A point compression method for elliptic curves defined over $GF(2^n)$

Brian King

Purdue School of Engineering
Indiana Univ. Purdue Univ. at Indianapolis
briking@iupui.edu

Abstract. Here we describe new tools to be used in fields of the form $Gf(2^n)$, that help describe properties of elliptic curves defined over $GF(2^n)$. Further, utilizing these tools we describe a new elliptic curve point compression method, which provides the most efficient use of bandwidth whenever the elliptic curve is defined by $y^2 + xy = x^3 + a_2x^2 + a_6$ and the *trace* of a_2 is zero.

1 Introduction

In [5, 9], Koblitz and Miller independently proposed to use elliptic curves over a finite field to implement cryptographic primitives. The benefits for utilizing elliptic curves as a public key primitive are well recognized: smaller bandwidth, fast key exchange and fast signature generation.

The focus of this paper will be with elliptic curves E defined over a field of the form $GF(2^n)$. In particular our contribution will be the development of new tools to be used in $GF(2^n)$ that help describe elliptic curve properties, as well as we develop a new method for point compression, which is the most efficient point compression described so far.[1] Our result answers a question that Seroussi raised in [12]. Here Seroussi stated that it may be possible to improve on his point compression algorithm but that no known efficient method existed. In addition to the point compression method we provide additional results which were derived from the tools developed for the point compression method. Integral to our work is method of *halving a point*.

2 Background mathematics-binary fields $GF(2^n)$ and elliptic curves

2.1 The trace operator in $GF(2^n)$

The trace function, denoted by Tr, is a homomorphic mapping[2] of $GF(2^n)$ onto $\{0, 1\}$. The trace of an element $\alpha \in GF(2^n)$, denoted by $Tr(\alpha)$ can be

[1] Point compression provides an improvement on bandwidth.
[2] $Tr(\alpha + \beta) = Tr(\alpha) + Tr(\beta)$.

computed (see [15]) as $Tr(\alpha) = \sum_{i=0}^{m-1} \alpha^{2^i}$. (In reality, the trace function can be computed *extremely efficiently*, see *Table 2* in the appendix.) For more information concerning the $Tr()$ operator and its importance see [7]. It can be shown that $Tr()$ is a linear operator which returns a 0 or a 1 and satisfies that $Tr(\alpha^2) = Tr(\alpha)$. In $GF(2^n)$, where n is odd (which is true for all binary fields that we are interested in), then $Tr(1) = 1$ (this can easily be derived given the above equation). Consequently for all $\alpha \in GF(2^n)$ with $Tr(\alpha) = 0$ we have $Tr(\alpha + 1) = 1$ and vice versa. 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 $Tr(b) = 0$ [7]. 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.

2.2 Elliptic curve operation

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

$$y^2 + xy = x^3 + a_2x^2 + a_6 \tag{1}$$

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 $P = (x, y) \neq \mathcal{O}$, $-P = (x, x + y)$
- and for all $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$, both not equal to the identity and $P_1 \neq -P_2$, $P_1 + P_2 = P_3 = (x_3, y_3)$ where $x_3, y_3 \in GF(2^n)$ and satisfy:

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a_2$$

and

$$y_3 = \lambda(x_1 + x_3) + x_3 + y_1$$

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 $P_1 = P_2$.

As stated earlier, the elliptic curve E_{a_2,a_6} is given by the equation $y^2 + xy = x^3 + a_2x^2 + a_6$. 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 $Tr(x + a_2 + \frac{a_6}{x^2}) = 0$. Observe that if β satisfies that $Tr(\beta + a_2 + \frac{a_6}{\beta^2}) = 0$ then there exists a z such that $z^2 + z = \beta + a_2 + \frac{a_6}{\beta^2}$. By setting $y = \beta \cdot z$ we see that $y^2 + \beta y = \beta^3 + a_2\beta^2 + a_6$. Hence $(\beta, y) \in E_{a_2,a_6}$. And so the condition that a nonzero field element β satisfies $Tr(\beta + a_2 + \frac{a_6}{\beta^2}) = 0$ is both a necessary and sufficient condition to determine if the element is the x-coordinate of a point on E_{a_2,a_6}.

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 belong to this subgroup of large prime order.

2.3 Point compression algorithms

In [15] an algorithm for point compression is described. We summarize it as follows. For a party to send a cryptographically relevant elliptic curve point P they need to send an ordered pair. However, rather than sending an ordered pair it is possible to send the x coordinate and one-bit of information. The corresponding y coordinate can be computed using x and this one-bit. This is because, by equation (1) we have $\frac{y^2}{x^2} + \frac{y}{x} = x + a_2 + \frac{a_6}{x^2}$. The problem is that there are two solutions to this equation, one solution has trace 1 and the other solution has trace 0. Consequently the only information concerning y needed to transmitted by the sender is the trace of trace of $\frac{y}{x}$. So if we are given x we can solve for a λ which satisfies $\lambda^2 + \lambda = x + a_2 + \frac{a_6}{x^2}$. One can determine y from λ, x and this one-bit. This method has been standardized in [16, 15] and has been patented. The result is that this method requires $n + 1$ bits to transmit a point on an elliptic curve defined over $GF(2^n)$.

