Pairings on elliptic curves – parameter selection and efficient computation

Michael Naehrig

Microsoft Research
mnaehrig@microsoft.com

Workshop on Elliptic Curve Computation
Redmond, 19 October 2010
Pairings on elliptic curves
parameter selection and efficient computation

Three parts:

- Pairings and pairing-friendly curves,
- an optimal ate pairing on BN curves using the polynomial parametrization,
- affine coordinates for pairing computation at high security levels.
The embedding degree

Let $E$ be an elliptic curve over $\mathbb{F}_q$ (of characteristic $p$) and

- $n = \#E(\mathbb{F}_q) = q + 1 - t$, $|t| \leq 2\sqrt{q}$,
- $r \mid n$ a large prime divisor of $n$ ($r \neq p, r \geq \sqrt{q}$).

The embedding degree of $E$ with respect to $r$ is the smallest positive integer $k$ with

$$r \mid q^k - 1.$$ 

Then

- $k$ is the order of $q$ modulo $r$,
- $r$-th roots of unity $\mu_r \subseteq \mathbb{F}_{q^k}^*$,
- for $k > 1$, $E[r] \subseteq E(\mathbb{F}_{q^k})$. 
The Tate pairing

The Tate-Lichtenbaum pairing

\[
T_r : E(\mathbb{F}_{q^k})[r] \times E(\mathbb{F}_{q^k})/[r] E(\mathbb{F}_{q^k}) \rightarrow \mathbb{F}_{q^k}^*/(\mathbb{F}_{q^k}^*)^r, \\
(P, Q + [r] E(\mathbb{F}_{q^k})) \mapsto f_{r,P}(D_Q)(\mathbb{F}_{q^k}^*)^r
\]

is a non-degenerate, bilinear map, where

- \(f_{r,P}\) is a function with divisor \((f_{r,P}) = r(P) - r(O)\),
- \(D_Q \sim (Q) - (O)\) has support disjoint from \(\{O, P\}\).

Assume \(k > 1\), can use the reduced Tate pairing

\[
t_r : E(\mathbb{F}_q)[r] \times E(\mathbb{F}_{q^k})[r] \rightarrow \mu_r, \\
(P, Q) \mapsto f_{r,P}(Q)^{q^k - 1}/r.
\]
Computing Miller functions

To compute $f_{m,P}(Q)$, $m \in \mathbb{Z}$, with Miller’s algorithm use

$$f_{2i,P}(Q) = f_{i,P}(Q) \frac{l_{[i]P,[i]P}(Q)}{v_{[2i]P}(Q)}$$

$$f_{i\pm1,P}(Q) = f_{i,P}(Q) \frac{l_{[i]P,\pm P}(Q)}{v_{[i\pm1]P}(Q)}.$$

- square-&-multiply-like loop,
- evaluate at $Q$ on the fly,
- update with fraction of line functions,
- on Edwards curves, use fraction of quadratic and line functions.

Computations are in $E(\mathbb{F}_q)$, $E(\mathbb{F}_{q^k})$ and $\mathbb{F}_{q^k}$. 
Common group choices, Tate and ate pairing

Arguments usually restricted to groups

\[ G_1 = E(\mathbb{F}_{q^k})[r] \cap \ker(\phi_q - [1]) = E(\mathbb{F}_q)[r], \]
\[ G_2 = E(\mathbb{F}_{q^k})[r] \cap \ker(\phi_q - [q]). \]

Get mainly two variants:

\[ \text{reduced Tate pairing} \]
\[ t_r : G_1 \times G_2 \to G_3, \ (P, Q) \mapsto f_{r,P}(Q)^{q^k-1}_r, \]

\[ \text{ate pairing} \ (T = t - 1, \log(T) \lesssim \log(r)/2) \]
\[ a_T : G_2 \times G_1 \to G_3, \ (Q, P) \mapsto f_{T,Q}(P)^{q^k-1}_r. \]

Has more efficient variants: optimal ate pairings that are computed from some \( f_{m,Q}(P) \) with \( \log(m) \approx \log(r)/\varphi(k) \).
Using a twist to represent $G_2$

Let $p > 5$ and $E : y^2 = x^3 + ax + b$.
Here: A twist $E'$ of $E$ is a curve isomorphic to $E$ over $\mathbb{F}_{q^k}$.

