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 [4], 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 [1,17,18], signature and identity-based signature schemes [3,5,6,12,27] and identity-based signcryption schemes [8,21].

Hashing into elliptic curves is also required for some passwords based authentication protocols, for instance the SPEKE (Simple Password Exponential Key Exchange) [20] and the PAK (Password Authenticated Key exchange) [9], and also for discrete-log based signature schemes such as [13] 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

* Work done while working for SAGEM company.

remains to determine which hashing algorithm should be used, and whether it is reasonable to see it as a random oracle.

In [4], 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 [9,13,20]), it is more efficient to use ordinary elliptic curves, since supersingular curves require much larger security parameters (due to the MOV attack [23]).

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 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 [7]). 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 [25]. The algorithm has running time $\mathcal{O}(\log^4 p)$ for any p, and $\mathcal{O}(\log^3 p)$ when $p \equiv 3 \pmod 4$. The rational maps in [25] were later simplified and generalized to hyper-elliptic curves by Ulas in [26]; 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 [19]. 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 [19], 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 [1,3,4,5,6,8,9,12,17,18,20,21,27]. In the random oracle model [2], 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 [11]). 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 [9,20]), 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 $\simeq 5/8$ of the points [15,16]; moreover it is easy to see that the distribution of $f(h(m))$ is not uniform in $\mathsf{Im}f$. 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 5.1).

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.* [22]. As shown in [14], 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 the subgroup of an elliptic curve (with cyclic or non-cyclic group) 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 algorithm into elliptic curves in characteristic 3. We summarize in Table 1 the known hashing algorithms into ordinary elliptic curves.

2 Preliminaries

2.1 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 [19] as the map $f_{a,b} : \mathbb{F}_q \to E(\mathbb{F}_q)$ such that $f_{a,b}(u) = (x, y)$ where:

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

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 1 (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 $q/4 < \#f_{a,b}(\mathbb{F}_q) < q$.*

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

char(K)	normal form	discriminant Δ	encoding	condition
$\neq 2, 3$	$y^2 = x^3 + ax + b$	$-16(4a^3 + 27b^2)$	Icart [19]	$p \equiv 2 \pmod 3$
			SW [25]	–
			SWU [26]	–
			SWU, Sec. 7	$p \equiv 3 \pmod 4$
2	$y^2 + xy = x^3 + ax^2 + b$	b	Icart [19]	odd n
			SW [25]	–
3	$y^2 = x^3 + ax^2 + b$	$-a^3 b$	Sec. 8.1	$\Delta \in Q$
			Sec. 8.2	$\Delta \notin Q$
			Sec. 8.3	–

Table 1. Known deterministic hashing algorithms into ordinary elliptic curves with discriminant $\Delta \neq 0$. 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 [22].

Definition 1 (Indifferentiability [22]). *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.

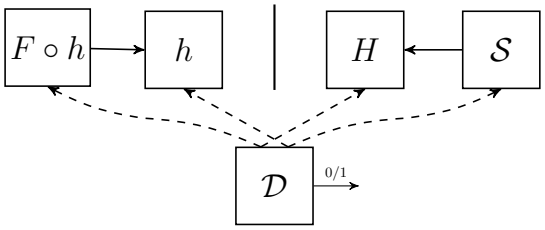

Fig. 1. The indifferentiability notion, illustrated with construction $C^h = F \circ h$ for some function F, and random oracles h and H.

It is shown in [22] 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 [22] or [14] for a proof.

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*. It can be seen as a generalization of the definition used in [4].

Definition 2 (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.

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:

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

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. The proof of the following Lemma is given in the full version of the paper [10]:

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

This implies 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 $\mathsf{Im}f$ since the distribution of $f(u)$ is not uniform in $\mathsf{Im}f$ for uniform $u \in \mathbb{F}_q$).

In this section, we describe a different construction which is almost as efficient. Namely we prove that if $h_1, h_2 : \{0,1\}^* \to \mathbb{F}_q$ are two hash functions in the random oracle model, then the hash function $H : \{0,1\}^* \to E(\mathbb{F}_q)$ defined by

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

is indifferentiable from a random oracle into the elliptic curve.

