A Random Oracle into Elliptic Curves

Jean-Sébastien Coron[1] and Thomas Icart[1,2]

[1]Université du Luxembourg
[2]Sagem Sécurité

Abstract. We provide the first construction of a hash function into an elliptic curve that is indifferentiable from a random oracle. Our construction is quite efficient; it is based on Icart's algorithm for hashing into elliptic curves in deterministic polynomial time.

1 Introduction

Some elliptic-curve cryptosystems require to hash into an elliptic curve, for instance the Boneh-Franklin identity based encryption scheme [2], in which the public-key for identity $id \in \{0,1\}^*$ is a point $Q_{id} = H_1(id)$ on the curve. Hashing into elliptic curves is also required for some passwords based authentication protocols, for instance the SPEKE (Simple Password Exponential Key Exchange) [5] and the PAK (Password Authenticated Key exchange) [3]. In those three 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 [2], Boneh and Franklin use a particular super-singular 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. This enables to hash using $f(h(m))$ where h is a classical hash function from $\{0,1\}^*$ to $\mathbb{F}_p$. The authors show that their IBE scheme is also secure when h is seen as a random oracle into $\mathbb{F}_p$. However, when no pairing operation is required (as in [3] and [5]), it is more efficient to use ordinary elliptic-curves, since super-singular curves require much larger security parameters (due to the MOV attack [8]).

A deterministic hash algorithm for any elliptic curve was recently published by Icart [4]. The algorithm is very efficient, faster than a scalar multiplication into the curve. Given any elliptic-curve E defined over $\mathbb{F}_p$, Icart actually defines a function f that is a rational function from $\mathbb{F}_p$ into the curve. Then given any hash function h into $\mathbb{F}_p$, one can use $H(m) = f(h(m))$ as a hash function into E. As shown in [4], H is one-way if h is one-way.

Therefore, one possibility could be to use $H(m) = f(h(m))$ in cryptosystems such as [3] and [5], and then assume that H behaves as a random oracle. However, one can easily see that this is not a reasonable assumption; namely Icart's function f does not generate all the elliptic curve points; only a fraction roughly $5/8$ of them are covered; consequently even if we see the underlying function h as a random oracle, the resulting hash function H does not behave as a random oracle. Therefore in this paper we would like to construct a hash function H into elliptic curves that behaves as a random oracle when h is seen as a random oracle, and H should work for any elliptic-curve, not only super-singular ones.

In this paper, we provide the first hash function construction satisfying this property. We use the indifferentiability framework of Maurer *et al.* [7] to show that any cryptosystem using our construction remains secure when the underlying hash function is seen as a random oracle. For this we introduce the notion of *admissible encoding*. Roughly speaking, an admissible encoding is a function that can be efficiently inverted with (almost) uniformly distributed inputs from uniformly distributed outputs. We show that if $f : A \to B$ is an admissible encoding, then $H(m) = f(h(m))$ is indifferentiable from a random oracle into B when $h : \{0,1\}^* \to A$ is seen as a random oracle.

However, we cannot apply this result to Icart's function directly, since Icart's function is *not* an admissible encoding; this is because as mentioned previously the output of Icart's function only covers a fraction of the elliptic curve points. Therefore, we introduce a weaker notion which we call *weak encoding*. Informally, a weak encoding $f : A \to B$ must be efficiently invertible with (almost) uniformly distributed inputs from uniformly distributed outputs, but the inverting algorithm is only required to work with non-negligible probability (over $b \in B$ and its own random coins), instead of probability $\simeq 1$ as for admissible encodings. In this paper we show that 1) Icart's function satisfies this notion of weak encoding, and 2) we can construct an admissible encoding from a weak encoding when working in a group. This enables to use Icart's function to build a hash function that is indifferentiable from a random oracle into the elliptic curve.