In [2], Seroussi described an even more efficient point compression algorithm. Suppose that $(x_2, y_2) \in E$. Then $Tr(x_2 + a_2 + \frac{a_6}{x_2^2}) = 0$. Again, we assume that (x_2, y_2) is a cryptographically relevant point, that is, it is a point of prime order p. Since (x_2, y_2) is of prime order, it is the double of some point (x_1, y_1). Seroussi, in [2], demonstrated that this implies that $Tr(x_2) = Tr(a_2)$. For completeness (as well as to demonstrate tools that we utilize later) we recreate it here. Suppose (x_2, y_2) is the double of some point $(x_1, y_1) \in E$. Thus $x_2 = x_1^2 + \frac{a_6}{x_1^2}$. Since $(x_1, y_1) \in E$ we have $Tr(x_1 + a_2 + \frac{a_6}{x_1^2}) = 0$. Further since $Tr(x^2) = Tr(x)$ we have

$$Tr(x_1 + a_2 + \frac{a_6}{x_1^2}) = Tr(x_1^2 + a_2 + \frac{a_6}{x_1^2}) = Tr(x_2 + a_2) = 0.$$

Therefore $Tr(x_2) = Tr(a_2)$. It was this property that Seroussi exploits in his compression algorithm. Let $Q = (x_2, y_2)$ be the cryptographically relevant point on the curve E. Consequently Q will belong to a subgroup of prime order and so Q is the double of some point P. Thus $Tr(x_2) = Tr(a_2)$. Given a field element $z = (\zeta_{n-1}, \ldots, \zeta_1, \zeta_0)$ in $GF(2^n)$, it can be represented by n bits. At least one of the bits is used to compute trace, let i denote the smallest index such that ζ_i is used to compute trace (note that it is very likely that $i = 0$), see *Table 2* for examples on how to efficiently compute the trace for the binary fields used in the NIST list of elliptic curves. Suppose $x_2 = (\xi_{n-1}, \xi_{n-2}, ..., \xi_1, \xi_0)$. Then to transmit the x-coordinate x_2 we only need to send $n - 1$ bits, since we can transmit $(\xi_{n-1}, \xi_{n-2}, ...\xi_{i+1}, \xi_{i-1}, ..., \xi_1, \xi_0)$. Now the receiver knows the curve and all of its parameters, thus they know i. Further, the receiver knows that x_2 satisfies $Tr(x_2) = Tr(a_2)$. Consequently the receiver can determine whether ξ_i should be a one or a zero. Once the receiver has x_2, they solve for z such that $z^2 + z = x_2 + a_2 + \frac{a_6}{x_2}$. Then y_2 can be computed by $y_2 = x_2 \cdot z$. The problem again is that there are two solution to this equation in z, one z-solution has trace 1 and the other z-solution has trace 0. Thus the only information needed to transmit y is the trace of the z-value. Hence only one bit needs to be transmitted to communicate y. Therefore Seroussi has demonstrated that only

n bits are needed to be transmit to a receiver a point on the elliptic curve E over $GF(2^n)$.

2.4 Halving a point

In [6], Knudsen introduced the *halving point coordinates* and the halving a point algorithm. Knudsen introduced the concept of halving a point in elliptic curve over $GF(2^n)$ to compute the scalar multiple kP. [3] Knudsen described how to compute $\frac{1}{2}P$ given a point $P = (x, y) \in E$, where P is a double of some point. At the heart of this computation is the representation of a point. Rather that using the affine coordinates of a point $P = (x, y) \in E$, Knudsen represented P as $P = (x, \lambda_P)$ where $\lambda_P = x + \frac{y}{x}$, which we refer to *halving coordinates*. Observe that given x and λ_P, y can be computed since $y = x(x + \lambda_P)$.
Let $Q = (u, \lambda_Q) = \frac{1}{2}P$ where $P = (x, \lambda_P)$. Then Knudsen [6] demonstrated that the following two equations could determine Q; first λ_Q can be determined by solving:

$$\lambda_Q^2 + \lambda_Q = a_2 + x. \tag{2}$$

Once one solves for λ_Q, u can be determined by computing

$$u = \sqrt{u^2} = \sqrt{x(\lambda_Q + 1) + y} = \sqrt{x(\lambda_Q + \lambda_P + x + 1)}. \tag{3}$$

Observe that $Tr(a_2 + x)$ must equal 0, which is true if and only if P is the double of some point, an observation that is used in both [14, 12]. It is trivial to demonstrate that the computed (u, λ_Q) is a “half” of P. Knudsen's algorithm requires one square root, one multiplication, one solve (which is the *halftrace*), and though not illustrated above, *one trace check*. So it will be very efficient.

The primary focus in [6] was with elliptic curves with a cofactor of 2, but Knudsen did not limit his work to only such curves. He provided formulae for the case when the cofactor is 2, as well as when the cofactor is 2^L (where $L > 1$). In [4], an improvement of Knudsen's halving algorithm for curves with a cofactor of 2^L where $L > 1$ was demonstrated.
Integral to our work will be the following algorithms.

SOLVE(s)
  if $Tr(s) \neq 0$
    return **No solution**
  let ζ be an arbitrary solution to
    the equation $w^2 + w = s$
  return ζ

HALF$(P = (x_P, \lambda_P)$
  if $Tr(x_P + a_2) \neq 0$
    return *No half point*
  $\lambda_Q = $ **SOLVE**$(x_P + a_2)$
  $u_Q = \sqrt{x_P(\lambda_Q + \lambda_P + x_P + 1)}$
  return (u_Q, λ_Q)

In the **SOLVE** equation, there are two solutions to the quadratic equation. So when ζ is assigned to be an arbitrary solution it meant that any one of the

[3] Independently, Schroepel [11] also developed the method of halving a point to perform cryptographic computations on an elliptic curve.

two solutions is returned. The Theorem described below demonstrates that not only will the **HALF** algorithm produce a half when the input point that can be halved, but that for any input the **HALF** algorithm will produce the correct output.

Theorem 1. *Let $P \in E$ then*
(i) If $Q = \mathbf{HALF}(P)$ then $Q \in E$ and $2Q = P$.
(ii) If $\mathbf{HALF}(P)$ returns No half point then for all $Q \in E$, $2Q \neq P$.