- A twist is given by
  \[
  E' : y^2 = x^3 + \left(\frac{a}{\omega^4}\right)x + \left(\frac{b}{\omega^6}\right), \omega \in \mathbb{F}_{q^k}^*
  \]
  with isomorphism $\psi : E' \to E, (x', y') \mapsto (\omega^2 x', \omega^3 y')$.

- If $E'$ is defined over $\mathbb{F}_{q^{k/d}}$ for $d \mid k$, and $\psi$ is defined over $\mathbb{F}_{q^k}$ and no smaller field, $d$ is called the degree of $E'$.

- Possible twist degrees: can have $d = 2$, $d = 4$ (for $b = 0$ only), $d = 3$ and $d = 6$ (both for $a = 0$ only).

- Let $d_0 = 6$ if $a = 0$, let $d_0 = 4$ if $b = 0$, and $d_0 = 2$ otherwise. Then there exists a unique twist $E'$ of degree $d = \gcd(d_0, k)$ with $r \mid \#E'(\mathbb{F}_{q^{k/d}})$. 


Using a twist to represent $G_2$

Let $E'$ be the unique twist of degree $d$ with $r \mid \# E'(\mathbb{F}_{q^k/d})$.

- Let $G'_2 = E'(\mathbb{F}_{q^k/d})[r]$, then $\psi : G'_2 \rightarrow G_2$ is a group isomorphism,
- if $\mathbb{F}_{q^k} = \mathbb{F}_{q^k/d}(\omega)$, $\psi$ is very convenient,
- points in $G_2$ almost have coefficients in subfield $\mathbb{F}_{q^k/d}$.
Minimal requirements for security

- $k$ should be small, but DLPs must be hard enough.

<table>
<thead>
<tr>
<th>Security level (bits)</th>
<th>EC base point order $r$ (bits)</th>
<th>Extension field size of $q^k$ (bits)</th>
<th>ratio $\rho \cdot k$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>NIST</td>
<td>ECRYPT</td>
</tr>
<tr>
<td>80</td>
<td>160</td>
<td>1024</td>
<td>1248</td>
</tr>
<tr>
<td>112</td>
<td>224</td>
<td>2048</td>
<td>2432</td>
</tr>
<tr>
<td>128</td>
<td>256</td>
<td>3072</td>
<td>3248</td>
</tr>
<tr>
<td>192</td>
<td>384</td>
<td>7680</td>
<td>7936</td>
</tr>
<tr>
<td>256</td>
<td>512</td>
<td>15360</td>
<td>15424</td>
</tr>
</tbody>
</table>

NIST/ECRYPT II recommendations

The $\rho$-value of $E$ is defined as $\rho = \log(q) / \log(r)$.
Balanced security

Do not want to waste resources, so balance the security as much as possible.

- If $\rho$ is too large, $q$ is larger than necessary.

- If $\rho_k$ is too large, $q^k$ is larger than necessary.

- If $\rho_k$ is too small, $r$ is larger than necessary.

\[
\log(r) = \rho \log(r)
\]
Pairing-friendly curves

Supersingular curves have small embedding degree ($k \leq 6$, large char $p > 3$: $k \leq 2$ only).

To find ordinary curves with small embedding degree:
Fix $k$ and find primes $r, p$ and an integer $n$ with the following conditions:

1. $n = p + 1 - t$, $|t| \leq 2\sqrt{p}$,
2. $r \mid n$,
3. $r \mid p^k - 1$,
4. $t^2 - 4p = Dv^2 < 0$, $D, v \in \mathbb{Z}$, $D < 0$, $|D|$ small enough to compute the Hilbert class polynomial for $\mathbb{Q}(\sqrt{D})$.