More precisely, given an elliptic-curve $\mathbb{E}$ defined over $\mathbb{F}_p$ with N points and generator G, our 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 Icart's function (or more generally any weak encoding into $\mathbb{E}$). Intuitively, the term $h_2(m).G$ in $H(m)$ plays the role of a one-time pad, to ensure that $H(m)$ can behave as a random oracle even though $f(h_1(m))$ does not reach all points in $\mathbb{E}$. Note that we 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. Our main result in this paper is that $H(m)$ is indifferentiable from a random oracle when h_1 and h_2 are seen as random oracles. Therefore $H(m)$ can be used in any cryptosystem provably secure with random oracle into elliptic curves, and the cryptosystem remains secure in the random oracle model for h_1 and h_2.

1.1 Related Work

An elliptic curve over a field $\mathbb{F}_{p^n}$ where $p > 3$ is defined by a Weierstrass equation:

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

where a and b are elements of $\mathbb{F}_{p^n}$. Throughout this paper, we note $E_{a,b}$ the curve associated to these parameters. It is well known that the set of points forms a group; we denote by $E_{a,b}(\mathbb{F}_{p^n})$ this group and by N its order. We denote $q = p^n$.

Super-singular Curves. A curve $E_{a,b}$ is called super-singular when $N = q + 1$. When $q \neq 1 \mod 3$, the map $x \mapsto x^3$ is a bijection, therefore the curves

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

are super-singular. One can then define the encoding

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

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

In the Boneh-Franklin scheme [2], one actually works in a subgroup $\mathbb{G}$ of prime order r of $E_{a,b}(\mathbb{F}_{p^n})$; we let ℓ such that $q+1 = \ell \cdot r$. 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 [2], Boneh and Franklin introduce the following notion of admissible encoding:

Definition 1 (Boneh-Franklin admissible encoding). *A function $f : A \to B$ 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 $b \in B$, $|f^{-1}(b)| = \ell$;*
3. *Samplable: there exists a probabilistic polynomial time algorithm that for any $b \in B$ returns a random element in $f^{-1}(b)$.*

The authors of [2] show that if $f : A \to \mathbb{G}$ is an admissible encoding, then the Boneh-Franklin scheme is secure with $H(m) = f(h(m))$, in the random oracle model for $h : \{0,1\}^* \mapsto A$. Since the function $f_{\mathbb{G}}$ is easily seen to be an admissible encoding, this shows that Boneh-Franklin is provably secure in the random oracle model with hash function $H_{\mathbb{G}}$ as defined in (1).

In this paper, we introduce a new notion of admissible encoding that is more general than the notion in [2]. This enables to use Icart's function that can work for any elliptic curve, instead of only super-singular ones. Moreover, the resulting hash function is indifferentiable from a random oracle; therefore, it can be used in any cryptosystem, not only in Boneh-Franklin.

1.2 Icart's Function

We consider the field $\mathbb{F}_{p^n}$ where $p > 3$ and $p^n = 2 \mod 3$. Let E be an elliptic curve over $\mathbb{F}_{p^n}$ with equation:

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

where $a, b \in \mathbb{F}_{p^n}$. In [4], Icart defines the function $f_{a,b} : \mathbb{F}_{p^n} \mapsto E$, with $f_{a,b}(u) = (x, y)$ where:

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

for $u \neq 0$, and $f_{a,b}(0) = \mathcal{O}$, the neutral element of the elliptic curve. It is easy to check that $f_{a,b}(u)$ is indeed a point of E for any $u \in \mathbb{F}_{p^n}$. We recall the following properties for $f_{a,b}$:

Lemma 1 (Icart). *The function $f_{a,b}$ is computable in deterministic polynomial time. For any point $P \in \mathsf{Im}(f_{a,b})$, we have that $f_{a,b}^{-1}(P)$ is computable in polynomial time and $|f_{a,b}^{-1}(P)| \leq 4$. We have $p^n/4 < |\mathsf{Im}(f_{a,b})| < p^n$.*

We note that Icart's function can also be defined in a field of characteristic 2 (see [4]).

2 Definitions

We recall the notion of indifferentiability introduced by Maurer *et al.* in [7]. 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* [1] 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.

The notion of indifferentiability [7] enables to show that an ideal primitive $\mathcal{H}_E$ (for example, a random oracle into an elliptic-curve E) can be replaced by a construction C that is based on some other ideal primitive $\mathcal{H}$ (for example, a random oracle into $\mathbb{F}_p$), and any cryptosystem secure with $\mathcal{H}_E$ remains secure with C and $\mathcal{H}$.