The proof is left as an exercise.

3 Some observations

Recall that when $(x_2, y_2) \in E$ with $x_2 \neq 0$, we must have $Tr(x_2 + a_2 + \frac{a_6}{x_2^2}) = 0$. Further, if (x_2, y_2) is a double of some point then $Tr(x_2) = Tr(a_2)$. Therefore if (x_2, y_2) is a double of some point then $Tr(\frac{a_6}{x_2^2}) = 0$. This condition can be shown to be both necessary and sufficient to imply that a point is the double of some point in E. The argument is as follows: Suppose $Tr(\frac{a_6}{x_2^2}) = 0$ where $(x_2, y_2) \in E$. Since $Tr(x_2 + a_2 + \frac{a_6}{x_2^2}) = 0$ we see that $Tr(x_2) = Tr(a_2)$. Consider the equation $x^2 + \frac{a_6}{x^2} = x_2$. Observe that if x satisfies this equation then x satisfies $Tr(x^2 + a_2 + \frac{a_6}{x^2}) = Tr(x_2 + a_2) = Tr(x + a_2 + \frac{a_6}{x^2}) = 0$. Thus there exists a y such that $(x, y) \in E$. Now this equation $x^2 + \frac{a_6}{x^2} = x_2$ is solvable, since it reduces to solving $x^4 + x_2 x^2 = a_6$ which is $x_2^2 t^2 + x_2^2 t = a_6$ by letting $x^2 = x_2 t$. This last equation reduces to $t^2 + t = \frac{a_6}{x_2^2}$. Since $\frac{a_6}{x_2^2}$ has trace 0, this is solvable. Once t is found, solve for x by letting $x^2 = x_2 t$ and computing $x = \sqrt{x^2}$.

Consequently the requirement for a point on E to be a double can be solely expressed as a condition existing between x and the parameter a_6. Of course the condition that given x there is some y such that $(x, y) \in E$ can be stated as: $Tr(x + a_2 + \frac{a_6}{x^2}) = 0$. Suppose a_6 is some fixed nonzero field element of $GF(2^n)$, and that x_0 be an arbitrary nonzero field element of $GF(2^n)$ where $Tr(\frac{a_6}{x_0^2}) = 0$. Then x_0 is the x-coordinate for a double of some point for ALL elliptic curves E_{a_2, a_6} which satisfy $Tr(a_2) = Tr(x_0)$.

3.1 A characterization of nonzero elements in $GF(2^n)$

Let a_6 be a fixed nonzero field element in $GF(2^n)$.

Let $x \in GF(2^n)$ with $x \neq 0$, we define the *characterization of* x to be the binary ordered pair $(Tr(x), Tr(\frac{a_6}{x^2}))$. The characterization of x will be helpful to identifying the x-coordinate of points that belong to an elliptic curve or its twist, as well as identifying field elements that are the x-coordinate of points which are doubles. The four possible characterizations are: (1,0), (0,1), (1,1) and (0,0). Those field elements which have characterization of (1,0) and (0,0) represent the field elements which are possible x-coordinates of the double of some elliptic curve point. (Whether a field element is an x-coordinate of a double depends on the trace of a_2. If $Tr(a_2) = 0$ then it would be those field element with character (0,0), whereas if $Tr(a_2) = 1$ then it would be those field element with character (1,0).)

Now consider the element $\frac{\sqrt{a_6}}{x}$. The characterization of $\frac{\sqrt{a_6}}{x}$ is

$$(Tr(\frac{\sqrt{a_6}}{x}), Tr(\frac{a_6}{\frac{\sqrt{a_6}}{x}^2})) = (Tr(\frac{\sqrt{a_6}}{x}), Tr(x^2)).$$

Since $Tr(x^2) = Tr(x)$ we see that the characterization of $\frac{\sqrt{a_6}}{x}$ is equal to $(Tr(\frac{a_6}{x^2}), Tr(x))$ which is a permutation of the characterization of x. The element $\frac{\sqrt{a_6}}{x}$ is of interest for the following reason: Let $T_2 = (0, \sqrt{a_6})$ then independent of the trace value of a_2 we will always have $T_2 \in E_{a_2,a_6}$. Further $T_2 = -T_2$ If x represents the x-coordinate of some point $P \in E_{a_2,a_6}$ then the x-coordinate of $P + T_2$ is $\frac{\sqrt{a_6}}{x}$.

Observe that if x is an x-coordinate of some point on the elliptic curve E_{a_2,a_6} then the characterization of x satisfies $(Tr(x), Tr(\frac{a_6}{x^2})) = (Tr(x), Tr(x) + Tr(a_2))$. Further the sum of the characterization coordinates of x equals $Tr(a_2)$.

We can define an equivalence relation R on $GF(2^n) \setminus \{0\}$ by: for each $x, y \in GF(2^n) \setminus \{0\}$ we say $x\mathsf{R}y$ provided $y = x$ or $y = \frac{\sqrt{a_6}}{x}$. Each equivalence class contains two elements except for the equivalence class for $\sqrt[4]{a_6}$, which possesses one element. Therefore there are $(2^n - 2)/2 + 1 = 2^{n-1}$ equivalence classes for $GF(2^n) \setminus \{0\}$.

