Mapping an Arbitrary Message to an Elliptic Curve when Defined over $GF(2^n)$

Brian King
Indiana University - Purdue University Indianapolis
723 W Michigan, SL 160 Indianapolis, IN 46202
(Email: briaking@gmail.com)
(Received Sept. 8, 2006; revised Nov. 8, 2006; and accepted Apr. 4, 2007)

Abstract

The use of elliptic curve cryptography (ECC) when used as a public-key cryptosystem for encryption is such that if one has a message to encrypt, then they attempt to map it to some point in the prime subgroup of the elliptic curve by systematically modifying the message in a deterministic manner. The applications typically used for ECC are the key-exchange, digital signature or a hybrid encryption systems (ECIES) all of which avoid this problem. In this paper we provide a deterministic method that guarantees that the map of a message to an elliptic curve point can be made without any modification. This paper provides a solution to the open problem posed in [7] concerning the creation of a deterministic method to map arbitrary message to an elliptic curve.

Keywords: Elliptic curve cryptography, public-key encryption, message mapping

1 Introduction

The use of elliptic curves for cryptography was first suggested independently by Koblitz and Miller [6, 9]. It is well recognized that ECC (Elliptic Curve Cryptography) is an excellent public-key cryptosystem for low-complexity devices and networks. The reason is that it requires less bandwidth, as well as less computational complexity when performing key exchange and/or constructing a digital signature.

An important aspect of ECC is that most discrete-log (DL) primitives possess an EC (Elliptic Curve) variant. For example, for the Diffie-Hellman key exchange (DH), the ElGamal public-key encryption, and Digital Signature Algorithm (DSA) there exists ECDH, EC ElGamal and ECDSA. In DL primitives, messages belong either to $\mathbb{Z}_{p-1}$ or $\mathbb{Z}_q$, which is a prime subgroup of $\mathbb{Z}_{p-1}$.

To apply ECC as a public-key cryptographic primitive one would require the message M to belong to the elliptic curve (in fact M would have to belong to a prime subgroup of the elliptic curve). A suitable modification would have the message M be the x or y coordinate of an elliptic curve point. However it is very unlikely that a message would be the x or y coordinate of some EC point.

This problem was easily overcome by the way the protocols ECDH and ECDSA were developed. In ECDH, the secret key is selected to be the x-coordinate of the computed EC point and in ECDSA the message that is to be signed is used to construct a scalar. If one would like to apply ECC to provide public-key encryption to encrypt some message, such as EC ElGamal cryptosystem, one would have to map the message to an EC point. The only know algorithm to achieve this is probabilistic.

This is outlined in Section 3. An alternative is to apply ECIES (Elliptic Curve Integrated Encryption System) [3]. This is a hybrid cryptosystem which applies both symmetric and asymmetric encryption (see Section 2.4). In this case there is no concern whether the message M can be mapped to an EC point, since the message is encrypted with a symmetric-key cryptosystem.

However there are cases when it is preferred to use a traditional public-key encryption system rather than a hybrid encryption system, since those systems that support a hybrid encryption system will have to implement two encryptions, one a public-key cryptosystem and the other a symmetric-key cryptosystems. For examples see Section 2.4 and Section 7.

In this work we describe a deterministic method which allows us to map any non-trivial (non-zero) message M (interpreted as a member of the field $GF(2^n)$) to a valid EC point (a point in the prime subgroup of a secure elliptic curve). This solves an open problem concerning the construction of a deterministic method to map messages to an elliptic curve that was posed in [7].

Table 1: Computing trace of a field element for all fields used in the construction of NIST curves defined over $GF(2^n)$ as described in [5]

NIST Curve types	Generating polynomial	$Tr(\mu)$ for $\mu = \mu_{n-1}u^{n-1} + \cdots + \mu_1 u + \mu_0$
K-163, B-163	$p(t) = t^{163} + t^7 + t^6 + t^3 + 1$	$Tr(\mu) = \mu_0 + \mu_{157}$
K-233, B-233	$p(t) = t^{233} + t^{74} + 1$	$Tr(\mu) = \mu_0 + \mu_{159}$
K-283, B-283	$p(t) = t^{283} + t^{12} + t^7 + t^5 + 1$	$Tr(\mu) = \mu_0 + \mu_{277}$
K-409, B-409	$p(t) = t^{409} + t^{87} + 1$	$Tr(\mu) = \mu_0$
K-571, B-571	$p(t) = t^{571} + t^{10} + t^5 + t^2 + 1$	$Tr(\mu) = \mu_0 + \mu_{561} + \mu_{569}$

2 Background

2.1 $GF(2^n)$ Tools

The trace function, denoted by Tr, is a linear mapping[1] of $GF(2^n)$ onto $\mathbb{Z}_2$. The trace of an element $\alpha \in GF(2^n)$, denoted by $Tr(\alpha)$ can be computed as $Tr(\alpha) = \sum_{i=0}^{n-1} \alpha^{2^i}$ (Chapter 2 of [8]). However, in most binary fields, the trace can be computed by examining "certain bits" of the field element (see Table 1). The trace function satisfies that $Tr(\alpha^2) = Tr(\alpha)$. Further, if n is odd, an assumption we will always make due to security[2] then $Tr(1) = 1$. Consequently for all $\alpha \in GF(2^n)$ with $Tr(\alpha + 1) = Tr(\alpha) + 1$.

For a given $b \in GF(2^n)$, the quadratic equation $\lambda^2 + \lambda = b$ in $GF(2^n)$ has a solution if and only if $Tr(b) = 0$ (Chapter 2 of [8]). Observe that if λ is a solution to the above quadratic equation, then $\lambda + 1$ is also a solution, and $Tr(\lambda + 1) = Tr(\lambda) + 1$. Hence, whenever n is odd (which we always will assume), for each solvable quadratic equation there is a solution with trace 1 and a solution with trace 0. One can compute the solution to $\lambda^2 + \lambda = b$ by computing $\lambda = \sum_{j=0}^{(n-1)/2} b^{2^{2j}}$ (Chapter 2 of [1]). Integral to our work will be the algorithm: **Solve(s)** which returns No solution whenever $Tr(s) \neq 0$, otherwise it returns an arbitrary solution to the equation $z^2 + z = s$.