Given such parameters, a corresponding elliptic curve over $\mathbb{F}_p$ can be constructed using the CM method.
## Pairing-friendly curve construction methods

Freeman, Scott, Teske: A taxonomy of pairing-friendly elliptic curves

<table>
<thead>
<tr>
<th>security</th>
<th>construction</th>
<th>curve</th>
<th>$k$</th>
<th>$\rho$</th>
<th>$\rho k$</th>
<th>$d$</th>
<th>$k/d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>128</td>
<td>BN (Ex. 6.8)</td>
<td>$a = 0$</td>
<td>12</td>
<td>1.00</td>
<td>12</td>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Ex. 6.10</td>
<td>$b = 0$</td>
<td>8</td>
<td>1.50</td>
<td>12</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>Freeman (5.3)</td>
<td>$a, b \neq 0$</td>
<td>10</td>
<td>1.00</td>
<td>10</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Constr. 6.7+</td>
<td>$a, b \neq 0$</td>
<td>12</td>
<td>1.75</td>
<td>21</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>192</td>
<td>KSS (Ex. 6.12)</td>
<td>$a = 0$</td>
<td>18</td>
<td>1.33</td>
<td>24</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>KSS (Ex. 6.11)</td>
<td>$b = 0$</td>
<td>16</td>
<td>1.25</td>
<td>20</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Constr. 6.3+</td>
<td>$a, b \neq 0$</td>
<td>14</td>
<td>1.50</td>
<td>21</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>256</td>
<td>Constr. 6.6</td>
<td>$a = 0$</td>
<td>24</td>
<td>1.25</td>
<td>30</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Constr. 6.4</td>
<td>$b = 0$</td>
<td>28</td>
<td>1.33</td>
<td>37</td>
<td>4</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>Constr. 6.24+</td>
<td>$a, b \neq 0$</td>
<td>26</td>
<td>1.17</td>
<td>30</td>
<td>2</td>
<td>13</td>
</tr>
</tbody>
</table>
If \( u \in \mathbb{Z} \) such that

\[
\begin{align*}
p &= p(u) = 36u^4 + 36u^3 + 24u^2 + 6u + 1, \\
n &= n(u) = 36u^4 + 36u^3 + 18u^2 + 6u + 1
\end{align*}
\]

are both prime, then there exists an ordinary elliptic curve

- with equation \( E : y^2 = x^3 + b, \ b \in \mathbb{F}_p \),
- \( r = n = \#E(\mathbb{F}_p) \) is prime, i.e. \( \rho \approx 1 \),
- the embedding degree is \( k = 12 \),
- \( t(u)^2 - 4p(u) = -3(6u^2 + 4u + 1)^2 \),
- there exists a twist \( E' : y^2 = x^3 + b/\xi \) over \( \mathbb{F}_{p^2} \) of degree 6 with \( n \mid \#E'(\mathbb{F}_{p^2}) \).
BN curves
(Barreto-N., 2005)

\[ p = p(u) = 36u^4 + 36u^3 + 24u^2 + 6u + 1, \]
\[ n = n(u) = 36u^4 + 36u^3 + 18u^2 + 6u + 1, \]
\[ E : y^2 = x^3 + b, \]
\[ E' : y^2 = x^3 + b/\xi \]

Thus we can represent \( G_2 \) by \( G'_2 = E'(\mathbb{F}_{p^2})[n] \).

- Replace all points \( R \in G_2 \) by \( R' \in G'_2 \) via \( R = \psi(R') \),
- curve arithmetic over \( \mathbb{F}_{p^2} \) instead of \( \mathbb{F}_{p^{12}} \),
- represent field extensions of \( \mathbb{F}_{p^2} \) using \( \xi \)

\[ \mathbb{F}_{p^{2j}} = \mathbb{F}_{p^2}[X]/(X^j - \xi), \quad j \in \{2, 3, 6\}. \]
An optimal ate pairing on BN curves

Input: $P \in G_1 = E(\mathbb{F}_p)$, $Q = \psi(Q')$, $Q' \in G'_2 \subseteq E'(\mathbb{F}_{p^2})$, $m = 6u + 2 = (1, m_{s-1}, \ldots, m_0)_{\text{NAF}}$.

Output: $a_{\text{opt}}(Q, P)$.