For all $i, j \in \{0, 1\}$ we define

$$\mathcal{A}_{(i,j)} = \{x \in GF(2^n) \setminus \{0, \sqrt[4]{a_6}\} : \ x \text{ has characterization } (i,j) \ \}.$$

For all $x \in \left(\mathcal{A}_{(i,j)} \cup \mathcal{A}_{(1+i,1+j)}\right)$, x will be the x-coordinate of some point on the elliptic curve E_{a_2,a_6} where $Tr(a_2) = i + j$. In fact for all $P \in E_{a_2,a_6} \setminus \mathcal{O}$, if $x_P \notin \{0, \sqrt[4]{a_6}\}$ then $x_P \in \left(\mathcal{A}_{(i,j)} \cup \mathcal{A}_{(1+i,1+j)}\right)$. Of course $\mathcal{A}_{(i+j,0)}$ will contain elements which are the x-coordinate of a double of some point in E_{a_2,a_6} and $\mathcal{A}_{(1+i+j,1)}$ will contain elements which are the x-coordinate of a point in E_{a_2,a_6} which are not doubles.

Let $P_1, P_2 \in E_{a_2,a_6}$. Then the following can be established by utilizing the definition of point addition in E_{a_2,a_6}. If $x_{P_1} \in \mathcal{A}_{(i+j,0)}$ and $x_{P_2} \in \mathcal{A}_{(i+j,0)}$ and $P_1 + P_2 \neq \mathcal{O}$ then $x_{P_1+P_2} \in \mathcal{A}_{(i+j,0)}$. If $x_{P_1} \in \mathcal{A}_{(i+j,0)}$ and $x_{P_2} \in \mathcal{A}_{(1+i+j,1)}$ then $x_{P_1+P_2} \in \mathcal{A}_{(1+i+j,1)}$.

Since we have that for each x, the characterization of $\frac{a_6}{\sqrt{x}}$ is the permutation of the characterization of x, this implies that $|\mathcal{A}_{0,1}| = |\mathcal{A}_{1,0}|$ and that both $|\mathcal{A}_{0,0}|$ and $|\mathcal{A}_{1,1}|$ are even. Also since half of the elements in $GF(2^n)$ have trace 0 and the remaining elements have trace 1, we can infer that if $Tr(a_6) = 1$ then the number of elements of $GF(2^n)$ which have trace 0 is $1 + |\mathcal{A}_{0,1}| + |\mathcal{A}_{0,0}|$, whereas the number of elements which have trace 1 is $1 + |\mathcal{A}_{1,0}| + |\mathcal{A}_{1,1}|$. Thus when $Tr(a_6) = 1$ we have $|\mathcal{A}_{0,0}| = |\mathcal{A}_{1,1}|$. If $Tr(a_6) = 0$ then the number of elements of $GF(2^n)$ which have trace 0 is $1 + 1 + |\mathcal{A}_{0,1}| + |\mathcal{A}_{0,0}|$ and the number of elements which have trace 1 is $|\mathcal{A}_{1,0}| + |\mathcal{A}_{1,1}|$. Therefore when $Tr(a_6) = 0$ we see that $|\mathcal{A}_{1,1}| = |\mathcal{A}_{0,0}| + 2$.

Theorem 2. *The number of points on an elliptic curve E_{a_2,a_6} satisfies:*

(i) $|E_{a_2,a_6}| = 1+1+2\cdot|\mathcal{A}_{0,1}|+2\cdot|\mathcal{A}_{1,0}| = 1+1+2\cdot 2\cdot|\mathcal{A}_{1,0}| = 2+4\cdot|\mathcal{A}_{1,0}|$ *provided that* $Tr(a_2)=1$

(ii) $|E_{a_2,a_6}| = 4+4\cdot|\mathcal{A}_{0,0}|$ *provided that* $Tr(a_2)=0$ *and* $Tr(a_6)=1$

(iii) $|E_{a_2,a_6}| = 8+4\cdot|\mathcal{A}_{0,0}|$ *provided that* $Tr(a_2)=0$ *and* $Tr(a_6)=0$

Proof. The proof of (i): Suppose $Tr(a_2) = 1$. The elliptic curve E_{a_2,a_6} will include the point of infinity, and the point $(0,\sqrt{a_6})$. In addition, for each $x \in (\mathcal{A}_{(1,0)} \cup \mathcal{A}_{(0,1)})$ there will exist two values of y such that $(x,y) \in E_{a_2,a_6}$. Lastly recall that $|\mathcal{A}_{(1,0)}| = |\mathcal{A}_{(0,1)}|$. Therefore $|E_{a_2,a_6}| = 1+1+2\cdot|\mathcal{A}_{0,1}|+2\cdot|\mathcal{A}_{1,0}| = 1+1+2\cdot2\cdot|\mathcal{A}_{1,0}| = 2+4\cdot|\mathcal{A}_{1,0}|$.

The proofs of (ii) and (iii) follow from a similar counting argument.

Recall that $|\mathcal{A}_{(i,i)}|$ is even for $i = 0,1$. Therefore an elliptic curve will have a cofactor of 2 iff $Tr(a_2) = 1$ and $1+2\cdot|\mathcal{A}_{(0,1)}|$ is prime. An elliptic curve will have a cofactor of 4 iff $Tr(a_2) = 0$, $Tr(a_6) = 1$ and $1+|\mathcal{A}_{(0,1)}|$ is prime. For $L > 2$, an elliptic curve will have cofactor of 2^L iff $Tr(a_2) = 0$, $Tr(a_6) = 0$ and $1+|\mathcal{A}_{(0,0)}|/2^{L-2}$ is prime.

As described by the above theorem the number of points on an elliptic curve, depends on the characterization of elements in $GF(2^n)$ and the trace of the elliptic curve parameters a_2 and a_6. If we fix the parameter a_6 and vary the parameter a_2 then the characterization for each x in $GF(2^n)$ will be fixed. Therefore we have the following (this same result is provided in [2]).

