z}_{\ell_1}$ and $v \in \mathbb{Z}_{\ell_2}$ uniformly at random, and returns $P = (1/\ell) \cdot Q + u \cdot N' \cdot G_1 + v \cdot G_2$. Here $1/\ell$ is computed in $(\mathbb{Z}_{N'})^*$. We have that $\ell P = Q$ as required and P is uniformly distributed in $M_\ell^{-1}(Q)$. □

C.3 Proof of Proposition 4

Let p be an integer. We first show that reduction modulo p is an admissible encoding from $\{0,1\}^\ell$ to $\mathbb{Z}_p$ if $2^\ell \gg p$.

Lemma 14 **($\{0,1\}^\ell \to \mathbb{Z}_p$).** *Let p be an integer and k be a security parameter. Let $\ell = k + \lceil \log_2 p \rceil + 1$. The function $\textsc{Mod}_p : [0, 2^\ell - 1] \to \mathbb{Z}_p$ with $\textsc{Mod}_p(x) = x \bmod p$ is a 2^{-k}-generalized admissible encoding.*

Proof. Let $\mu \in \mathbb{Z}$ such that $2^\ell - p < \mu \cdot p \leq 2^\ell$. We consider the sequence:

$$\{0,1\}^\ell \xrightarrow{F} [0, \mu \cdot p[\xrightarrow{G} \mathbb{Z}_p$$

where $F(x) = x \bmod (\mu \cdot p)$ and $G(y) = y \bmod p$. We show that both F and G are generalized admissible encodings; therefore from Lemma 12 the composition of F and G remains a generalized admissible encoding. For F we actually prove a slightly more general result:

Lemma 15. *Let $F : S \to (S \cup \Delta_2) \setminus \Delta_3$ be a polynomially computable function such that $F(x) = x$ for all $x \in S \setminus \Delta_1$. Assume that set membership for $S \setminus \Delta_1$ can be decided in polynomial time. Then F is an ε-generalized admissible encoding, with $\varepsilon = (\#\Delta_1 + \#\Delta_2 + \#\Delta_3)/\#S$.*

Proof. Given $x \in S \cup \Delta_2$, the sampling algorithm $\mathcal{I}_F(x)$ returns x for $x \in S \setminus \Delta_1$ and $\perp$ otherwise. Therefore for uniform $x \in (S \cup \Delta_2) \setminus \Delta_3$ the distribution of $\mathcal{I}_F(x)$ is ε-indistinguishable from the uniform distribution in S, with $\varepsilon = (\#\Delta_1 + \#\Delta_2 + \#\Delta_3)/\#S$. □

Applying Lemma 15 with $S = \{0,1\}^\ell$, $\Delta_1 = [\mu \cdot p, 2^\ell - 1[$, $\Delta_2 = \emptyset$ and $\Delta_3 = \Delta_1$, we obtain that F is an ε-generalized admissible encoding, with $\varepsilon = 2p/2^\ell \leq 2^{-k}$. Moreover, it is easy to see that G is an admissible encoding according to Definition 1; therefore it is an ε-generalized admissible encoding with $\varepsilon = 0$. From Lemma 12 the composition of two generalized admissible encodings remains a generalized admissible encoding. This concludes the proof of Lemma 14. □

We now proceed with the proof of Proposition 4. With p a $2k$-bit prime and $\ell = 3k$, from Lemma 14 we obtain that reduction mod p is a 2^{-k+1}-admissible encoding from $\{0,1\}^\ell$ to $\mathbb{Z}_p$. Using Lemma 12, this shows that $F : \{0,1\}^\ell \times \{0,1\}^\ell \to E(\mathbb{F}_p)$ with:

$$F(u,v) = f(u \bmod p) + f(v \bmod p)$$

is an ε-generalized admissible encoding with $\varepsilon = 2^9 \cdot 2^{-k} + 2^{-k+1} \leq 2^{10} \cdot 2^{-k}$. Applying Lemma 11, this proves Proposition 4.

C.4 Proof of Proposition 5

We consider the function $G : \{0,1\}^\ell \mapsto \mathbb{Z}_p$ with $G(u) = u \mod p$. Since $p = 2^\ell - \omega$, applying Lemma 15 with $S = \{0,1\}^\ell$, $\Delta_1 = [p, 2^\ell[$, $\Delta_2 = \emptyset$ and $\Delta_3 = \Delta_1$, we obtain that G is an ε-generalized admissible encoding with $\varepsilon = 2\omega \cdot 2^{-\ell}$.