1: $R \leftarrow Q$, $f \leftarrow 1$
2: for $(i \leftarrow s - 1; i \geq 0; i - -)$ do
3: $f \leftarrow f^2 \cdot l_{R,R}(P)$, $R \leftarrow [2]R$
4: if $(m_i = \pm 1)$ then
5: $f \leftarrow f \cdot l_{R,\pm Q}(P)$, $R \leftarrow R \pm Q$
6: end if
7: end for
8: if $u < 0$ then
9: $f \leftarrow 1/f$, $R \leftarrow -R$
10: end if
11: $Q_1 = \phi_p(Q)$, $Q_2 = \phi_{p^2}(Q)$
12: $f \leftarrow f \cdot l_{R,Q_1}(P)$, $R \leftarrow R + Q_1$
13: $f \leftarrow f \cdot l_{R,-Q_2}(P)$, $R \leftarrow R - Q_2$
14: $f \leftarrow f^{p^6-1}$
15: $f \leftarrow f^{p^2+1}$
16: $f \leftarrow f(p^4 - p^2 + 1)/n$
17: return $f$
An optimal ate pairing on BN curves

\textbf{Input:} \( P \in G_1 = E(\mathbb{F}_p), Q = \psi(Q'), Q' \in G'_2 \subseteq E'(\mathbb{F}_{p^2}) \),
\( m = 6u + 2 = (1, m_{s-1}, \ldots, m_0)_{\text{NAF}} \).

\textbf{Output:} \( a_{\text{opt}}(Q, P) \).

1: \( R \leftarrow Q, f \leftarrow 1 \)
2: \textbf{for} \( (i \leftarrow s - 1; i \geq 0; i --) \) \textbf{do}
3: \hspace{1em} \( f \leftarrow f^2 \cdot l_{R,R}(P), R \leftarrow [2]R \)
4: \hspace{1em} \textbf{if} \ (m_i = \pm 1) \textbf{ then}
5: \hspace{2em} \( f \leftarrow f \cdot l_{R,\pm Q}(P), R \leftarrow R \pm Q \)
6: \hspace{1em} \textbf{end if}
7: \textbf{end for}
8: \textbf{if} \ u < 0 \textbf{ then}
9: \hspace{1em} \( f \leftarrow 1/f, R \leftarrow -R \)
10: \textbf{end if}
11: \( Q_1 = \phi_p(Q), Q_2 = \phi_{p^2}(Q) \)
12: \( f \leftarrow f \cdot l_{R,Q_1}(P), R \leftarrow R + Q_1 \)
13: \( f \leftarrow f \cdot l_{R,-Q_2}(P), R \leftarrow R - Q_2 \)
14: \( f \leftarrow f^{p^6-1} \)
15: \( f \leftarrow f^{p^2+1} \)
16: \( f \leftarrow f(p^4-p^2+1)/n \)
17: \textbf{return} \( f \)
The importance of suitable curve parameters

The best performance is obtained by choosing

- $6u + 2$ as sparse as possible,
- $u$ sparse or with a good addition chain,
- $p \equiv 3 \pmod{4}$, so $\mathbb{F}_{p^2} = \mathbb{F}_p(i)$, $i^2 = -1$,
- $\xi$ as "small" as possible to make field extension arithmetic more efficient.

One should also consider non-pairing operations:

- elliptic curve scalar multiplication,
- square root and cube root computation.

Constrained devices might not even need to compute pairings in certain pairing-based protocols.

- In some scenarios, pairings on Edwards curves could be the best choice.
Theorem
Given a BN curve $E : y^2 = x^3 + b$ with $b = N(\xi)$ for $\xi \in \mathbb{F}_{p^2}$, then the sextic twist $E' : y^2 = x^3 + b/\xi$ satisfies $\#E(\mathbb{F}_p) \mid \#E'(\mathbb{F}_{p^2})$.