2.2 ECC over $GF(2^n)$

For the finite field $GF(2^n)$, the standard equation for a non supersingular elliptic curve is:

$$y^2 + xy = x^3 + a_2x^2 + a_6, \quad (1)$$

where $a_2, a_6 \in GF(2^n)$, $a_6 \neq 0$. The points $P = (x, y)$, where $x, y \in GF(2^n)$, that satisfy the equation, together with the point $\mathcal{O}$, called the point of infinity, form an additive abelian group E_{a_2,a_6}. Here addition in E_{a_2,a_6} is defined by: for all $P \in E_{a_2,a_6}$

- $P + \mathcal{O} = P$,
- for $P = (x, y) \neq \mathcal{O}$, the point $-P$ is computed as $-P = (x, x + y)$
- and for all $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$, where both $P_1 \neq \mathcal{O}$, $P_2 \neq \mathcal{O}$ and $P_1 \neq -P_2$, the point $P_1 + P_2$ is computed as $P_1 + P_2 = P_3 = (x_3, y_3)$ where $x_3, y_3 \in GF(2^n)$ and satisfy:

$$\begin{aligned} x_3 &= \lambda^2 + \lambda + x_1 + x_2 + a_2, \quad \text{and} \\ y_3 &= \lambda(x_1 + x_3) + x_3 + y_1, \end{aligned}$$

where $\lambda = \frac{y_1+y_2}{x_1+x_2}$ if $P_1 \neq P_2$ and $\lambda = x_1 + \frac{y_1}{x_1}$ for $P_1 = P_2$.

In a cryptographic application, the elliptic curve will be selected so that E_{a_2,a_6} will contain a large subgroup of prime order. The cryptographically relevant points will be non-trivial elements of this subgroup of large prime order.

If $(x, y) \in E_{a_2,a_6}$ and $x \neq 0$ then $\frac{y^2}{x^2} + \frac{y}{x} = x + a_2 + \frac{a_6}{x^2}$. By making the substitution $z = \frac{y}{x}$ we see that $z^2 + z = x + a_2 + \frac{a_6}{x^2}$. Since this quadratic equation is solvable, we see that $Tr(x + a_2 + \frac{a_6}{x^2}) = 0$. Further, a necessary condition for a point $P = (x_1, y_1) \in E_{a_2,a_6}$ to belong to the prime subgroup of E_{a_2,a_6} (when defined over $GF(2^n)$) is that $Tr(x_1) = Tr(a_2)$ [11].

2.3 EC ElGamal Public-key Encryption

A method that allows one to use an elliptic curve as public-key encryption system is EC ElGamal encryption.

Suppose there exists message $\mathcal{X}$, where $\mathcal{X} \in E_{a_2,a_6}$ such that $\mathcal{X}$ belongs to the prime subgroup. Now suppose Alice possesses $\mathcal{X}$ and would like to privately send it to Bob using EC ElGamal encryption. Alice gets Bob's public key Y (where $Y = kP$, k is Bob's private key and P is a generator of the prime subgroup). She then selects a random integer r in $\mathbb{Z}_q$, where q is the order of the prime subgroup, and compute rP and sets $C_1 = rP$. Alice then computes $C_2 = rY + \mathcal{X}$. Alice then sends (C_1, C_2) to Bob. Bob decrypts (C_1, C_2) by computing $kC_1 - C_2$ which is $\mathcal{X}$.

2.4 ECIES

ECIES [3], *Elliptic Curve Integrated Encryption System*, is a hybrid cryptosystem that uses both asymmetric-key

[1] $Tr(\alpha + \beta) = Tr(\alpha) + Tr(\beta)$.

[2] Due to an attack by [2], it is recommended that whenever one uses an elliptic curve E defined over $GF(2^n)$, that n should be prime.

and symmetric-key cryptosystem. It is based on DLIES (Discrete-Log Integrated Encryption System). Suppose E_{a_2,a_6} is a secure elliptic curve defined over $GF(2^n)$. Let P be a generator of the prime subgroup. Suppose Y is the public-key of Alice. Then $Y = kP$, where k is Alice's private-key. Optionally S_1, S_2 are shared information. Suppose Bob wishes to send Alice message M. Bob generates a random scalar w and computes $R = wP$ and derives a shared secret $S = wY$. Then by applying a key derivation function KDF, Bob computes the encryption key k_e and MAC key k_m where $k_e||k_m = KDF(S||S_1)$. Let $c = E_{k_e}(M)$ and $d = MAC_{k_m}(c||S_2)$. Bob sends $R||c||d$ to Alice.

Alice decrypts and verifies the integrity of the message in the following way. First she uses her private key k_{Alice}, and computes $k_{Alice}R$ which is S. Since she has both S_1 and S_2, she computes $k_e||k_m = KDF(S||S_1)$. She then computes $D_{k_e}(c)$. Lastly, she verifies the integrity of the transmission by computing $MAC(c||S_2)$ which is d.

ECIES provides an effective tool for public-key encryption, especially since it supports the encryption of large amounts of content. There are two drawbacks of ECIES. First it requires the use of two cryptosystems, no matter what the size of the content that needs to be encrypted. That is, there are cases for which "small messages" need to be encrypted, the use of ECIES would still require the implementation and use of two cryptosystems, whereas traditional public-key encryption systems like RSA and ElGamal can be used to encrypt small messages. Secondly, there are cases where anonymity is important. In situations where anonymity of private communication is necessary, one could use a MIX network [4], in order to achieve anonymity. We provide a brief summary. In such an application, all messages are encrypted and transmitted to a system of servers. Here a server re-encrypts and shuffles all the ciphertexts, then forwards them to the next server in the system. This server re-encrypts and shuffles ciphertexts forwarding them to the next server. This process continues until all servers have re-encrypted and shuffled ciphertexts. Threshold cryptography can be used to decrypt the messages. ECIES cannot be applied in any application that requires re-encryption, due to the use of the symmetric cryptosystem.

3 Prior Techniques to Mapping/Embed a Plaintext Message to an EC Point

3.1 A Mapping for Elliptic Curves Defined over Prime Fields

The only existing map of plaintext messages to an EC point is a probabilistic algorithm. We describe the map as outlined in Chapter 16 of [13], this algorithm is usually attributed to Koblitz. In [13], this method is described as a mapping of a plaintext message to an EC point where the elliptic curve is defined over a prime field $\mathbb{Z}_p$. We describe this technique, for elliptic curves defined over prime fields, and then describe how to modify it as a method for mapping plaintext messages to an EC point where the elliptic curve is defined over $GF(2^n)$.