Definition 2 ([7]). *A Turing machine C with oracle access to an ideal primitive $\mathcal{H}$ is said to be $(t_D, t_S, q, \varepsilon)$-indifferentiable from an ideal primitive $\mathcal{H}_E$ if there exists a simulator S with oracle access to $\mathcal{H}_E$ and running in time at most t_S, such that for any distinguisher D running in time at most t_D and making at most q queries, it holds that:*

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

$C^{\mathcal{H}}$ is simply said to be indifferentiable from $\mathcal{H}_E$ if ε is a negligible function of the security parameter n, for polynomially bounded q, t_D and t_S.

It is shown in [7] that the indifferentiability notion is the "right" notion for substituting one ideal primitive with a construction based on another ideal primitive. That is, if $C^{\mathcal{H}}$ is indifferentiable from an ideal primitive $\mathcal{H}_E$, then $C^{\mathcal{H}}$ can replace $\mathcal{H}_E$ in any cryptosystem, and the resulting cryptosystem is at least as secure in the $\mathcal{H}$ model as in the $\mathcal{H}_E$ model; see [7] or [6] for a proof.

We also recall the definition of statistically indistinguishable distributions.

Definition 3. *Given two random variables X and Y over a set S, we say that the distribution of X and Y are ε-statistically indistinguishable if:*

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

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

3 A Random Oracle into Elliptic Curves

3.1 Previous Construction

Given an elliptic curve $E : y^2 = x^3 + ax + b$ defined over $\mathbb{F}_{p^n}$, let $f_{a,b}$ be Icart's function recalled in Section 1.2. Given a hash function $h : \{0,1\}^* \mapsto \mathbb{F}_{p^n}$, the following hash function $H : \{0,1\}^* \mapsto E$ is defined in [4]:

$$H(m) = f_{a,b}(h(m))$$

It is shown in [4] that H is one-way if h is one-way. However, it is easy to see that $H(m)$ does *not* behave like a random oracle when the underlying function h is seen as a random oracle; this is because f_{ab} does not reach all points of E.[1]

[1] moreover one can see that $f_{ab}(u)$ is not uniformly distributed in $\mathsf{Im} f_{a,b}$ when u is uniformly distributed in $\mathbb{F}_{p^n}$.

3.2 Admissible Encoding

Our goal in this paper is to construct a hash function into an elliptic-curve, that behaves as a random oracle when the underlying hash function is seen as a random oracle. First, we introduce our new notion of *admissible encoding*.

Definition 4 (Admissible Encoding). *A function $F : S \mapsto R$ is said to be a ε-admissible encoding if:*

1. *F is computable in deterministic polynomial time;*
2. *there exists a probabilistic polynomial time algorithm $\mathcal{I}_F$ such that given $r \in R$ as input, $\mathcal{I}_F$ outputs s such that either $F(s) = r$ or $s = \perp$, and the distribution of s is ε-statistically indistinguishable from the uniform distribution in S when r is uniformly distributed in R.*

Note that an admissible encoding F must be "almost surjective"; namely since by definition the distribution of $\mathcal{I}_F(r)$ is statistically close to uniform in S for uniformly distributed $r \in R$, we can have $\mathcal{I}_F(r) = \perp$ only with negligible probability. Note also that the distribution of $F(s)$ must be statistically close to uniform in R when s is uniformly distributed in S. Finally we note that our definition of admissible encoding is more general than the definition in [2] recalled in Section 1.1.

3.3 Indifferentiability

The following theorem shows that if $F : S \mapsto R$ is an admissible encoding, then:

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

is indifferentiable from a random oracle into R when $h : \{0,1\}^* \to S$ is seen as a random oracle; see Section 4 for the proof.

Theorem 1. *Let $F : S \mapsto R$ be a ε-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 $h : \{0,1\}^* \mapsto S$, with $\varepsilon' = 2q\varepsilon$.*

3.4 Weak Encoding

One can easily see however that Icart's function f is *not* an admissible encoding into the elliptic-curve E, since $\mathsf{Im} f$ covers only a fraction of the elliptic-curve points. Therefore, we introduce a weaker notion which we call a *weak encoding*.

