Point Halving on Elliptic Curves over Binary Fields

Kenny C.K. Fong

Centre for Applied Cryptographic Research
University of Waterloo

July 31, 2002

Outline

1. Motivation: ECDSA
2. Background: binary fields, elliptic curves, group law
3. Point halving: algorithm and cost
4. Scalar multiplication using the halve-and-add method
5. Conclusion and remarks

Motivation : ECDSA

- The computationally most expensive operation in the *Elliptic Curve Digital Signature Algorithm* (ECDSA) is **scalar multiplication**.
- Traditionally, the basic technique for scalar multiplication is the double-and-add method.
- A new method for scalar multiplication was independently discovered by Knudsen and Schroeppel in 1999. The idea is to replace all point doublings in the double-and-add method with a faster operation called **point halving**.

Elliptic Curves over Binary Fields

- A binary field $\mathbb{F}_{2^m}$ is a finite field of characteristic 2.
- An elliptic curve E over $\mathbb{F}_{2^m}$ is defined by an equation of the form

$$y^2 + xy = x^3 + ax^2 + b \qquad (1)$$

  where $a, b \in \mathbb{F}_{2^m}$, and $b \neq 0$.
- The set $E(\mathbb{F}_{2^m})$ consists of all points $(x, y) \in \mathbb{F}_{2^m} \times \mathbb{F}_{2^m}$ which satisfy equation (1), together with a special point $\mathcal{O}$ called the *point at infinity*.

Group Law on Elliptic Curves

$E(\mathbb{F}_{2^m})$ forms an (additive) abelian group with the following group law ($\mathcal{O}$ serves as the identity element):

1. $P + \mathcal{O} = \mathcal{O} + P = P$ for all $P \in E(\mathbb{F}_{2^m})$.
2. Let $P = (x, y) \in E(\mathbb{F}_{2^m})$. Denote $-P = (x, x + y) \in E(\mathbb{F}_{2^m})$. Define $P + (-P) = \mathcal{O}$.
3. Let $P_1 = (x_1, y_1), P_2 = (x_2, y_2) \in E(\mathbb{F}_{2^m})$ with $P_1 \neq -P_2$. Define $P_3 = P_1 + P_2 = (x_3, y_3)$ where

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a, \quad y_3 = (x_1 + x_3)\lambda + x_3 + y_1,$$

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

Scalar Multiplication in ECDSA

- NIST has recommended 5 random elliptic curves over binary fields for U.S. Federal Government use.
- In ECDSA, we have to choose a point $G \in E(\mathbb{F}_{2^m})$ of prime order n as a domain parameter. We call this fixed point the *base point* or the *generator*.
- Scalar multiplication in ECDSA is an operation that computes kP where $0 \leq k < n$ and $P \in \langle G \rangle$.

Point Doubling vs Point Halving

- **Point doubling**: given $P = (x, y)$, compute $2P = (u, v)$.
  Let $\lambda = x + y/x$. Calculate:

$$u = \lambda^2 + \lambda + a, \tag{2}$$

$$v = x^2 + u(\lambda + 1). \tag{3}$$

- **Point halving**: given $2P = (u, v)$, compute $P = (x, y)$.
  Solve:

$$\lambda^2 + \lambda = u + a \qquad \text{for } \lambda, \tag{4}$$

$$x^2 = v + u(\lambda + 1) \qquad \text{for } x. \tag{5}$$

Compute: $y = \lambda x + x^2$ (since $\lambda = x + y/x$).

Trace

- Let $c \in \mathbb{F}_{2^m}$. Define $\mathrm{Tr}(c) = c + c^2 + c^{2^2} + \ldots + c^{2^{m-1}}$.
- $\mathrm{Tr}(c) \in \{0, 1\}$ for all $c \in \mathbb{F}_{2^m}$.
  PROOF. $\mathrm{Tr}(c) + (\mathrm{Tr}(c))^2 = 0$.
- The *trace* is a linear function: $\mathrm{Tr}(c + d) = \mathrm{Tr}(c) + \mathrm{Tr}(d)$.
- All NIST-recommended random elliptic curves over binary fields have $\mathrm{Tr}(a) = 1$.
- Let $P = (x, y) \in \langle G \rangle$. Then $\mathrm{Tr}(x) = \mathrm{Tr}(a)$.
  PROOF. $\mathrm{Tr}(x) = \mathrm{Tr}(\lambda^2 + \lambda + a) = \underbrace{\mathrm{Tr}(\lambda + \lambda^2)}_{0} + \mathrm{Tr}(a)$.