Theorem 3. *Let* $\gamma \in GF(2^n)$ *such that* $Tr(\gamma) = 0$ *then for all* a_2, a_6 *we have*

$$|E_{a_2+\gamma,a_6}| = |E_{a_2,a_6}|$$

Proof. For a fixed a_2 and a γ with $Tr(\gamma)$=0, we have $Tr(a_2+\gamma) = Tr(a_2)$

A consequence of this theorem is that if E_{a_2,a_6} 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. In [13], it was shown that these curves are isomorphic to each other.

Let $a_2, a_6 \in GF(2^n)$. Then this fixes some elliptic curve E_{a_2,a_6}. Let $\gamma \in GF(2^n)$ where $Tr(\gamma) = 0$. Then [13] has established that both E_{a_2,a_6} and $E_{a_2+\gamma,a_6}$ are isomorphic. But we will see that we can make even more inferences concerning the isomorphism. Suppose that E_{a_2,a_6} has a cofactor of 2^L. Then for all $P = (x,y) \in E_{a_2,a_6}$, there exists a $\zeta \in GF(2^n)$ such that $(x,\zeta) \in E_{a_2+\gamma,a_6}$. It can be shown that $\zeta = y + x\cdot\mathbf{Solve}(\gamma)$. That is, $(x, y+x\cdot\mathbf{Solve}(\gamma)) \in E_{a_2+\gamma,a_6}$. Let $\lambda = \frac{y+x\cdot\mathbf{Solve}(\gamma)}{x}$, then $\lambda^2+\lambda = \frac{y^2}{x^2}+\frac{y}{x}+\mathbf{Solve}^2(\gamma)+\mathbf{Solve}(\gamma) = x + a_2 + \frac{a_6}{x^2} + \gamma = x + (a_2+\gamma) + \frac{a_6}{x^2}$. It is obvious by the tools that we have developed, that the point $P = (x,y) \in E_{a_2,a_6}$ is a double of some point iff the point $(x, y+x\cdot\mathbf{Solve}(\gamma)) \in E_{a_2+\gamma,a_6}$ is a double of some point in $E_{a_2+\gamma,a_6}$. Further whenever $P = (x,y) \in G \subset E_{a_2,a_6}$ (where G is the subgroup of large prime order), then $(x, y+x\cdot\mathbf{Solve}(\gamma))$ belongs to a subgroup of $E_{a_2+\gamma,a_6}$ of the same prime order as G. Thus we see that not only are E_{a_2,a_6} and $E_{a_2+\gamma,a_6}$ isomorphic, when $Tr(\gamma) = 0$, but that this isomorphism is trivial to compute.

Consequently the only relevant parameters to consider for a_2 are 0 and 1 (as long as n is odd). In the WTLS specification of WAP [16], an elliptic curve identified as *curve 4* in the specification, is defined where the a_2 parameter is described in *Table 1* (see below). Since the $Tr(a_2) = 1$, this curve is isomorphic to E_{1,a_6} where the parameter a_6 is given in Table 1. The elliptic curve E_{1,a_6} has a subgroup of large prime order, the same as the order given in Table 1. This subgroup of E_{1,a_6} has a generator $G' = (g'_x, g'_y)$ where $g'_x = G_x$ and $g'_y = G_y + G_x \cdot \mathbf{SOLVE}(072546B5435234A422E0789675F432C89435DE5243)$. From an implementation point of view it is much more efficient to use the elliptic curve E_{1,a_6} then the curve described in Table 1, for whenever one has to perform a field multiplication with a_2, if $a_2 = 1$ then it is free. This type of field multiplication would always be needed when one implements the elliptic curve using a projective point representation. Thus the parameters of curve 4 in WTLS specification should be changed to reflect this.

generating polynomial	$t^{163} + t^8 + t^2 + t + 1$
a_2	$072546B5435234A422E0789675F432C89435DE5242$
a_6	$00C9517D06D5240D3CFF38C74B20B6CD4D6F9DD4D9$
order of the generator $G = (G_x, G_y)$	$040000000000000000000001E60FC8821CC74DAEAFC1$
G_x	$07AF69989546103D79329FCC3D74880F33BBE803CB$
G_y	$01EC23211B5966ADEA1D3F87F7EA5848AEF0B7CA9F$
cofactor	2

Table 1

4 An improved point compression method

Let G denote the set of points of prime order and let $T_2 = (0, \sqrt{a_6})$.

If $Tr(a_2) = 0$ then $x^2 + \frac{a_6}{x^2} = 0$ is solvable, with solution $x = \sqrt[4]{a_6}$. Now characterization of $\sqrt[4]{a_6}$ is $(Tr(\sqrt[4]{a_6}), Tr(\frac{a_6}{(\sqrt[4]{a_6})^2}) = (Tr(a_6), Tr(a_6))$. Thus T_2 is the double of some point with an x-coordinate of $\sqrt[4]{a_6}$. Let Q_1 and Q_3 denote the two points of E which are $\frac{1}{2}T_2$.

Suppose $Tr(a_6) = 1$ and $Tr(a_2) = 0$. Then the x-coordinates of Q_1 and Q_3 have characterization $(Tr(a_6), Tr(a_6)) = (1, 1)$. Therefore both Q_1 and Q_3 are not doubles of any points. Thus we see that there exists a subgroup of order 4 which contains $\mathcal{O}, Q_1, T_2$, and Q_3. Let $P \in G \setminus \{\mathcal{O}\}$, then the characterization of x_P is (0,0) and the characterization of x_{P+T_2} is (0,0). The characterizations of x_{P+Q_1} and x_{P+Q_3} are (1,1), this follows from that fact that both Q_1 and Q_3 are NOT DOUBLEs of any points. Observe that given an point $P = (x, y)$ in G, the field element $\frac{\sqrt{a_6}}{x}$ is the x-coordinate of an EC point which is in the coset $G + T_2$. Now all points $R \in G + T_2$ do have a half but all of its halves do not have a half. Therefore if we found a y such that $R = (x_R, y) \in E$, and then set $\lambda = x_R + \frac{y}{x_R}$ (so that $R = (x_R, \lambda)$ using Knudsen's definition [6]) and compute $(u, \lambda_U) = \mathbf{HALF}(x_R, \lambda)$ then $\mathbf{HALF}(u, \lambda_U) =$ *No half point.*