Definition 5 (Weak Encoding). *A function $f : S \mapsto R$ is said to be a (α, ε)-weak encoding if:*

1. *f is computable in deterministic polynomial time.*
2. *there exists a probabilistic polynomial time algorithm $\mathcal{I}_f$, which given as input r uniformly distributed in R, outputs $s \in S \cup \perp$ such that $f(s) = r$ or $s = \perp$, and:*
   (a) $\Pr[s \neq \perp] \geq \alpha$
   (b) *the distribution of s conditioned on $s \neq \perp$ is ε-statistically indistinguishable from the uniform distribution in S.*

Probabilities are taken over $r \in R$ and the random coins of $\mathcal{I}_f$. If $\alpha(k) > 1/p(k)$ for some polynomial $p(k)$ and large enough k, and $\varepsilon(k) < 1/p'(k)$ for any polynomial $p'(k)$ and large enough k, we say that f is a weak encoding.

The difference with an admissible encoding is that for a weak encoding, algorithm $\mathcal{I}_f$ is only required to invert r for at least a polynomial fraction of the inputs (with still a statistically close to uniform distribution of outputs). Therefore the function $f : S \mapsto R$ need not be almost surjective, nor is it required that $f(u)$ is statistically close to uniform in R when u is uniform in S.

The following lemma shows that Icart's function is a weak encoding (see Section 5 for the proof).

Lemma 2 (Icart's Encoding). *Icart's function f_{ab} is an (α, ε)-weak encoding from $\mathbb{F}_{p^n}$ to $E_{a,b}$, where $\alpha = p^n/(4N)$ and $\varepsilon = 0$, where N is the order of $E_{a,b}$.*

3.5 From Weak Encoding to Admissible Encoding

Finally, we show how to turn a weak encoding into an admissible encoding when the output set is a group (see Section 6 for the proof).

Lemma 3 (Weak → Admissible Encoding). *Let $\mathbb{G}$ be a cyclic group of order N and let G be a generator of $\mathbb{G}$. Let $f : A \to \mathbb{G}$ be an (α, ε)-weak encoding. Then the function $F : A \times \mathbb{Z}_N \to \mathbb{G}$ with:*

$$F(a, x) := f(a) + x.G$$

is a ε'-admissible encoding into $\mathbb{G}$, where $\epsilon' = (1 - \alpha)^T + \varepsilon$ for any T, polynomial in k. For $T = -k/\log_2(1 - \alpha)$, one can take $\varepsilon' = 2^{-k} + \varepsilon$. Then if f is a weak encoding, F is an admissible encoding.

We note that it is easy to generalize the construction to a group with a finite set of generators.

3.6 Our Construction

To summarize, given an elliptic-curve defined over $\mathbb{F}_p$ with N points and a generator G, our 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 any weak encoding into $\mathbb{E}$, such as Icart's function.

Theorem 2. *Let $E : y^2 = x^3 + ax + b$ be an elliptic curve over $\mathbb{F}_{p^n}$ and let $f_{a,b} : \mathbb{F}_{p^n} \mapsto E$ be Icart's function. Let G be a generator of E of order N. The construction*

$$H(m) = f_{a,b}(h_1(m)) + h_2(m).G$$

is $2 \cdot q_D \cdot (1 - \alpha)^T$-indifferentiable from a random oracle, when hash functions $h_1 : \{0, 1\}^* \to \mathbb{F}_p$ and $h_2 : \{0, 1\}^* \to \mathbb{Z}_N$ are seen as random oracles. Letting $T = -k/\log_2(1 - \alpha)$, we have that the construction is $2 \cdot q_D \cdot 2^{-k}$-indifferentiable from a random oracle, where q_D is the number of distinguisher's queries.

4 Proof of Theorem 1

We must show that given a function $F : S \mapsto R$ that is a ε-admissible encoding, the construction $H(m) = F(h(m))$ is indifferentiable from a random oracle, in the random oracle model for $h : \{0,1\}^* \mapsto S$. We first describe our simulator.