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 [7]), 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:

$$F(u, v) = f(u) + f(v)$$

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 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.

In the full version of this paper, we show that Proposition 1 directly implies Criterion 2, and that Criterion 3 easily follows from the point counting of [15,16]. Additionally, we prove that F is also an admissible encoding when using Icart's function f in characteristic 2.

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 3 (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:

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

From inequality (1) we have that any invertible function with bounded preimage and bounded $\#R/\#S$ is a weak encoding; in particular, this is the case for Icart's function (the proof is given in the full version of the paper [10]).

Lemma 3. *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 full version of this paper [10]. 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, this is the case when f is Icart's function. 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 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 Extensions

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:

$$H(m) := \ell\big(f_{a,b}(h_1(m)) + f_{a,b}(h_2(m))\big) \tag{2}$$

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

Proposition 2. *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 the full version of the paper [10] 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 3. *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 the full version of the paper [10] for the proof.

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 $\mathbb{F}_q$, for $q \equiv 3 \pmod 4$. Note that this condition is usually satisfied in practice, since it enables to compute square roots efficiently.

Proposition 4 (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:[6]

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

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 (3) 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 4. □

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 $(X_3, h_3^{(q+1)/4})$

In the full version of the paper [10] we show that our simplified SWU algorithm is a weak encoding into the curve. Therefore it can be used with the general construction from Section 5. An implementation is also provided in the full version of the paper [10].

[6] A similar equation was used in [24] 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.

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[7] 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 the full version of the paper [10].

8.1 Algorithm for $\Delta \in Q$

Proposition 5. *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.

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

$$g(u \cdot x) = u^3 \cdot g(x) \tag{4}$$

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 $\Delta = -a^3 b \in Q$, 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 (4):

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

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 5. Moreover from equation (4) since $u^3 \notin Q$ either $g(x)$ or $g(u \cdot x)$ must be a quadratic residue. □

From Proposition 5 we easily deduce a deterministic encoding algorithm.

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

8.2 Algorithm for $\Delta \notin Q$

Proposition 6. *Let $\mathbb{F}$ be a field of characteristic 3 and $g(x) = x^3 + ax^2 + b$ with $\Delta = -a^3 b \notin Q$. 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 [26]. 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 $X_2(t)$. □

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 $z = (-z_0t^2 + 2y_0t - a\eta z_0)/(a\eta - t^2)$
2. Let $y = y_0 + t \cdot (z - z_0)$
3. Let $k = a/(b - y^2)$
4. Find the unique solution α of the linear system $\alpha^3 + k \cdot \alpha = -k/a$
5. Let $x = 1/\alpha$ and output (x, y)

We show in Appendix B that this also defines a deterministic encoding into elliptic curves.