If $Tr(a_6) = 0$ and $Tr(a_2) = 0$, then the half of T_2 is Q_1 and Q_3, and both Q_1 and Q_3 are doubles. So there exists a subgroup of order 2^{m+1} which contains

Q_1, T_2, Q_3. Thus $\frac{1}{2^m}T_2 \in E$, but $\frac{1}{2^m}T_2$ does not have a half. Again if $P = (x, y) \in G$ then $\frac{\sqrt{a_6}}{x}$ is the x-coordinate of $P + T_2$. If we compute y such that $(\frac{\sqrt{a_6}}{x}, y) \in E$, set $\lambda = \frac{\sqrt{a_6}}{x} + \frac{y}{\frac{\sqrt{a_6}}{x}}$ then repeatedly call the **HALF** function eventually we will arrive at *No half point*, i.e. $\mathbf{HALF}^{m+1}(\frac{\sqrt{a_6}}{x}, \lambda) =$ *No half point.*

4.1 A point compression for E_{a_2,a_6} when $Tr(a_2) = 0$

Let $\alpha \in GF(2^n)$ and represent $\alpha = (\rho_{n-1}, \ldots, \rho_1, \rho_0)$. Let i denote the smallest subscript such that ρ_i is used to compute trace of r (for most fields i will be 0). Let $\zeta = (\xi_{n-1}, ..., \xi_0) \in GF(2^n)$ such that $Tr(\zeta) = 0$ (which equals $Tr(a_2)$=0). If a sender Alice wishes to transmit ζ to the receiver Bob they should send $compress(\zeta) = (\xi_{n-1}, ..\xi_{i+1}, \xi_{i-1}, ..., \xi_0)$ which is merely ζ where we have removed the i^{th} term. If a receiver Bob receives $compress(\zeta)$ then Bob will be able to reconstruct ζ. Since Bob knows all parameters of the elliptic curve he knows both $Tr(a_2) = 0$ and the smallest subscript i which is used to compute the trace. Thus Bob knows which bit ξ_i was omitted, by guessing $\xi_i = 0$ and computing the trace of the corresponding field element, Bob can verify whether his guess was correct. His guess was correct if the trace value equals $Tr(a_2)$. Otherwise, if the trace value doesn't equal $Tr(a_2)$, then Bob knows the correct ζ satisfied $\xi_i = 1$. Thus $n - 1$ bits are required to communicate an element $\zeta \in GF(2^n)$ where $Tr(\zeta) = 0$ and where $Tr(a_2) = 0$.

If a receiver is able to compute the x-coordinate of point P then the receiver will compute y as follows: first compute $z = \mathbf{SOLVE}(x + a_2 + \frac{a_6}{x^2})$ then compute $y = x \cdot z$. The problem is that there are two solutions to $\mathbf{SOLVE}(x + a_2 + \frac{a_6}{x^2})$, one with trace 0 and the other with trace 1. So the sender must communicate the trace of $\frac{y}{x}$ which we will denote as ϵ. If $z = \mathbf{SOLVE}(x + a_2 + \frac{a_6}{x^2})$ and if $Tr(z) = \beta$ then $y = x \cdot z$, else if $Tr(z) \neq \epsilon$ then $y = x \cdot (z + 1)$.

We now describe how to accomplish a point compression of $n - 1$ bits. Let T_2 denote the point $(0, \sqrt{a_6}) \in E$, then T_2 has a half since $Tr(a_2) = 0$. Let $P = (x, y)$ be a cryptographically relevant point on E. Then P belongs to G a subgroup of prime order, thus the trace of x is 0. The goal is that the sender will submit to the receiver $n - 1$ bits such that the receiver will be able to expand these bits to compute P. The sender and the receiver share the elliptic curve parameters, and both know the underlying field. Now for the sender to send $P = (x, y)$, they do the following: If $\frac{y}{x}$ has trace 0 the sender sets $\zeta = x$, else if $Tr(\frac{y}{x}) = 1$ the sender sets $\zeta = \frac{\sqrt{a_6}}{x}$. [4] Then since $Tr(a_2) = 0$ we have $Tr(\zeta) = 0$. Thus to transmit ζ the sender sends $compress(\zeta)$ which is $n - 1$ bits. When the receiver receives $compress(\zeta)$ they will be able to reconstruct ζ as described above, since $Tr(\zeta) = 0$. At this time they compute y by first solving $z = \mathbf{SOLVE}(\zeta + a_2 + \frac{a_6}{\zeta^2})$ where z satisfies $Tr(z) = 0$. They then set $y = \zeta \cdot z$. Since $Tr(a_2) = 0$ there exists an m such that $\frac{1}{2^m}T_2 \in E$ (here $T_2 = (0, \sqrt{a_6})$) but $\frac{1}{2^m}T_2$ does not have a half. Since the receiver knows all elliptic curve parameters they

[4] $\frac{x}{\sqrt{a_6}}$ is the x coordinate of the point $P + T_2$, when $Tr(x) = 0$ we have $Tr(\frac{\sqrt{a_6}}{x}) = 0$.