4.1 Our Simulator

The simulator must simulate random oracle h to the distinguisher $\mathcal{D}$. The simulator has access to random oracle H. Our simulator maintains a list L of previously answered queries. Our simulator is based on algorithm $\mathcal{I}_F$ from admissible encoding F; formally:

Simulator $\mathcal{S}$:
Input: $m \in \{0,1\}^*$
Output: $s \in S$

1. If $(m, s) \in L$, then return s
2. Query $H(m) = r$
3. Let $s \leftarrow \mathcal{I}_F(r)$
4. Append (m, s) to L.
5. Return s

4.2 Indifferentiability

We show that the systems (C^h, h) and $(H, \mathcal{S}^H)$ are indistinguishable. We consider a distinguisher making at most q 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 $2q$ queries. We can then describe the full interaction between the distinguisher and the system as a sequence of triples:

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

In system (C^h, h), we have that the h_i's are uniformly and independently distributed in S, and $H_i = F(h_i)$ for all i. In system $(H, \mathcal{S}^H)$, we have that $H_i = F(h_i)$ except if $h_i = \perp$, by definition of algorithm $\mathcal{I}_F$ from admissible encoding F. Moreover, the definition of admissible encoding F implies that the distribution of h_i is ε-indistinguishable from the uniform distribution in S. Therefore, we obtain that the statistical distance between View in system (C^h, h) and View in system $(H, \mathcal{S}^H)$ is at most $2q\varepsilon$. This terminates the proof of Theorem 1.

5 Proof of Lemma 2

We actually prove a more general result than Lemma 2.

Lemma 4. *Let $f : S \to R$ be a polynomially computable function such that Im_f is at least a polynomial fraction of R. If there exists a polynomial-time algorithm Inv that for any r outputs $f^{-1}(r)$ in polynomial-time, then f is a weak encoding.*

Note that under the hypothesis of Lemma 4 the size of $f^{-1}(r)$ must be polynomially bounded for all r. From Lemma 1 we have that the hypotheses of Lemma 4 are satisfied for Icart's encoding function $f_{a,b}$; this proves Lemma 2.

5.1 Proof of Lemma 4

We must describe a polynomial-time algorithm $\mathcal{I}_F$ that given $r \in R$ outputs s such that $f(s) = r$ or $s = \perp$. We let B be an upper-bound on the size of $f^{-1}(r)$ for all r; from the hypotheses we can take B polynomial in the security parameter. Moreover we let $\beta = |\mathsf{Im} f|/|R|$; we have $\beta(k) > 1/\mathsf{poly}(k)$ for some $\mathsf{poly}(k)$.

Algorithm $\mathcal{I}_F$:
Input: $r \in R$
Outputs $s \in S$ such that $f(s) = r$ or $s = \perp$

1. Compute the set $X = f^{-1}(r)$ using algorithm Inv
2. Let $\delta_r = |X|/B$
3. With probability $1 - \delta_r$ return $\perp$
4. Return a random element s in X.

First, we compute the probability that algorithm $\mathcal{I}_F$ returns $s \neq \perp$ when input r is uniformly distributed in r:

$$\Pr[s \neq \perp] = \sum_{r \in R} \frac{1}{|R|} \cdot \delta_r = \sum_{r \in R} \frac{1}{|R|} \cdot \frac{|f^{-1}(r)|}{B} = \frac{|S|}{|R| \cdot B}$$

Since we have:

$$\beta = \frac{|\mathsf{Im} f|}{|R|} \leq \frac{|S|}{|R|}$$

we obtain:

$$\Pr[s \neq \perp] \geq \frac{\beta}{B} > \frac{1}{\mathsf{poly}'(k)}$$

Now we consider the distribution of s conditioned on $s \neq \perp$, for uniformly distributed $r \in R$. We consider a given $u \in S$; if $s = u$, then we must have $s \neq \perp$ and $r = f(u)$; therefore:

$$\Pr[s = u] = \Pr[s = u \wedge s \neq \perp \wedge r = f(u)]$$

which gives:

$$\Pr[s = u] = \Pr[s = u | s \neq \perp \wedge r = f(u)] \cdot \Pr[s \neq \perp | r = f(u)] \cdot \Pr[r = f(u)]$$