Acknowledgments

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

References

1. J. Baek and Y. Zheng. Identity-based threshold decryption. In *Public Key Cryptography*, pages 262–276, 2004.
2. M. Bellare and P. Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *ACM Conference on Computer and Communications Security*, pages 62–73, 1993.
3. A. Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In *Public Key Cryptography*, pages 31–46, 2003.
4. D. Boneh and M. K. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO*, pages 213–229, 2001.
5. D. Boneh, C. Gentry, B. Lynn, and H. Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *EUROCRYPT*, pages 416–432, 2003.
6. D. Boneh, B. Lynn, and H. Shacham. Short signatures from the weil pairing. In *ASIACRYPT*, pages 514–532, 2001.
7. C. Boyd, P. Montague, and K. Q. Nguyen. Elliptic curve based password authenticated key exchange protocols. In *ACISP*, pages 487–501, 2001.
8. X. Boyen. Multipurpose identity-based signcryption (a swiss army knife for identity-based cryptography). In *CRYPTO*, pages 383–399, 2003.
9. V. Boyko, P. D. MacKenzie, and S. Patel. Provably secure password-authenticated key exchange using diffie-hellman. In *EUROCRYPT*, pages 156–171, 2000.
10. E. Brier, J.-S. Coron, T. Icart, D. Madore, H. Randriam, and M. Tibouchi. Efficient indifferentiable hashing into ordinary elliptic curves. Cryptology ePrint Archive, Report 2009/340, 2009. `http://eprint.iacr.org/`. Full version of this paper.
11. R. Canetti, O. Goldreich, and S. Halevi. The random oracle methodology, revisited. *J. ACM*, 51(4):557–594, 2004.
12. J. C. Cha and J. H. Cheon. An identity-based signature from gap diffie-hellman groups. In *Public Key Cryptography*, pages 18–30, 2003.
13. B. Chevallier-Mames. An efficient cdh-based signature scheme with a tight security reduction. In *CRYPTO*, pages 511–526, 2005.
14. J.-S. Coron, Y. Dodis, C. Malinaud, and P. Puniya. Merkle-damgård revisited: How to construct a hash function. In *CRYPTO*, pages 430–448, 2005.
15. R. R. Farashahi, I. E. Shparlinski, and J. F. Voloch. On hashing into elliptic curves, 2010. preprint available from `http://www.ma.utexas.edu/users/voloch/preprint.html`.
16. P.-A. Fouque and M. 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/`.
17. C. Gentry and A. Silverberg. Hierarchical id-based cryptography. In *ASIACRYPT*, pages 548–566, 2002.
18. J. Horwitz and B. Lynn. Toward hierarchical identity-based encryption. In *EUROCRYPT*, pages 466–481, 2002.

19. T. Icart. How to hash into elliptic curves. In S. Halevi, editor, *CRYPTO*, volume 5677 of *Lecture Notes in Computer Science*, pages 303–316. Springer, 2009.
20. D. P. Jablon. Strong password-only authenticated key exchange. *SIGCOMM Comput. Commun. Rev.*, 26(5):5–26, 1996.
21. B. Libert and J.-J. Quisquater. Efficient signcryption with key privacy from gap diffie-hellman groups. In *Public Key Cryptography*, pages 187–200, 2004.
22. U. M. Maurer, R. Renner, and C. Holenstein. Indifferentiability, impossibility results on reductions, and applications to the random oracle methodology. In *TCC*, pages 21–39, 2004.
23. A. Menezes, T. Okamoto, and S. A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Transactions on Information Theory*, 39(5):1639–1646, 1993.
24. J.-F. Mestre. Rang de courbe elliptiques d'invariant donné. *Comptes rendus de l'Académie des sciences. Série 1, Mathématique*, 314(12):297–319, 1992.
25. A. Shallue and C. van de Woestijne. Construction of rational points on elliptic curves over finite fields. In *ANTS*, pages 510–524, 2006.
26. M. Ulas. Rational points on certain hyperelliptic curves over finite fields. *Bull. Polish Acad. Sci. Math.*, 55(2):97–104, 2007.
27. F. Zhang and K. Kim. Id-based blind signature and ring signature from pairings. In *ASIACRYPT*, pages 533–547, 2002.

A Proof of Proposition 1

This appendix gives a proof of Proposition 1. For the sake of brevity, the proofs of some technical lemmas are omitted in this extended abstract, and can be found in the full version [10].

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

$$u^4 - 6xu^2 + 6yu - 3a = 0 \tag{5}$$

In other words, Icart's function is the algebraic correspondence between $\mathbb{P}^1$ and E given by (5).

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 4. *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.*

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 $\rho_1, \ldots, \rho_{12}$ 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 5. *The fibers of s at all geometric points of E_0 are smooth connected curves on $S_{\bar{\mathbb{F}}_q}$ of genus 49.*

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 6. *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*

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

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

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

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.

B 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:

$$\alpha^3 + \frac{a}{b - y^2} \cdot \alpha = -1/(b - y^2) \tag{6}$$

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 (6), we must have $-a/(b - y^2) \notin Q$ so that equation (6) 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.