The standard equation of an elliptic curve defined over a prime field $\mathbb{Z}_p$ is given by

$$y^2 = x^3 + ax + b \bmod p,$$

where $a, b \in \mathbb{Z}_p$ are the elliptic curve parameters. Suppose M is a plaintext message. Clearly no matter the message, M can be interpreted as a binary string. Thus M can be interpreted as a member of the field $\mathbb{Z}_p$. The fundamental property of this algorithm is that there is a probability of $1/2$ that $M^3 + aM + b$ has a square root mod p.

Let K be an integer and suppose that M satisfies $(M + 1)K < p$. Then for each j, $j = 0$ to $K - 1$, let $x_j = M \cdot K + j \mod p$. One computes $z_j = x_j^3 + ax_j + b$, and checks whether z_j has a square root, if so one stops. Otherwise one continues computing the next x_j. This process continues, only terminating when a square root is found, in which we have a successful mapping, or when one has exhausted all possibilities, in this case the algorithm was unsuccessful in mapping the plaintext message to an EC point. The probability that the mapping algorithm will not succeed in mapping M to an EC point is $\leq \frac{1}{2^K}$. The algorithm is formally presented in Algorithm 1.

Algorithm 1 Probabilistic mapping of plaintext message to a point on an elliptic curve defined over a prime field [13]

```
for j = 0 to K − 1 do
    let x_j = M · K + j  mod p
    if z_j = x_j^3 + a x_j + b has a square root mod p then
        break
    end if
end for
if j < K then
    compute y_j a square root of z_j  mod p
    map M to (x_j, y_j)
else
    output "unsuccessful in attempt to map M to an EC point"
end if
```

The sender encrypts the EC point (encoded message) sends it to the receiver. The receiver decrypts the ciphertext and computes the EC point (x, y). Then the message M can be decoded by the receiver by computing $\left\lfloor \frac{x}{K} \right\rfloor$.

3.2 A Mapping for Elliptic Curves Defined over Binary Fields

The following mapping is outlined in [7].

To modify Algorithm 1, for elliptic curves defined over binary fields, we need to make a series of observations. Recall that half of the element in $GF(2^n)$ have a trace of 0. Secondly, recall that (x, y) belongs to the elliptic

curve, i.e. satisfies Equation (1), provided

$$\frac{y^2}{x^2} + \frac{y}{x} = x + a_2 + \frac{a_6}{x^2}. \quad (2)$$

Further if (x, y) belongs to the prime subgroup of the elliptic curve then

$$Tr(x) = Tr(a_2). \quad (3)$$

Note that the necessary Equations (2) and (3) are sufficient conditions to establish that (x, y) belongs to the prime subgroup of E_{a_2,a_6} provided the curve has a cofactor of 2. Observe that for Equations (2) and (3) to hold, one must have $Tr(x) = Tr(a_2)$ and $Tr(\frac{a_6}{x^2}) = 0$. Recall that due to a security requirement n should be odd, thus $Tr(1) = 1$. In practice it is very easy to determine if an element in a binary field has trace 0. In [5], the relations for computing the trace of a field element for all fields used for the NIST binary curves [10] were provided. For example, given the binary field $GF(2^{233})$ with generating polynomial $t^{233} + t^{74} + 1$, for each field element $\mu = \mu_{232}u^{232} + \mu_{231}u^{231} + \ldots + \mu_1 u + \mu_0 \in GF(2^{233})$, the trace of μ is $Tr(\mu) = \mu_0 + \mu_{159}$ (where + is an XOR).

Since many of the secure elliptic curves have cofactor 2, and in order to provide a simple demonstration of how to modify Algorithm 1 for binary fields, we will assume that E_{a_2,a_6} has a cofactor 2.

Observe that there is a natural mapping between elements of $GF(2^n)$ and nonnegative integers $< 2^n$, such that each $\tau = \mu_{n-1}u^{n-1} + \mu_{n-2}u^{n-2} + \cdots + \mu_1 u^1 + \mu_0$ is associated with integer $t = \mu_{n-1}2^{n-1} + \mu_{n-2}2^{n-2} + \cdots + \mu_1 2^1 + \mu_0$. Consequently there is a natural ordering $<_*$ that exists on $GF(2^n)$ such that $\tau <_* \tau'$ if and only if $t < t'$. Suppose that M is a plaintext message. Trivially this message can be interpreted as a member of the binary field $GF(2^n)$ when one uses a polynomial representation. Suppose that there exists positive integer κ such that $M \cdot u^\kappa$, when interpreted as a polynomial in u, has degree less than n. Thus $M \cdot u^\kappa <_* u^n$. There are $r = 2^\kappa$ many elements τ of $GF(2^n)$, satisfying $\tau < u^\kappa$. We enumerate them as $\tau_0, \tau_1, \ldots, \tau_{2^\kappa - 1}$ where $\tau_i <_* \tau_j$ whenever $i < j$. For $k = 0$ to $2^\kappa - 1$, we compute $\alpha_j = M \cdot u^\kappa + \tau_j$, and compute $Tr(\alpha_j + a_2 + \frac{a_6}{\alpha_j^2})$. If $Tr(\alpha_j + a_2 + \frac{a_6}{\alpha_j^2}) = 0$, then α_j is an x-coordinate of some point on the elliptic curve E_{a_2,a_6}. If it is, then we check if $Tr(\alpha_j) = Tr(a_2)$. If this second condition is true then α_j is an x-coordinate of a point in the prime subgroup. If any of these two conditions fail, we compute the next α_j. This process continues until we have either found an x-coordinate of a point in the prime subgroup or we have exhausted all τ_j. If we find an α_j, which is an x-coordinate of a point in the prime subgroup, then we compute $\lambda =$**Solve**$(\alpha_j + a_2 + \frac{a_6}{\alpha_j^2})$, and we map the message M to $(\alpha_j, \alpha_j \cdot \lambda)$. If we exhaust all τ_j we output unsuccessful to map M to the elliptic curve. The complete algorithm is given in Algorithm 2.

Since half of the elements of $GF(2^n)$ have trace zero, the probability that $\alpha_j + a_2 + \frac{a_6}{\alpha_j^2}$ has trace zero is 1/2.