From the definition of algorithm $\mathcal{I}_F$, we have:

$$\Pr[s = u | s \neq \perp \wedge r = f(u)] = \frac{1}{|X_u|}$$

where $X_u = f^{-1}(f(u))$, and:

$$\Pr[s \neq \perp | r = f(u)] = \delta_{f(u)} = \frac{|X_u|}{B}$$

This gives:

$$\Pr[s = u] = \frac{1}{|X_u|} \cdot \frac{|X_u|}{B} \cdot \frac{1}{|R|} = \frac{1}{B \cdot |R|}$$

and eventually:

$$\Pr[s = u | s \neq \perp] = \frac{\Pr[s = u]}{\Pr[s \neq \perp]} = \frac{1}{B \cdot |R|} \cdot \frac{|R| \cdot B}{|S|} = \frac{1}{|S|}$$

which shows that the distribution of s conditioned on $s \neq \perp$ is uniform in S; this terminates the proof of Lemma 4.

6 Proof of Lemma 3

We consider the following inverting algorithm $\mathcal{I}_F$:

Algorithm $\mathcal{I}_F$:
Input: $P \in \mathbb{G}$
Output: $(a, z) \in A \times \mathbb{Z}_N$ such that $P = F(a, z) = f(a) + z.G$, or $\perp$

1. For $i = 1$ to T:
   (a) Randomly chooses $z \in \mathbb{Z}_N$ and computes $Z = z.G$
   (b) Let $X = P - Z \in \mathbb{G}$
   (c) Compute $a = \mathcal{I}_f(X)$
   (d) If $a \neq \perp$, return (a, z)
2. Return $\perp$.

It is easy to see that for $(a, z) \neq \perp$, we have $P = F(a, z) = f(a) + z.G$ as required. We must show that for a uniformly distributed input P, the distribution of (a, z) is statistically close to uniform in $A \times \mathbb{Z}_N$.

We first consider the distribution of (a, z) for a fixed input P. Since f is a (α, ε)-weak encoding and for every i the group element $X = P - z.G$ is uniformly and independently distributed in $\mathbb{G}$, at step i we have $a = \perp$ with probability at most $1 - \alpha$, and eventually algorithm $\mathcal{I}_F$ outputs $a = \perp$ with probability at most $(1 - \alpha)^T$. Moreover, conditioned on $a \neq \perp$, the distribution of a in (a, z) is ε-statistically indistinguishable from the uniform distribution in A.

Let (a_P, z_P) be the random variable obtained for a fixed P, conditioned on $(a_P, z_P) \neq \perp$. We have that the distribution corresponding to $P' = P + v.G$ for any $v \in \mathbb{Z}_N$ is given by $(a_P, z_P + v)$. Therefore, for input P' uniformly distributed in $\mathbb{G}$, the value of z in $(a, z) = (a_P, z_P + v)$ is uniformly distributed in $\mathbb{Z}_N$ and independently from a. Then for uniformly distributed P' and conditioned on $(a, z) \neq \perp$, the distribution of (a, z) is ε-statistically indistinguishable from the uniform distribution in $A \times \mathbb{Z}_N$. Finally, since $(a, z) = \perp$ with probability at most $(1 - \alpha)^T$, the distribution of (a, z) is ε'-statistically indistinguishable from the uniform distribution, with:

$$\varepsilon' = \varepsilon + (1 - \alpha)^T$$

which terminates the proof of Lemma 3.

7 Extension to Prime Order Subgroup

We have seen in Section 3 how to construct a hash function $H(m)$ into an elliptic curve E that is indifferentiable from a random oracle into E. However, in many applications only a prime order subgroup of E is used. Therefore, we show how to construct a random oracle into a subgroup.

We start by showing that the composition of two admissible encodings remains an admissible encoding.

Lemma 5. *Let $F : R \mapsto S$ be a ε_1-admissible encoding and $G : S \mapsto T$ be a ε_2-admissible encoding. Then $G \circ F$ is a $(\varepsilon_1 + \varepsilon_2)$-admissible encoding from R to T.*

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

Now we show that multiplication by a cofactor is an admissible encoding. More precisely, let E be an Abelian group of order N, and let $\mathbb{G}$ be a prime-order subgroup of order q with $N = r \cdot q$, where r is called the co-factor. Let $\mathbb{G}_r$ be the subgroup of order r.