Point Halving for $\mathbf{Tr}(a) = 1$ case

1. Solve

$$\widehat{\lambda}^2 + \widehat{\lambda} = u + a$$

   for $\widehat{\lambda}$, obtaining $\widehat{\lambda} = \lambda$ or $\widehat{\lambda} = \lambda + 1$.

2. Consider

$$\widehat{x}^2 = v + u(\widehat{\lambda} + 1).$$

   - $\mathrm{Tr}(x^2) = \mathrm{Tr}(x) = \mathrm{Tr}(a) = 1.$
   - $\mathrm{Tr}(v + u((\lambda + 1) + 1)) = \mathrm{Tr}(v + u(\lambda + 1)) + \underbrace{\mathrm{Tr}(u)}_{1}.$
   - Hence $\mathrm{Tr}(v + u(\widehat{\lambda} + 1))$ identifies λ.

3. Find $x = \sqrt{v + u(\lambda + 1)}$, and then $y = \lambda x + x^2$.

Point Halving: Algorithm and Cost

INPUT: $2P = (u, v)$.

OUTPUT: $P = (x, y)$.

Steps	Cost (B-163)
1. Solve $\widehat{\lambda}^2 + \widehat{\lambda} = u + a$ for $\widehat{\lambda}$.	$\approx 2/3$ field mult
2. Find $T = v + u(\widehat{\lambda} + 1)$.	≈ 1 field mult
3. If $\mathrm{Tr}(T) = 1$ then $\lambda = \widehat{\lambda}$, $x = \sqrt{T}$	Trace $\approx$ free
else $\lambda = \widehat{\lambda} + 1$, $x = \sqrt{T + u}$.	sqrt $\approx 1/2$ field mult
4. Find $y = \lambda x + x^2$.	≈ 1 field mult
5. Return (x, y).	

POINT DOUBLING: 2 mult + 1 inv (affine), 4 mult (projective)

POINT HALVING: ≈ 3 mult

Double-and-add Method

INPUT: $k = (k_t, \ldots, k_1, k_0)_2, P \in E(\mathbb{F}_{2^m})$.
OUTPUT: kP.

1. $Q \longleftarrow \mathcal{O}$.
2. For i from 0 to t do
   - If $k_i = 1$ then $Q \longleftarrow Q + P$.
   - $P \longleftarrow 2P$.
3. Return Q.

Halve-and-add Method

INPUT: $k = (k_t, \ldots, k_1, k_0)_2, P \in E(\mathbb{F}_{2^m})$.
OUTPUT: kP.

1. Solve

$$k = k_t 2^t + \cdots + k_1 t + k_0 \equiv k'_t/2^t + \cdots + k'_1/2 + k'_0 \pmod{n}$$

   for k'_i; i.e., compute

$$2^t k \bmod n = k'_0 2^t + k'_1 2^{t-1} + \cdots + k'_t.$$

2. $Q \longleftarrow \mathcal{O}$.
3. For i from 0 to t do
   - If $k'_i = 1$ then $Q \longleftarrow Q + P$.
   - $P \longleftarrow P/2$.
4. Return Q.

Timings (in μs) for B-163 on 500MHz Celeron

Field operations	
multiplication	2.31
inversion	16.92
I/M	7.32
solve QE	1.55
sqrt	0.82
Curve operations	
point doubling (affine)	28.49
point doubling (projective)	6.42
point halving	3.14
Scalar multiplication	
double-and-add (projective)	1954
halve-and-add	1662

Conclusion and Remarks

- Point halving requires storage for at least 34 field elements.
- Both Knudsen and Schroeppel have got patents pending on point halving.
- Point halving will become much more attractive if $I/M \leq 5$. However, our implementation shows that $I/M > 7$. On the contrary, Schroeppel has claimed $I/M \approx 3$.
- Even if $I/M > 5$, point halving is still attractive if we use w-NAF methods.