Using Lemma 12, this shows that $F : \{0,1\}^\ell \times \{0,1\}^\ell \to E(\mathbb{F}_p)$ with:

$$F(u,v) = f(u \bmod p) + f(v \bmod p)$$

is an ε-generalized admissible encoding with $\varepsilon = 2^9 \cdot 2^{-\ell/2} + 2\omega \cdot 2^{-\ell}$. Applying Lemma 11, this proves Proposition 5.

D Icart's Function in Characteristic 2

In characteristic 2 we consider an elliptic curve defined by the following equation:

$$E_{a,b} : Y^2 + XY = X^3 + aX^2 + b$$

where a and b are elements of $\mathbb{F}_{2^n}$. When n is odd, we have $2^n - 1 \not\equiv 0 \pmod 3$, which implies that the map $x \mapsto x^3$ is a bijection over $\mathbb{F}_{2^n}$. Let

$$\begin{aligned} f_{a,b} : \mathbb{F}_{2^n} &\to (\mathbb{F}_{2^n})^2 \\ u &\mapsto (x, y) \end{aligned}$$

where

$$\begin{aligned} x &= (v^4 + v^3 + b)^{1/3} + v \\ y &= ux + v^2 \\ v &= a + u + u^2 \end{aligned}$$

One can check that for any $u \in \mathbb{F}_{2^n}$, $f_{a,b}(u)$ is indeed a point of $E_{a,b}$; we refer to [23] for more details. Icart's function in characteristic 2 satisfies the same property as in characteristic $p > 3$:

Lemma 16 (Icart [23]). *The function $f_{a,b}$ is computable in deterministic polynomial time. For any point $\varpi \in f_{a,b}(\mathbb{F}_{2^n})$, the set $f_{a,b}^{-1}(\varpi)$ is computable in polynomial time and $\# f_{a,b}^{-1}(\varpi) \leq 4$.*

E Shallue-Woestijne in Characteristic 2

In this section, we recall the Shallue-Woestijne algorithm in characteristic 2 (see [31]). An elliptic curve over a field $\mathbb{F}_{2^n}$ is a set of points $(x, y) \in (\mathbb{F}_{2^n})^2$ verifying the equation:

$$E_{a,b} : Y^2 + X \cdot Y = X^3 + a \cdot X^2 + b$$

where $a, b \in \mathbb{F}_{2^n}$. Let g be the rational function

$$g : x \mapsto x^{-2} \cdot \left(x^3 + a \cdot x^2 + b\right)$$

Letting $Z = Y/X$, the equation for $E_{a,b}$ can be rewritten as:

$$Z^2 + Z = g(X) \tag{15}$$

Theorem 4 (Shallue-Woestijne [31]). *Let $g(x) = x^{-2} \cdot \left(x^3 + a \cdot x^2 + b\right)$ where $a, b \in \mathbb{F}_{2^n}$. Let*

$$X_1(t, w) = \frac{t \cdot c}{1 + t + t^2} \qquad X_2(t, w) = t \cdot X_1(t, w) + c \qquad X_3(t, w) = \frac{X_1(t, w) \cdot X_2(t, w)}{X_1(t, w) + X_2(t, w)}$$

where $c = a + \omega + \omega^2$. Then $g(X_1(t, w)) + g(X_2(t, w)) + g(X_3(t, w)) \in h(\mathbb{F}_{2^n})$ where h is the map $h : z \mapsto z^2 + z$.

From Theorem 4, we have that at least one of the $g(X_i(t, w))$ must be in $h(\mathbb{F}_{2^n})$, which leads to a point in $E_{a,b}$. Namely, we have that $h(\mathbb{F}_{2^n}) = \{z \in \mathbb{F}_{2^n} \mid \mathrm{Tr}\,(z) = 0\}$, where Tr is the trace operator $\mathrm{Tr} : \mathbb{F}_{2^n} \to \mathbb{F}_2$ with:

$$\mathrm{Tr}\,(z) = \sum_{i=0}^{n-1} z^{2^i}$$

From Theorem 4 we have $\sum_i \mathrm{Tr}\,(g(X_i)) = 0$ and therefore at least one of the X_i must satisfy $\mathrm{Tr}\,(g(X_i)) = 0$.