Suggestions for choosing BN curves:

- Choose low-weight $u$ s.t.
- $6u + 2$ has low weight, and s.t.
- $p \equiv 3 \pmod{4}$, i.e. $\mathbb{F}_{p^2} = \mathbb{F}_p(i)$, $i^2 = -1$,
- choose "small" $\xi = c^2 + id^3$, s.t. $b = c^4 + d^6$ is small,
- get obvious simple generator $P = (-d^2, c^2)$ of $E(\mathbb{F}_p)$,
- and point $P' = (-id, c) \in E'(\mathbb{F}_{p^2})$, that (almost) always gives a generator $Q' = [h]P'$ of $E'(\mathbb{F}_{p^2})[n]$, where $\#E'(\mathbb{F}_{p^2}) = hn$. 
Implementation-friendly BN curves

Example curve:

\[ u = -(2^{62} + 2^{55} + 1), \quad c = 1, \quad d = 1 \]

Then

- \( p \equiv 3 \pmod{4} \),
- \( p \) has 254 bits,
- \( 6u + 2 \) has NAF-weight 5,
- \( E : y^2 = x^3 + 2, \; P = (-1, 1) \),
- \( \xi = 1 + i \),
- \( E' : y^2 = x^3 + (1 - i), \; Q' = [h](-i, 1) \).

http://eprint.iacr.org/2010/429
The pairing algorithm can be improved in all parts by improving arithmetic in $\mathbb{F}_p$.

Can the polynomial shape

$$p = 36u^4 + 36u^3 + 24u^2 + 6u + 1$$

be used to speed up multiplication modulo $p$?

Fan, Vercauteren, Verbauwhede (CHES 2009) demonstrate this for hardware with $u = 2^l + s$, $s$ small.

What about software?
Using the polynomial representation
joint work with P. Schwabe and R. Niederhagen,
inspired by Dan Bernstein’s Curve25519 paper

Consider the ring $R = \mathbb{Z}[x] \cap \mathbb{Z}[\sqrt{6}ux]$ and the element

$$P = 36u^4x^4 + 36u^3x^3 + 24u^2x^2 + 6ux + 1$$
$$= (\sqrt{6}ux)^4 + \sqrt{6}(\sqrt{6}ux)^3 + 4(\sqrt{6}ux)^2 + \sqrt{6}(\sqrt{6}ux) + 1.$$

Then $P(1) = p$.

Represent $f \in \mathbb{F}_p$ as a polynomial $F \in R$

$$F = f_0 + f_1 \cdot \sqrt{6}(\sqrt{6}ux) + f_2 \cdot (\sqrt{6}ux)^2 + f_3 \cdot \sqrt{6}(\sqrt{6}ux)^3$$
$$= f_0 + f_1 \cdot (6u)x + f_2 \cdot (6u^2)x^2 + f_3 \cdot (36u^3)x^3$$

such that $F(1) = f$.

$f$ corresponds to coefficient vector $[f_0, f_1, f_2, f_3], \ f_i \in \mathbb{Z}$. 
Polynomial multiplication and degree reduction

- Polynomial multiplication of $f$ and $g$ gives polynomial with 7 coefficients.

$$f \cdot g = h_0 + h_1 \cdot (6u)x + h_2 \cdot (6u^2)x^2 + h_3 \cdot (36u^3)x^3 + h_4 \cdot (36u^4)x^4 + h_5 \cdot (216u^5)x^5 + h_6 \cdot (216u^6)x^6$$

- Reduce modulo $P$ using

$$(36u^4)x^4 = -(36u^3)x^3 - 4(6u^2)x^2 - (6u)x - 1.$$
Four coefficients are not enough

- 256-bit numbers in 4 coefficients: Each coefficient 64 bits, small multiples in the reduction are larger than 128 bits.
- Easy to realize in hardware, not in software, for software we need more coefficients.
- Idea: Consider $u = v^3$, use 12 coefficients $f_0, \ldots, f_{11}$

$$f = f_0 + 6v f_1 x + 6v^2 f_2 x^2 + 6v^3 f_3 x^3 + 6v^4 f_4 x^4$$
$$+ 6v^5 f_5 x^5 + 6v^6 f_6 x^6 + 36v^7 f_7 x^7 + 36v^8 f_8 x^8$$
$$+ 36v^9 f_9 x^9 + 36v_{10} f_{10} x^{10} + 36v_{11} f_{11} x^{11}.$$ 

$v$ has about 21 bits, product coefficients have about 42 bits.
- Double-precision floats have 53-bit mantissa.
- Use double-precision floats, still some space to add up coefficients and compute small multiples.
Reducing coefficients