Algorithm 2 Probabilistic mapping of plaintext message to point on an elliptic curve defined over a binary field as outlined in [7]

```
r = 2^κ
j = 0
while j ≤ r do
   set α_j = M · u^κ + τ_j
   if Tr(α_j + a_2 + a_6/α_j^2) = 0 and Tr(α_j) = Tr(a_2) then
      break
   else
      j = j + 1
   end if
end while
if j ≤ r then
   Solve for λ where λ^2 + λ = α_j + a_2 + a_6/α_j^2
   output EC point (α_j, α_j · λ)
else
   output "unsuccessful in attempt to map M to an EC point"
end if
```

Since half of the points of the curve belong to the prime subgroup (recall the cofactor of the curve is 2), the probability that an α_j is an x-coordinate of an EC point that belongs to the prime subgroup is $1/2 \cdot 1/2 = 1/4$. Therefore, for the given κ, the probability that Algorithm 1 is unsuccessful in mapping M to the elliptic curve is $(\frac{1}{4})^{2^\kappa} = (\frac{1}{4})^r = \frac{1}{4^r}$ where $r = 2^\kappa$.

To recover the message M, the receiver decrypts the ciphertext and computes the point $X = (x, y)$. The receiver computes x/u^κ as a polynomial division and ignores the fractional part. The computation is equivalent to a right shift by κ. The result is M.

4 Isomorphisms of Binary Elliptic Curves

We now make a series of observations concerning the elliptic curve parameters that define the elliptic curve in Equation (1). The following result is provided in [1, 12].

Theorem 1. *[12] Let $\gamma \in GF(2^n)$ such that $Tr(\gamma) = 0$ then for all a, b we have $|E_{a+\gamma,b}| = |E_{a,b}|$.*

A consequence of this theorem is that if $E_{a,b}$ represents a cryptographically relevant elliptic curve defined over $GF(2^n)$, then there exists 2^{n-1} many cryptographically relevant curves defined over the same field that are "equivalent" (this follows from the fact that half of the field elements of $GF(2^n)$ have trace zero). In [12], it was shown that these curves are isomorphic to each other, and we now demonstrate that the isomorphism can easily be computed and applied.

Theorem 2. *Suppose $\gamma \in GF(2^n)$ such that $Tr(\gamma) = 0$. Then $E_{a+\gamma,b}$ and $E_{a,b}$ are isomorphic. Further, the map $f : E_{a+\gamma,b} \longrightarrow E_{a,b}$ where $f(\mathcal{O}) = \mathcal{O}$ and for all $(x, y) \in E_{a+\gamma,b}$ $f(x, y) = (x, y + x \cdot$ **Solve**$(\gamma))$ is an isomorphism.*

Proof. Let $(\tilde{x}, \tilde{y})$ denote points in $E_{a,b}$ and (x, y) denote points in $E_{a+\gamma,b}$. We will show that $E_{a+\gamma,b}$ and $E_{a,b}$ are

isomorphic, as well as that the map $f : E_{a+\gamma,b} \mapsto E_{a,b}$ is an isomorphism where for $P = (x, y) \in E_{a+\gamma,b} \setminus \{\mathcal{O}\}$ $f(P) = f((x, y) = (x, y + x \cdot \mathbf{Solve}(\gamma)) = (\tilde{x}, \tilde{y}) \in E_{a,b}$. The equation for $E_{a,b}$ is given by Equation (1), and the equation for $E_{a+\gamma,b}$ is given by

$$y^2 + xy = x^3 + (a + \gamma)x^2 + b.$$

We will show

1) $f(-P) = -f(P)$.

2) $f(2P) = 2f(P)$.

3) For $P \neq \pm Q$, we have $f(P + Q) = f(P) + f(Q)$.

Let the mapping f satisfy that $f(\mathcal{O}) = \mathcal{O}$ and for $P = (x, y) \in E_{a+\gamma,b} \setminus \{\mathcal{O}\}$ $f(P) = f((x, y) = (x, y+x\cdot\mathbf{Solve}(\gamma)) = (\tilde{x}, \tilde{y}) \in E_{a,b}$. Then trivially we see f is both one-to-one and onto. For brevity let S denote $\mathbf{SOLVE}(\gamma)$, thus $S^2 + S = \gamma$.

To establish Item 1). Let $P \in E_{a+\gamma,b}$, assume $P = (x, y)$, then $f(P) = (x, y + x \cdot S)$. Thus $-f(P) = (x, y + x \cdot S + x)$. Now consider $f(-P)$ observe that $f(-P) = f(-(x, y)) = f(x, y + x) = (x, y+x+x\cdot S) = -f(P)$. Consequently $f(-P) = -f(P)$.

To establish Item 2). Let $P \in E_{a+\gamma,b}$, Again assume $P = (x, y)$. Then $2P = (x_3, y_3)$ where $x_3 = \lambda^2 + \lambda + a + \gamma$ and $y_3 = \lambda(x_1 + x_3) + x_3 + y_1$ such that $\lambda = x_1 + \frac{y_1}{x_1}$. Now $f(2P) = (x_3, y_3 + x_3 \cdot S)$. Let $f(P)$ be denoted by $(\tilde{x_1}, \tilde{y_1})$ where $\tilde{x_1} = x_1$ and $\tilde{y_1} = y_1 + x_1 \cdot S$. Now consider $2f(P) = (\tilde{x_3}, \tilde{y_3})$. Then $(\tilde{x_3}, \tilde{y_3})$ satisfies $\tilde{x_3} = \tilde{\lambda}^2 + \tilde{\lambda} + a$ and $\tilde{y_3} = \tilde{\lambda}(\tilde{x_1} + \tilde{x_3}) + \tilde{x_3} + \tilde{y_1}$ where $\tilde{\lambda} = \tilde{x_1} + \frac{\tilde{y_1}}{\tilde{x_1}}$. Observe that $\tilde{\lambda} = \tilde{x_1} + \frac{\tilde{y_1}}{\tilde{x_1}} = x_1 + \frac{y_1 + x_1 \cdot S}{x_1} = x_1 + \frac{y_1}{x_1} + S = \lambda + S$. Thus $\tilde{x_3} = (\lambda + S)^2 + \lambda + S + a = \lambda^2 + \lambda + S^2 + S + a = \lambda^2 + \lambda + a + \gamma = x_3$. Also, $\tilde{y_3} = \tilde{\lambda}(\tilde{x_1} + \tilde{x_3}) + \tilde{x_3} + \tilde{y_1} = (\lambda+S)(x_1+x_3)+x_3+y_1+x_1\cdot S = \lambda(x_1+x_3)+x_3+y_1+S\cdot x_1+S\cdot x_3+x_1\cdot S = \lambda(x_1+x_3)+x_3+y_1+x_3\cdot S = y_3+x_3\cdot S$. Therefore $f(2P) = 2f(P)$.