Given such a X_i it remains to compute a solution of the equation $Z^2 + Z = g(X_i)$. This is a linear equation in $\mathbb{F}_{2^n}$, so finding Z amounts to solving a linear system. When n is odd, the solutions can be found more efficiently: the solutions of $Z^2 + Z = \beta$ are given by $\mathrm{HTr}(\beta)$ and $\mathrm{HTr}(\beta) + 1$, where HTr is the half trace, defined as:

$$\mathrm{HTr} : z \mapsto \sum_{i=0}^{(n-1)/2} z^{2^{2i}}$$

We note that in practice a prime extension degree n is generally used, which means that n is usually odd. We obtain the following pseudo-code for the Shallue-Woestijne algorithm in characteristic 2, when the extension degree n is odd. Note that the algorithm takes as input two values $t, w \in \mathbb{F}_{2^n}$; in practice, one can take $w = 0$.

```
Shallue-Woestijne algorithm in characteristic 2:
Input: parameters $a, b \in \mathbb{F}_{2^n}$ and input $t, w \in \mathbb{F}_{2^n}$
Output : $(x, y) \in E_{a,b}$
 1. $c \leftarrow a + w + w^2$
 2. $X_1 \leftarrow t \cdot c/(1 + t + t^2)$
 3. $X_2 \leftarrow t \cdot X_1 + c$
 4. $X_3 \leftarrow X_1 \cdot X_2/(X_1 + X_2)$
 5. For $i = 1$ to 3:
    (a) $h_i \leftarrow ({X_i}^3 + a \cdot {X_i}^2 + b)/{X_i}^2$
    (b) If $\mathrm{Tr}\,(h_i) = 0$, return $(X_i,\ \mathrm{HTr}(h_i) \cdot X_i)$
```

E.1 Analysis

As for the Ulas maps, two parameters are needed to generate a point. A general way to analyze the algorithm is to fix one of the parameter and to let the other vary.

Lemma 17 (fixed t). *The Shallue-Woestijne algorithm with fixed t such that $t^2+t \neq 1$ has pre-image size at most 6. The encoding can be inverted on its image in polynomial time.*

Lemma 18 (fixed w). *The Shallue-Woestijne algorithm with fixed w such that $w^2 + w \neq a$ has pre-image size at most 6. The encoding can be inverted on its image in polynomial time.*

Proof. We first give the expression of the rational function $X_3(t, w)$:

$$X_3(t, w) = \frac{c \cdot (t^2 + t)}{1 + t + t^2}$$

Since we have for all i, $\deg_t(X_i(t, w)) = 2$ and $\deg_w(X_i(t, w)) = 2$, it is easy to see that each point has at most 6 pre-images in both cases. □

F Analysis of the Algorithm from Section 8.3

We consider the elliptic curve equation $y^2 = x^3 + ax^2 + b$ which we rewrite $x^3 + ax^2 + (b - y^2) = 0$. Letting $\alpha = 1/x$, we get:

$$\frac{1}{\alpha^3} + \frac{a}{\alpha^2} + (b - y^2) = 0$$

Multiplying by $\alpha^3/(b-y^2)$, this gives:

$$\alpha^3 + \frac{a}{b-y^2} \cdot \alpha = -1/(b-y^2) \tag{16}$$

Given $k \in \mathbb{F}$ we consider the function $f(\alpha) = \alpha^3 + k \cdot \alpha$. In char 3 this is a linear function. We have:

$$f(\alpha) = 0 \Leftrightarrow \alpha = 0 \text{ or } \alpha^2 = -k$$

Therefore f is bijective if and only if $-k \notin Q$. When f is bijective its inverse can be computed in deterministic polynomial time by solving a linear system.

Since $k = a/(b-y^2)$ in equation (16), we must have $-a/(b-y^2) \notin Q$ so that equation (16) has a unique solution. This is equivalent to $-(b-y^2)/a \notin Q$ or $-(b-y^2)/a = \eta \cdot z^2$ for some fixed $\eta \notin Q$. This gives:

$$a\eta z^2 - y^2 + b = 0$$

which is the equation of a conic which is easy to parameterize. Such parameterization is computed at steps 1 and 2 of the algorithm in Section 8.3.