know m. The receiver computes $\mathbf{HALF}^{m+1}(\zeta, \zeta + \frac{y}{\zeta})$, if a point is returned, then the receiver knows $P = (x, y) = (\zeta, y)$. However if $\mathbf{HALF}^{m+1}(\zeta, \zeta + \frac{y}{\zeta})$ returns *No half point* then $\zeta = \frac{\sqrt{a_6}}{x}$. So they compute x by $x = \frac{\sqrt{a_6}}{\zeta}$. Then they compute $z = \mathbf{SOLVE}(x + a_2 + \frac{a_6}{x^2}) = \mathbf{SOLVE}(\frac{\sqrt{a_6}}{\zeta} + a_2 + \zeta^2)$ but this time they select z so that z has trace 1. Finally they compute y by $y = x \cdot z$. Many of the elliptic curves for which $Tr(a_2) = 0$, will have a cofactor of 4 which implies that m will be 1. That is, if $Tr(a_2) = 0$ and the cofactor of the elliptic curve is 4, then T_2 belongs to a subgroup of order 4, thus $\frac{1}{2}T_2$ exists but $\frac{1}{2^2}T_2$ does not exist. All binary elliptic curves in the NIST recommended list of curves [10] for which $Tr(a_2) = 0$ have cofactors of 4.

Theorem 4. *Let E_{a_2,a_6} be an elliptic curve defined over $GF(2^n)$ where $Tr(a_2) = 0$ then there exists an efficient point compression algorithm that will allow a sender to transmit $n - 1$ bits to send a point on the curve of prime order.*

Consequently, we see that this point compression method requires less bandwidth than the patented compression methods described in [2, 15] whenever $Tr(a_2) = 0$.

4.2 Point compression algorithm for E_{a_2,a_6} where $Tr(a_2) = 1$

Thus we see that if $Tr(a_2) = 0$ there exists a point compression method that is superior to the previous point compression methods. It would be preferred to provide a point compression method which is the most efficient, and which utilizes comparable techniques for all cases. And so we now describe a point compression method for the case $Tr(a_2) = 1$. For the case $Tr(a_2) = 1$ we will demonstrate a method which is as efficient as the method by Seroussi, the benefit is that the form is comparable to the method that we described above.

Let $P = (x, y)$ be a cryptographically relevant point on E. Then P belongs to G a subgroup of prime order. Thus the characterization of x is (0,1). The method is such that the sender will submit to the receiver n bits such that the receiver will be able to expand these bits to compute P. Given x, one computes $z = \mathbf{SOLVE}(x + a_2 + \frac{a_6}{x^2})$ since there are two solutions one needs to know the correct trace value of the z-solution. y then satisfies $y = zx$. To provide a unified approach to point compression we suggest that if $Tr(\frac{y}{x}) = 0$ the sender sets $\zeta = x$, otherwise if $Tr(\frac{y}{x}) = 1$ the sender sets $\zeta = \frac{\sqrt{a_6}}{x}$.

Suppose a sender and a receiver exchange an elliptic curve point. If the receiver receives ζ where $Tr(\zeta) = 0$ then the exchanged point $P = (x, y)$ is such that $x = \zeta$ and y satisfies $Tr(\frac{y}{x}) = 0$. First the receiver computes $\lambda = \mathbf{SOLVE}(\zeta + a_2 + \frac{a_6}{\zeta^2})$ where $Tr(\lambda) = 0$. Then the receiver sets $y = x \cdot \lambda$. If the receiver receives ζ where $Tr(\zeta) = 1$ then the exchanged point $P = (x, y)$ is such that $x = \frac{\sqrt{a_6}}{\zeta}$ and y satisfies $Tr(\frac{y}{x}) = 1$. First the receiver computes $\lambda = \mathbf{SOLVE}(\frac{\sqrt{a_6}}{\zeta} + a_2 + \zeta^2)$ where $Tr(\lambda) = 1$. Then the receiver sets $y = x \cdot \lambda$.

The efficiency (here we measure it in terms of the number of field operations that need to be computed) is as efficient (perhaps slightly more efficient) than

Seroussi's method [12]. In our method the receiver will perform (in the worst case) two trace checks, an inversion, a square, a multiply and a **SOLVE**. The receiver may have precomputed and stored the $\sqrt{a_6}$. Although in [4], it was demonstrated that a square root can be computed as nearly as efficient as a square (even when using a polynomial basis to represent a field element) for many fields $GF(2^n)$. In Seroussi's method a bit needs to be guessed, inserted into the stream, a trace check, a bit may need to be changed, a square, a multiply, an inversion, a **SOLVE**, and one more trace check.

5 Attacking a users key using invalid ECC parameters

Our last observation concerning utilizing the tools that we have developed in this paper, is its use to efficiently check an elliptic curve parameter. It is important that during a key exchange a receiver checks elliptic curve parameters before utilizing these parameters with their private key [1]. One important parameter check is to verify that a received point is a point of prime order. Here we will assume that the sender and receiver are performing some type of elliptic curve key exchange and that the receiver receives a point $J_{received} = (x, y)$. The receiver has private key k and will compute $kJ_{received} = (a, b)$. In the end both receiver and sender will have derived (a, b). Of course they will *hash* a. If the receiver does not check that $J_{received}$ is of prime order then the sender may be able to detect a bit of the receivers key k.