To establish Item 3). Consider $P, Q \in E_{a+\gamma,b}$ where $P \neq \pm Q$. Suppose $P = (x_1, y_1), Q = (x_2, y_2)$, then $P + Q = (x_3, y_3)$ such that $x_3 = \lambda^2 + \lambda + x_1 + x_2 + a + \gamma$ and $y_3 = \lambda(x_1 + x_3) + y_1 + x_3$ where $\lambda = \frac{y_1+y_2}{x_1+x_2}$. Consider $f(P) = (\tilde{x_1}, \tilde{y_1})$ and $f(Q) = (\tilde{x_2}, \tilde{y_2})$ Then $\tilde{x_i} = x_i$ and $\tilde{y_i} = y_i + x_i \cdot S$ for $i = 1$ and 2. Now consider $(\tilde{x_3}, \tilde{y_3})$ which is $f(P) + f(Q)$. Then $\tilde{x_3} = \tilde{\lambda}^2 + \tilde{\lambda} + \tilde{x_1} + \tilde{x_2} + a$ and $\tilde{y_3} = \tilde{\lambda}(\tilde{x_1} + \tilde{x_3}) + \tilde{y_1} + \tilde{x_3}$. Observe that $\tilde{\lambda} = \frac{\tilde{y_2}+\tilde{y_1}}{\tilde{x_2}+\tilde{x_1}} = \frac{y_2+x_2\cdot S+y_1+x_1\cdot S}{x_2+x_1} = \lambda + S$. Thus $\tilde{x_3} = \tilde{\lambda}^2 + \tilde{\lambda} + \tilde{x_1} + \tilde{x_2} + a = \lambda^2 + S^2 + \lambda + S + x_1 + x_2 + a = \lambda^2 + \lambda + x_1 + x_2 + a + \gamma = x_3$. Now $\tilde{y_3} = \tilde{\lambda}(\tilde{x_1}+\tilde{x_3})+\tilde{y_1}++\tilde{x_3} = (\lambda+S)(x_1+x_3)+y_1+S+x_3 = \lambda(x_1 + x_3) + S \cdot (x_1 + x_3) + y_1 + x_1 \cdot S + x_3 = y_3 + x_3 \cdot S$. Thus $f(P) + f(Q) = f(P + Q)$.

Consequently we see that $E_{a+\gamma,b}$ and $E_{a,b}$ are isomorphic, where the isomorphism $f(x, y) = (x, y + x \cdot S)$. Observe that the inverse map f^{-1} satisfies $f^{-1}(\tilde{x}, \tilde{y}) = (\tilde{x}, \tilde{y} + \tilde{x} \cdot S)$. □

Now if one was asked to compute the scalar multiple kP for $P \in E_{a+\gamma,b}$, then one can compute the scalar multiple $kf(P)$ in $E_{a,b}$. Once $kf(P)$ has been computed, since $kf(P) = f(kP)$, we can compute kP by $kP = f^{-1}(kf(P))$. Thus if $(\tilde{x}, \tilde{y}) = kf(P)$, then $kP = (\tilde{x}, \tilde{y} + \tilde{x} \cdot S)$. The extra complexity cost of computing kP using $kf(P)$ (outside of the scalar arithmetic) is the cost of computing $f(P)$ which will require one field multiplication, and the cost of computing $f^{-1}(kf(P))$ which requires one field multiplication and the cost of computing S, where $S = \mathbf{SOLVE}(\gamma)$.

5 Mapping M to an EC Point

Thus we see that for all $\gamma \in GF(2^n)$ with $Tr(\gamma) = 0$ the curves $E_{a_2+\gamma,a_6}$ E_{a_2,a_6} are isomorphic where the isomorphism $f : E_{a_2+\gamma,a_6} \mapsto E_{a_2,a_6}$ is given by $f(x, y) = (x, y + x \cdot S)$. Further, the inverse $f^{-1} = (x, y + x \cdot S)$ (again $S = \mathbf{Solve}(\gamma)$). Due to the isomorphic properties of f, we see that the prime subgroup in $E_{a_2+\gamma,a_6}$ is isomorphic to the prime subgroup in E_{a_2,a_6}. Moreover, any kP computation on $E_{a_2+\gamma,a_6}$ can be computed using the elliptic curve E_{a_2,a_6} by computing $kf(P)$ and then computing $f^{-1}(kf(P))$.

Suppose M is a non-trivial message (non-zero) belonging to $GF(2^n)$. In practice, M may simply be some binary string which can then be interpreted as an element of $GF(2^n)$ utilizing the polynomial basis representation.

We first assume that E_{a_2,a_6} is a "cryptographically suitably secure elliptic curve", for example it could be one of the NIST recommended elliptic curves [10]. Select $x_1 \in_R GF(2^n) \setminus \{0\}$, such that x_1 is an x-coordinate of some EC point that belongs to the prime subgroup $\mathcal{G}$.[3] As $M \in GF(2^n)$ we compute