G Implementation of the Simplified SWU Algorithm

In the following we provide an implementation our simplified SWU algorithm from Section 7, using the Sagemath library [33].

```
def genParams(k=160):
  p=1
  while (p % 4)==1:
    p=random_prime(2^k)
  F=GF(p)
  a=F.random_element()
  b=F.random_element()
  return (p,a,b)

def simpleSWU(p,a,b,t):
  alpha=-t^2
  x2=-b/a*(1+1/(alpha^2+alpha))
  x3=alpha*x2
  h2=x2^3+a*x2+b
  h3=x3^3+a*x3+b
  if is_square(h2):
    return (x2,h2^((p+1)//4))
  else:
    return (x3,h3^((p+1)//4))

def testSimpleSWU():
  p,a,b=genParams()
  t=GF(p).random_element()
  x,y=simpleSWU(p,a,b,t)
  print "p=",p,"\na=",a,"\nb=",b,"\nx=",x,"\ny=",y
  print "y^2==x^3+ax+b:",y^2==x^3+a*x+b
```

H Implementation of Hash Algorithms in Characteristic 3

In the following we provide an implementation of the 3 hash algorithms in characteristic 3 from Section 8, using the Sagemath library [33].

```
def genNonSquare(f):
  while True:
    x=f.random_element()
    if not is_square(x):
      return x

def genParamsInQ(n=53):
  f=GF(3^n,’t’)
  a=f.random_element()
  c=f.random_element()
  b=-a*c^2
  eta=genNonSquare(f)
  return (f,a,b,c,eta)

def algo1(params,t):
  f,a,b,c,eta=params
  x=c*(1-1/(eta*t^2))
  if is_square(x^3+a*x^2+b):
    return (x,(x^3+a*x^2+b).sqrt())
  else:
    xp=eta*t^2*x
    return (xp,(xp^3+a*xp^2+b).sqrt())

def genParamsNotInQ(n=53):
  f=GF(3^n,’t’)
  Delta=genNonSquare(f)
  a=f.random_element()
  b=-Delta/a^3
  x=PolynomialRing(f,’x’).gen()
  x0=(x^3+a*x^2+b).roots()[0][0]
  eta=genNonSquare(f)
  return (f,a,b,x0,eta)

def algo2(params,t):
  f,a,b,x0,eta=params
  x=-2*x0*(1+1/(eta*t^2))
  x1=x+x0
  x2=eta*t^2*x+x0
  if is_square(x1^3+a*x1^2+b):
    return (x1,(x1^3+a*x1^2+b).sqrt())
  else:
     return (x2,(x2^3+a*x2^2+b).sqrt())

def genParamsAlgo3(n=53):
  f=GF(3^n,'t')
  a=f.random_element()
  b=f.random_element()
  eta=genNonSquare(f)
  while True:
    z0=f.random_element()
    sy0=a*eta*z0^2+b
    if is_square(sy0):
      return (f,a,b,eta,sy0.sqrt(),z0)

def algo3(params,t):
  f,a,b,eta,y0,z0=params
  z=(-z0*t^2+2*y0*t-a*eta*z0)/(a*eta-t^2)
  y=y0+t*(z-z0)
  k=a/(b-y^2)
  av=PolynomialRing(f,'x').gen()
  alpha=(av^3+k*av+k/a).roots()[0][0]
  return (1/alpha,y)

def testAlgo(algo,params,f,a,b):
  t=f.random_element()
  x,y=algo(params,t)
  print "x=",x
  print "y=",y
  print "Point in curve:",y^2==x^3+a*x^2+b

def testAlgo1(n=53):
  params=genParamsInQ(n)
  f,a,b,c,eta=params
  testAlgo(algo1,params,f,a,b)

def testAlgo2(n=53):
  params=genParamsNotInQ(n)
  f,a,b,x0,eta=params
  testAlgo(algo2,params,f,a,b)

def testAlgo3(n=53):
  params=genParamsAlgo3(n)
  f,a,b,eta,y0,z0=params
  testAlgo(algo3,params,f,a,b)
```