Lemma 6. *Assume that there exists a randomized polynomial time algorithm* $\mathsf{Gen}(\mathbb{G}_r)$ *that generates uniformly distributed elements in* $\mathbb{G}_r$*. Then the map* $M_r : E \mapsto \mathbb{G}$ *with* $M_r(G) = r.G$ *is a* ε*-admissible encoding, with* $\varepsilon = 0$*.*

Proof. Firstly, M_r is a deterministic map computable in polynomial time. Secondly, we describe an algorithm $\mathcal{I}_M$ that computes a random preimage of $P \in \mathbb{G}$ under M_r. Algorithm $\mathcal{I}_M$ first computes a random element $G_r \in \mathbb{G}_r$ thanks to $\mathsf{Gen}(\mathbb{G}_r)$. Then it computes $P' = (1/r) \cdot P + G_r$. Clearly, we have $r \cdot P' = P$. Moreover, P' has the uniform distribution in E when P is uniformly distributed in $\mathbb{G}$. □

We note that when cofactor r is small, or when a base of generators of $\mathbb{G}_r$ is known, we can easily construct such algorithm $\mathsf{Gen}(\mathbb{G}_r)$; however, when the factorization of r is unknown, it is unclear how to find such algorithm.

Let E be an elliptic-curve with N points and cyclic generator G_E, and with a prime order subgroup $\mathbb{G}$ of order q and with $G = r.G_E$ as a generator. Combining Lemma 5 and Lemma 6 we have that:

$$F'(u, x) = M_r\left(f(u) + x.G_E\right) = r.f(u) + (r \cdot x).G_E$$

is an admissible encoding from $\mathbb{F}_p \times \mathbb{Z}_N$ to $\mathbb{G}$. However we see that $F'(u, x)$ only depends on $x \mod q$ (instead of $x \mod N$). Therefore our final construction is $F : \mathbb{F}_p \times \mathbb{Z}_q \rightarrow \mathbb{G}$ with:

$$F(u, y) = r.f(u) + y.G$$

where G is a generator of subgroup $\mathbb{G}$; it is easy to see that this map is also an admissible encoding. The corresponding hash function $H : \{0,1\}^* \mapsto \mathbb{G}$ is then:

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

where $h_1 : \{0,1\}^* \mapsto \mathbb{F}_p$ and $h_2 : \{0,1\}^* \mapsto \mathbb{Z}_q$ are two hash functions, and H is indifferentiable from a random oracle into $\mathbb{G}$, in the random oracle model for h_1 and h_2.

8 Extension to Random Oracles into Strings

The constructions in the previous sections were based on hash functions into $\mathbb{F}_{p^n}$ or $\mathbb{Z}_N$ that were seen as random oracles. However in practice a hash function outputs a fixed length string, not an element of $\mathbb{F}_{p^n}$ or $\mathbb{Z}_N$. Therefore in this section show how to construct a hash function into $\mathbb{F}_{p^n}$ or $\mathbb{Z}_N$ that is indifferentiable from a random oracle into $\mathbb{F}_{p^n}$ or $\mathbb{Z}_N$, given a hash function seen as a random oracle into $\{0,1\}^\ell$. Actually it suffices to construct an admissible encoding from $\{0,1\}^\ell$ to $\mathbb{Z}_N$ for any N; namely for $\mathbb{F}_{p^n}$ there is a simple bijection with $\mathbb{Z}_{p^n}$.

Lemma 7 (From $\{0,1\}^\ell$ to $\mathbb{Z}_N$). *Let $\mathbb{Z}_N$ be an integer modular ring and let k be a security parameter. Let $\ell = k + \lceil \log_2 N \rceil + 1$. The function* $\text{Mod}_N : [0, 2^\ell - 1] \mapsto \mathbb{Z}_N$ *with:*

$$\text{Mod}_N(b) = b \mod N$$

is a 2^{-k}-admissible encoding.