$$(\frac{M}{x_1})^2 + \frac{M}{x_1} + x_1 + a_2 + \frac{a_6}{x_1^2} \tag{4}$$

Let γ denote $(\frac{M}{x_1})^2 + \frac{M}{x_1} + x_1 + a_2 + \frac{a_6}{x_1^2}$. Then $Tr(\gamma) = Tr((\frac{M}{x_1})^2 + \frac{M}{x_1} + x_1 + a_2 + \frac{a_6}{x_1^2})$. Thus $Tr((\frac{M}{x_1})^2 + \frac{M}{x_1}) + Tr(x_1 + a_2 + \frac{a_6}{x_1^2}) = 0 + 0 = 0$. Consequently $Tr(\gamma) = 0$. Now consider the elliptic curve $E_{a_2+\gamma,a_6}$. Observe that (x_1, M) belongs to $E_{a_2+\gamma,a_6}$, since substituting (x_1, M) into the expression given in (4) results in $(\frac{M}{x_1})^2 + \frac{M}{x_1} + x_1 + a_2 + \frac{a_6}{x_1^2} = 0$. Further, due to Theorem 1, this curve is isomorphic to E_{a_2,a_6}. Further the map $f : E_{a_2+\gamma,a_6} \longrightarrow E_{a_2,a_6}$ is given by $f(x, y + x \cdot S)$. The inverse mapping $f^{-1} : E_{a_2,a_6} \longrightarrow E_{a_2+\gamma,a_6}$ is $f^{-1}(x, y+x\cdot S)$. As $f(x_1, M)$ belongs to the large prime subgroup of E_{a_2,a_6}, we see that (x_1, M) belongs to the same size prime subgroup of $E_{a_2+\gamma,a_6}$.

We summarize this process in Algorithm 3.

[3] We use the notation $\in_R$ to indicate we select it uniformly random

Thus the prime subgroup $\mathcal{G}_\gamma$ of $E_{a_2+\gamma,a_6}$ is isomorphic to the prime subgroup $\mathcal{G}$ contained in E_{a_2,a_6}. That is if $X \in \mathcal{G}$ the prime subgroup of E_{a_2,a_6} and E_k is a secure cryptosystem over $\mathcal{G}$ then E_k is a secure cryptosystem over $\mathcal{G}_\gamma$ which is the prime subgroup of $E_{a_2+\gamma,a_6}$ under this isomorphism. For example, suppose X represents the plaintext and E_k represents EC ElGamal encryption, and let $Y_{Bob,pub}$ denote Bob's public key. Bob's private key $k_{Bob,priv}$ is such that $Y_{Bob,pub} = k_{Bob,priv}P$, where P is the generator of the prime subgroup $\mathcal{G}$. To encrypt Alice selects an r and computes $C_1 = rP$ and $C_2 = rY_{Bob,pub} + X$. Now if $\tilde{X}$ belongs to $E_{a_2+\gamma,a_6}$ rather than E_{a_2,a_6}, we need to modify the EC ElGamal cryptosystem as follows. We apply the mapping $\rho : E_{a_2,a_6} \mapsto E_{a_2+\gamma,a_6}$ where $\rho(P) = \rho((x,y)) = (x + y \cdot S) = (\tilde{x}, \tilde{y}) = \tilde{P}$. Now to apply the EC ElGamal cryptosystem in $E_{a_2+\gamma,a_6}$, we use $\tilde{Y}_{Bob,pub} = \rho(Y_{Bob,pub})$ as Bob's public key (where the modification is determined by the mapping ρ). The generator is $\tilde{P} = \rho(P)$. Alice selects r randomly and computes $\tilde{C_1} = r\tilde{P}$. Alice then computes $\tilde{C_2} = r\tilde{Y}_{Bob,pub} + \tilde{X}$. The ciphertext $(\tilde{C_1}, \tilde{C_2})$ is sent to Bob. Bob decrypts $(\tilde{C_1}, \tilde{C_2})$ by computing $k\tilde{C_1} - \tilde{C_2}$, the result is the plaintext $\tilde{X}$.

Algorithm 3 A deterministic mapping of message M to an EC point

1: Select $x_1 \in_R GF(2^n)$ such that x_1 is an x-coordinate of an EC point that belongs to the prime subgroup $\mathcal{G}$
2: Let $P_M = (x_1, M)$
3: Set $\gamma \leftarrow (\frac{M}{x_1})^2 + \frac{M}{x_1} + x_1 + a_2 + \frac{a_6}{x_1^2}$ {Then (x_1, M) belongs to $E_{a_2+\gamma,a_6}$}

We use $\in_R$ to represent that we select the element uniformly random.

Notice that γ does not provide any direct information about the message M, due to the fact that x_1 was selected randomly.

Consider a computational comparison of Algorithm 3 and Algorithm 2. Algorithm 3 will require two field multiplications, one field inverse and a selection of a random x-coordinate of a point in the prime subgroup (these three field operations can be used to aid this computation). Algorithm 2 will require at least two field multiplications, one field inverse and one **Solve**. However, due to the probabilistic nature of the algorithm it may repeatedly require these operations to be computed. Also note that as the length of M increases for a fixed elliptic curve E_{a_2,a_6}, the likelihood of Algorithm 2 mapping M into the prime subgroup decreases, whereas Algorithm 3 guarantees the map of any message of length less than the length of a field element.

Concerning the selection of an x_1 such that it is an x-coordinate of some point belonging to the prime subgroup. There is a deterministic way to perform this computation by selecting a random point of the prime subgroup and then using the x-coordinate of this point. Alternately one could select x_1 using the following criteria. If E_{a_2,a_6} has a cofactor of 2 then it is sufficient to select x_1 such that $Tr(x_1) = T(a_2)$ and $Tr(x_1 + a_2 + \frac{a_6}{x_1^2}) = 0$. If E_{a_2,a_6} has a cofactor of 4 then if x is selected as such that $Tr(x) = Tr(a_2)$ and $Tr(\frac{a_6}{x^2}) = 0$ then x is the x-coordinate of a point that belongs to the elliptic curve and this point is a double of an EC point, then either x or $\frac{\sqrt{a_6}}{x}$ is an x-coordinate of a point belonging to the prime subgroup [5].

A natural question to ask: "is it secure to use the y-coordinate as the secret". That is, for a given y-coordinate how many points on the elliptic curve have the same y-coordinate. We observe the following.