We will describe the attack and the remedy for the case when the elliptic curve parameter a_2 satisfies $Tr(a_2) = 0$. Let G represent the subgroup of E of prime order. The attack made by the sender is as follows. The sender sends a point $J_{received} \in G + T_2$, of course the x-coordinate of $J_{received}$ has trace 0. The only way the receiver can determine that J belongs to the coset $G + T_2$, is to compute $pJ_{received}$ where p is the prime order of G. If the receiver does not check the order of $J_{received}$ then when the receiver computes $kJ_{received}$, if $k_0 = 0$ then $kJ_{received} \in G$, if $k_0 = 1$ then $kJ_{received}$ belongs to the coset $G+T_2$. Thus the low bit of the key is vulnerable to this attack. A solution is that if G is a subgroup of order p then the receiver should compute $pJ_{received}$ to verify that it is the identity $\mathcal{O}$, but this will be at a cost of performance. If an elliptic curve has a cofactor of 2^m (which is true for all curves in [10, 16]), then there is an efficient method which will allow us to distinguish between a point in G and a point in the coset $G + T_2$. The alternative (the efficient check) is to first determine m such that $\frac{1}{2^m}T_2 \in E$ but where $\frac{1}{2^m}T_2$ does not have a half. Then the receiver computes $\mathbf{Half}^{m+1}(x, x + \frac{y}{x})$. If the result is a point then element was of prime order, otherwise it belonged to the coset.

In some cases this parameter check will be trivial. For example suppose that the elliptic curve has a cofactor of 2. Then a parameter check is trivial, simply determine if $(x, y) \in E$ and $Tr(x) = 1$.

6 Conclusion

Our work has provided several new tools in $GF(2^n)$ that provided great insight into elliptic curve defined over $GF(2^n)$. It has provided a new way to view the number of points on an elliptic curve. As well as provide us a mean to choose more efficient elliptic curve parameters (for example curve 4 in the WTLS list). Our main result is new point compression method which is superior to prior methods whenever $Tr(a_2) = 0$. Lastly we have demonstrated how the halving algorithm can be utilized to check elliptic curve parameters.

References

1. Adrian Antipa, Daniel R. L. Brown, Alfred Menezes, Ren Struik, Scott A. Vanstone: "Validation of Elliptic Curve Public Keys". *Public Key Cryptography - PKC 2003* 211-223
2. I.F. Blake, Nigel Smart, and G. Seroussi, *Elliptic Curves in Cryptography.* London Mathematical Society Lecture Note Series. Cambridge University Press, 1999.
3. Darrel Hankerson, Julio Lopez Hernandez and Alfred Menezes. "Software Implementation of Elliptic Curve Cryptography over Binary Fields". In *CHES 2000.* p. 1-24.
4. B. King and B. Rubin. "Revisiting the point halving algorithm". *Technical Report.*
5. Neal Koblitz, *Elliptic curve cryptosystems*, Mathematics of Computation, Vol. 48, No. 177, 1987, 203-209.
6. Erik Woodward Knudsen. "Elliptic Scalar Multiplication Using Point Halving". In *Advances in Cryptology - ASIACRYPT '99.* LNCS Vol. 1716, Springer, 1999, p. 135-149
7. R. Lidl and H. Niederreiter. *Finite Fields*, Second edition, Cambridge University Press, 1997.
8. Alfred Menezes,*Elliptic Curve Public Key Cryptosystems*, Kluwer Academic Publishers, 1993.
9. Victor S. Miller, "Use of Elliptic Curves in Cryptography", In *Advances in Cryptology CRYPTO 1985*,Springer-Verlag, New York, 1985, pp 417-42
10. NIST, *Recommended elliptic curves for federal use*, http://www.nist.gov
11. Rich Schroeppel. "Elliptic Curves: Twice as Fast!". In*Rump session of CRYPTO 2000.*
12. G. Seroussi. "Compact Representation of Elliptic Curve Points over F_{2^n}", *HP Labs Technical Reports* , `http://www.hpl.hp.com/techreports/98/HPL-98-94R1.html` , pg. 1-6.
13. J. Silverman. *The Arithmetic of Elliptic Curves.* Springer-Verlag.New York. 1986.
14. N. P. Smart. A note on the x-coordinate of points on an elliptic curve in characteristic two. Information Processing Letters, 80(?):261–263, October 2001
15. *IEEE P1363 Appendix A.* http://www.grouper.org/groups/1363
16. *WTLS Specification*, http://www.wapforum.org

7 Appendix

7.1 NIST recommended curves in $GF(2^n)$

In July 1999 NIST releases a list of recommended but not required curves to use for Elliptic curve cryptography when dealing with federal agencies. Today several of these curve have been adopted by many standards. Our interest is in those curves over the binary field $GF(2^n)$. The curves listed are: K-163, B-163, K-233, B-233, K-283, B-283, K-409, B-409,K-571, and B-571 where the K-*** refers to a Koblitz curve whose Weierstrass equation if of the form

$$y^2 + xy = x^3 + a_2x^2 + 1$$

and B-*** refer to a "random curve" whose Weierstrass equation is of the form

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

For Koblitz curve K-163 the coefficient $a = 1$, for the remaining Koblitz curves K-233, K-283, K-409, and K-571 the coefficient $a = 0$. Thus K-163 the $Tr(a_2) = 1$ and for the other four Koblitz curves K-233, K-283, K-409, and K-571 the $Tr(a_2) = 0$. The table provided below demonstrate a very efficient way to perform a trace check when utilizing a NIST curve. We have reproduced this table, which was originally given in [4].

Curve types	Generating polynomial	condition for $\mu \in GF(2^n)$ to satisfy $Tr(\mu) = 0$
K-163, B-163	$p(t) = t^{163} + t^7 + t^6 + t^3 + 1$	$\mu_0 = \mu_{157}$
K-233, B-233	$p(t) = t^{233} + t^{74} + 1$	$\mu_0 = \mu_{159}$
K-283, B-283	$p(t) = t^{283} + t^{12} + t^7 + t^5 + 1$	$\mu_0 = \mu_{277}$
K-409, B-409	$p(t) = t^{409} + t^{87} + 1$	$\mu_0 = 0$
K-571, B-571	$p(t) = t^{571} + t^{10} + t^5 + t^2 + 1$	$\mu_0 + \mu_{561} + \mu_{569} = 0$

Table 2