Proof. See Appendix A.

Our construction is then modified as follows. We consider an elliptic curve $E_{a,b}(\mathbb{F}_p)$ of prime order N and generator G, with p a $2k$-bit prime. We define the hash function $H : \{0,1\}^* \mapsto E_{a,b}(\mathbb{F}_p)$ with:

$$H(m) := f_{a,b}\big(h_1(m) \mod p\big) + \big(h_2(m) \mod N\big).G$$

where h_1 and h_2 are two hash functions from $\{0,1\}^*$ to $\{0,1\}^{3k}$. From Lemma 5 and 7 we obtain the following result.

Lemma 8. *The previous hash function H is $2 \cdot q_D \cdot 2^{-k}$-indifferentiable from a random oracle, in the random oracle model for h_1 and h_2.*

Remark 1. We only need a single hash function $h : \{0,1\}^* \to \{0,1\}^{3k}$ instead of h_1 and h_2 since we can obtain h_1 and h_2 by prepending a bit as input of h.

Remark 2. Instead of using two strings of $3k$-bit each, we can use a single string of $5k$-bit only. Namely one can show that the construction:

$$H'(m) := f_{a,b}\big(h(m) \mod p\big) + \big(h(m) \mod N\big).G$$

is $2 \cdot q_D \cdot 2^{-k}$-indifferentiable from a random oracle, in the random oracle model for $h : \{0,1\}^* \mapsto \{0,1\}^{5k}$.

9 Conclusion

We have described the first construction of a hash function into elliptic curves that is indifferentiable from a random oracle, based on Icart's function. Our construction is efficient and can be used in password-based authentication protocols over elliptic curves.

References

1. 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.
2. 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.
3. Victor Boyko, Philip D. MacKenzie, and Sarvar Patel. Provably secure password-authenticated key exchange using diffie-hellman. In *EUROCRYPT*, pages 156–171, 2000.
4. Thomas Icart. How to hash into an elliptic-curve. In *CRYPTO 2009 (to appear)*. Publicly available on `http://eprint.iacr.org/`.
5. David P. Jablon. Strong password-only authenticated key exchange. *SIGCOMM Comput. Commun. Rev.*, 26(5):5–26, 1996.

6. C. Malinaud J.S. Coron, Y. Dodis and P. Puniya. Merkle-damgård revisited: How to construct a hash function. In *CRYPTO*, 2005.
7. 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.
8. 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.

A Proof of Lemma 7

Let $\mu = \left\lfloor \frac{2^\ell}{N} \right\rfloor$, which gives:

$$2^\ell - N < \mu N \leq 2^\ell.$$

The algorithm $\mathcal{I}_{\text{MOD}}$ is as follows. Given as input $n \in \mathbb{Z}_N$, it randomly selects an integer r in $[0, \mu - 1]$ and returns $b = n + rN$.

Clearly, the element b satisfies $b \mod N = n$. Moreover when n is uniformly distributed in $\mathbb{Z}_N$, then b is uniformly distributed in $[0, \mu N - 1]$. We must show that the distribution of b is statistically indistinguishable from the uniform distribution in $\left[0, 2^\ell - 1\right]$. We have:

$$\begin{aligned}
\sum_{i=0}^{2^\ell - 1} \left| \Pr[b = i] - \frac{1}{2^\ell} \right| &= \sum_{i=0}^{\mu N - 1} \left| \frac{1}{\mu N} - \frac{1}{2^\ell} \right| + \sum_{i=\mu N}^{2^\ell - 1} \left| 0 - \frac{1}{2^\ell} \right| \\
&= \frac{\mu N (2^\ell - \mu N)}{\mu N 2^\ell} + \frac{2^\ell - \mu N}{2^\ell} \\
&= 2 \cdot (1 - \frac{\mu N}{2^\ell}) < 2 \cdot (1 - \frac{2^\ell - N}{2^\ell}) \\
&< \frac{N}{2^{\ell - 1}} < \frac{1}{2^k}
\end{aligned}$$

which shows that the distribution of b is 2^{-k}-indistinguishable from the uniform distribution in $[0, 2^\ell - 1]$. This terminates the proof of Lemma 7.