Theorem 3. *Let M be a binary string belonging to $GF(2^n)$. Let x_1 be a fixed field element such that $Tr(x_1) = T(a_2)$ and $Tr(x_1 + a_2 + \frac{a_6}{x_1^2}) = 0$. Now consider $\mathcal{A} = \{x : (x, M) \in E_{a_2+\gamma,a_6}\}$, then $|\mathcal{A}| \leq 3$.*

Proof. For a fixed curve $E_{a_2+\gamma,a_6}$, we have $(x_1, M) \in E_{a_2+\gamma,a_6}$. Thus $M^2 + x_1M = x_1^3 + (a_2 + \gamma)x_1^2 + a_6$. Suppose that there exists other x-coordinates α such that $(\alpha, M) \in E_{a_2+\gamma,a_6}$. Then $M^2+\alpha M = \alpha^3+(a_2+\gamma)\alpha^2+a_6$. Taking the difference of the two equations over the field $GF(2^n)$ we get $(x_1 + \alpha)M = x_1^3 + \alpha^3 + (a_2 + \gamma)(x_1^2 + \alpha^2)$ If $\alpha = x_1$ then there is no other solution. Otherwise if $x_1 \neq \alpha$, we can multiply both sides by $(x_1 + \alpha)^{-1}$. So, we have $M = x_1^2 + x_1\alpha + \alpha^2 + (a_2 + \gamma)(x_1 + \alpha)$ Thus $\alpha^2 + \alpha(x_1 + a_2 + \gamma) = M + x_1^2 + (a_2 + \gamma)x_1$. If $x_1 = a_2 + \gamma$, where $x_1 \neq \alpha$, then there is only one α that satisfies this equation. Otherwise, if $x_1 \neq a_2 + \gamma$ then $(x_1 + a_2 + \gamma)^{-1}$ exists. Thus $(\frac{\alpha}{x_1+a_2+\gamma})^2 + \frac{\alpha}{x_1+a_2+\gamma} = \frac{M+x_1^2+(a_2+\gamma)x_1}{x_1+a_2+\gamma}$. Let $\lambda = \frac{\alpha}{x_1+a_2+\gamma}$ then we have $\lambda^2 + \lambda = \frac{M+x_1^2+(a_2+\gamma)x_1}{x_1+a_2+\gamma}$. Observe that this equation has a solution if and only if $Tr(\frac{M+x_1^2+(a_2+\gamma)x_1}{x_1+a_2+\gamma}) = 0$. Of course if there are solutions then there are two solutions, $\frac{\alpha}{x_1+a_2+\gamma}$ and $\frac{\alpha}{x_1+a_2+\gamma} + 1 = \frac{\alpha+x_1+a_2+\gamma}{x_1+a_2+\gamma}$. Thus $|\mathcal{A}| \leq 3$. □

Recall that in the case of ECDH, the secret-key is the x-coordinate of the elliptic curve point. Further, for each x-coordinate of an EC point there are precisely two points belonging to the elliptic curve that have this x-coordinate. Suppose that $E_k(\cdot)$ is a secure cryptosystem for $\mathcal{G}$ a prime subgroup of E_{a_2,a_6}. Let's assume that the cryptosystem E_k represents random mapping from a non-trivial element of the prime subgroup to a nontrivial member of the prime subgroup. Let M represent the secret (i.e. the x-coordinate) and C the ciphertext, then under this traditional method $Prob(M|C) = \frac{2}{|\mathcal{G}|-1}$.

Now consider our method of mapping a secret to an EC point, let M represent the secret and let $P_M = (x_1, M)$. Now let γ denote the field element as indicated in Algorithm 3. Further, let $C = E_K(P_M)$ be the ciphertext generated when we encrypt P_M using E_k as it is generalized in $E_{a_2+\gamma,a_6}$. Again assume that the cryptosystem E_k represents random mapping from a non-trivial element of the prime subgroup to a nontrivial member of the prime subgroup. Then if we apply our mapping to M and assume that E_k as generalized maps a non-trivial point of the prime subgroup to a non-trivial point of the subgroup we have $Prob(M|C) \leq \frac{3}{|\mathcal{G}|-1}$.

6 Performance Considerations

One of the reasons ECC is so popular is its performance advantages in resource starved environments. These advantages include bandwidth, computational complexity of signature generation, and computational complexity of key exchange. This would also include computational complexity of public-key decryption. We now consider how the use of our plaintext mapping to an EC point would affect performance. First it would certainly affect bandwidth.

In comparing the use of our algorithm to ECIES, when transmitting small messages, our algorithm may require slightly more bandwidth, the difference is very slight. Assuming we use EC point compression, our method would require the transmission of two field elements whereas ECIES requires the transmission of one field element and one AES ciphertext message .

Performance could also be affected by the required cryptographic computation. In our algorithm we would be using an elliptic curve $E_{a_2+\gamma,a_6}$ with parameters $a_2+\gamma$ and a_6, rather than the elliptic curve E_{a_2,a_6}, with parameters a_2 and a_6. Often an EC curve is selected to ensure good performance. Many elliptic curves are selected so that $a_2 = 1$. Thus if the parameter $a_2 + \gamma$ is used, then this could affect performance.

The performance effect would depend on many factors including the technique used to perform the necessary EC cryptographic computation (the scalar multiple). Possible considerations could be the representation used to perform the scalar multiple, possible representations include affine representation, projective representation, Montgomery method, and the halving a point algorithm. We limit the discussion to affine representation and projective point representation. For an affine representation, the possible use of an EC parameter of $a_2 + \gamma$ would not affect performance at all.

For a projective point representation, it is very likely that a field multiplication between a field element and EC parameter $a_2 + \gamma$ may be needed, further the number of times it could be needed may be approximately the length of the cryptographic key, denoted by $|key|$. The precise details will depend on the choice of the projective point representation, and possible manipulations of the formulae.

Though multiplications between $a_2 + \gamma$ and a field element may be required, we see that by applying Theorem 2 we would easily mitigate any performance effect. If the encoded plaintext message is an EC point $(x, y) \in E_{a_2+\gamma,a_6}$, then apply the mapping $(x, y) \mapsto (x, y + x \cdot S) = (\tilde{x}, \tilde{y})$, where $S =$**Solve**(γ) and $(x, y + x \cdot S) \in E_{a_2,a_6}$.

Now apply the cryptographic computation to $(\tilde{x}, \tilde{y})$, that is perform the arithmetic in E_{a_2,a_6}, then once the sender has completed the computation, the sender applies the inverse mapping $(\tilde{x}, \tilde{y}) \mapsto (\tilde{x}, \tilde{y} + \tilde{x} \cdot S) \in E_{a_2+\gamma,a_6}$. The result is that two field multiplications and one **Solve** would be needed, rather than $|key|$ many multiplications.

7 Applications

There are several possible applications that would benefit from our message mapping.

In a scenario where a series of private communications need to take place, it makes sense to set up a session key using public-key cryptography, and then use the session key with a symmetric-key cryptosystem. But in an application where only one message needs to be transmitted, and where the message is small, then the use of a symmetric cryptosystem is unnecessary[4]. Rather one could use the public-key cryptosystem to transport the message. ECC would be a practical method to perform the transport as long as it is possible to deterministically map the message to an EC point. Thus our algorithm would allow one to use ECC to transport small messages privately.

A second application is when an entity has created a secret (symmetric)key k and must transmit it to another party. For example, suppose that the secret key k is a some kind of digital rights key and was created by the content provider and that this key must be transmitted to another party by some service provider using the service provider's public-key system. Assuming that the public-key is created using an elliptic curve defined over $GF(2^n)$, the service provider proceeds as follows. Clearly the service provider could map the already computed symmetric key to a y-coordinate of an EC point of some curve $E_{a_2+\gamma,a_6}$ using the techniques that we have described earlier. This scenario could not be supported using the key agreement protocol ECDH, since the symmetric key would have been generated by the content-provider, not the service provider. Further as already mentioned, all prior mapping/embedding techniques were probabilistic in nature, and so to apply these techniques with the EC ElGamal system would not be preferred. Lastly, though the protocol ECIES could be applied in this situation, it requires the use of two cryptosystems the public-key system and a symmetric system, all to ensure the delivery of the digital rights management key.

A third application is as follows. The goal is to apply integrity to the key exchange. Alice gets Bob's public key Y, and generates random secret $x_1 \in GF(2^n)$ such that $Tr(x_1) = Tr(a_2)$ and $Tr(\frac{a_6}{x_1^2}) = 0$ and selects random scalar r and computes rP. She then computes $y_1 = H(x_1||S||r)$ where H is a secure hash function and then computes γ where $\gamma = (\frac{M}{x_1})^2 + \frac{M}{x_1} + x_1 + a_2 + \frac{a_6}{x_1^2}$. Alice then uses EC ElGamal to encrypt $M = (x_1, y_1)$ (using the elliptic curve $E_{a_2+\gamma,a_6}$), the ciphertext is $(rP_\gamma, M + rY_\gamma)$. Bob decrypts in the obvious way to get the secret key x_1 and can apply the hash function to demonstrate the integrity of the key exchange. Clearly one can use some variation of ECIES to provide secret key exchange with an integrity check. Create the key and encrypt it with the symmetric cryptosystem. Further ECIES will require

[4]Voicemail is an example of an application for which only one message needs to be transmitted.

slightly less bandwidth than our method. However our method allows us to use the elliptic curve itself to demonstrate integrity, where ECIES requires the use of a MAC, as well as the use of two cryptosystems.

In a voting application, anonymity needs to be supported. This can be achieved by using a MIX network, where each server will re-encrypt and shuffle ciphertexts and pass them to the next server in the network. ECC provides a good alternative to implement public-key cryptography in a voting application, due to bandwidth. One way to apply ECC in a voting application is to pre-select a point of the prime subgroup for each candidate. As voters vote for a candidate they encrypt the candidates's EC point. However this cannot be done, if one wants to support write-in candidates. To support write-in votes, one could use ECIES, but ECIES cannot be used in an application that requires re-encryption. Consequently to apply ECC in a voting application where anonymity is needed, and where one wants to support write-in votes, then we will need a deterministic method of mapping messages to a point on the elliptic curve. Again our algorithm can be applied to provide such a mapping.

8 Conclusion

In conclusion, we have constructed a deterministic method of mapping an arbitrary message to an EC point, solving an open problem posed in [7]. We noted that that the only known prior technique for mapping plaintext messages to EC points was probabilistic, yielding a small probability of failing. While ECIES is a good technique for encrypting messages, we have seen that there are cases for which it may be desirable to use the public-key cryptosystem to transport the encrypted message, rather than using a hybrid cryptosystem like ECIES.

References

[1] I. F. Blake, N. Smart, and G. Seroussi, *Elliptic Curves in Cryptography*, London Mathematical Society Lecture Note Series, Cambridge University Press, 1999.

[2] P. Gaudry, F. Hess, N. Smart, "Constructive and destructive facets of weil descent on elliptic curves," *Journal of Cryptology*, vol. 15, no. 1, pp. 19-46, Jan. 2002.

[3] Institute for Electrical and Electronics Engineers, *(IEEE) Standard 1363-2000, Standard Specifications for Public Key Cryptography*, Jan. 2000.

[4] M. Jakobsson, "Flash Mixing," *Proceedings of 1999 ACM Symposium on Principles of Distributed Computing (PODC)*, pp. 83-89, 1999,

[5] B. King, "An improved point compression method for elliptic Curves over $GF(2^n)$," *Public Key Cryptography - PKC 2004, 7th International Workshop on Theory and Practice in Public Key Cryptography*, LNCS 2947, pp. 333-345, Springer-Verlag, 2004.

[6] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203-209, 1987.

[7] N. Koblitz, *A Course in Number Theory and Cryptography*, Springer-Verlag, 2^{nd} Edition, New York, 1994.

[8] R. Lidl and H. Niederreiter, *Finite Fields*, Second edition, Cambridge University Press, 1997.

[9] V. S. Miller, "Use of elliptic curves in cryptography," *Advances in Cryptology Crypto '85*, LNCS 218,, pp 417-426 Springer-Verlag, New York, 1985.

[10] NIST, *Recommended Elliptic Curves for Federal Use*. (http://www.nist.gov)

[11] G. Seroussi, *Compact Representation of Elliptic Curve Points over* F_{2^n}, HP Labs Technical Reports, pp. 1-6. (http://www.hpl.hp.com/techreports/98/HPL-98-94R1.html2)

[12] J. Silverman, *The Arithmetic of Elliptic Curves*, Springer-Verlag, New York. 1986.

[13] W. Trappe, and L. C. Washington. *Introduction to Cryptography with Coding Theory*. 2^{nd} Edition, Prentice Hall, 2006.

Brian King received a Ph.D. in mathematics (1990) and a Ph.D. in Computer Science (2000). He is currently an assistant professor of Electrical and Computer Engineering at Indiana Univ. Purdue Univ. Indianapolis (IUPUI). Prior to joining IUPUI he worked in the Security Technologies La at Motorola Research Labs. His research interests include: wireless security, cryptography, threshold cryptography and low-complexity cryptosystems.