- At some point the coefficients will overflow (become larger than 53 bits)
- Need to do coefficient reduction (carry)
- Carry from $f_0$ to $f_1$
  
  \[
  c \leftarrow \text{round}\left(\frac{f_0}{6v}\right)
  \]
  
  \[
  f_0 \leftarrow f_0 - c \cdot 6v
  \]
  
  \[
  f_1 \leftarrow f_1 + c
  \]
- Carry from $f_1$ to $f_2$
  
  \[
  c \leftarrow \text{round}\left(\frac{f_1}{v}\right)
  \]
  
  \[
  f_1 \leftarrow f_1 - c \cdot v
  \]
  
  \[
  f_2 \leftarrow f_2 + c
  \]
- $f_0 \in [-3v, 3v], f_1 \in [-v/2, v/2]$
- Carry from $f_{11}$ goes to $f_0, f_3, f_6,$ and $f_9$
Implementation on a Core 2 processor

- Use fast vector instructions `mulpd` and `addpd`, 2 multiplications/ 2 additions in one instruction, 1 `mulpd` and 1 `addpd` (and one `mov`) per cycle.
- Problem: $\mathbb{F}_p$ arithmetic requires a lot of shuffling, combining etc., Solution: Implement arithmetic in $\mathbb{F}_{p^2}$.
- Use schoolbook multiplication in $\mathbb{F}_{p^2}$: 4 mults. in $\mathbb{F}_p$, squaring in $\mathbb{F}_{p^2}$: 2 multiplications in $\mathbb{F}_p$.
- Perform 2 $\mathbb{F}_p$ multiplications in parallel using vector instructions.
- Only two $\mathbb{F}_p$ polynomial reductions and two coefficient reductions per multiplication in $\mathbb{F}_{p^2}$ (also SIMD).
- To decide where to do a reduction, detect overflows, perform arithmetic on values and in parallel on worst-case values.
Performance results

- On an Intel Core 2 Quad Q6600 (65 nm): 4,134,643 cycles
- Comparison: Fastest published pairing benchmark (on one core) before: 10,000,000 cycles on a Core 2 by Hankerson, Menezes, Scott, 2008, Unpublished: 7,850,000 cyc on Core 2 T5500 (Scott 2010).
- New paper by Beuchat, González Díaz, Mitsunari, Okamoto, Rodríguez-Henríquez, and Teruya (Pairing 2010) claims: 2,330,000 cycles on a Core i7, 2,950,000 cycles on a Core 2 with Visual Studio 2008.

Cycle counts on a Core 2 Q6600 with gcc-4.3.3

<table>
<thead>
<tr>
<th></th>
<th>dclxvi</th>
<th>[BGM+10]</th>
</tr>
</thead>
<tbody>
<tr>
<td>multiplication in $\mathbb{F}_{p^2}$</td>
<td>$\sim 585$</td>
<td>$\sim 588$</td>
</tr>
<tr>
<td>squaring in $\mathbb{F}_{p^2}$</td>
<td>$\sim 359$</td>
<td>$\sim 487$</td>
</tr>
<tr>
<td>optimal ate pairing</td>
<td>$\sim 4,135,000$</td>
<td>$\sim 3,269,000$</td>
</tr>
</tbody>
</table>
Why is our software slower?

[BGM+10] uses Montgomery arithmetic in $\mathbb{F}_p$ and fast $64 \times 64$-bit integer multiplier.

Three reasons why we are slower

1. Restricted choice of $u = v^3$: need more operations in $\mathbb{F}_{p^2}$.
2. Additional coefficient reductions take quite a bit of time.
3. Multiplication is not (much) faster.

Why is our multiplication not faster?

- Always need to perform even number of $\mathbb{F}_p$ multiplications, have to use schoolbook instead of Karatsuba in $\mathbb{F}_{p^2}$, 4 instead of 3 multiplications in $\mathbb{F}_p$.
- Using vector instructions still requires quite some shuffling, overhead: 60 cycles per $\mathbb{F}_{p^2}$ multiplication.
But still...

- Fastest (current) implementation based on double-precision floating-point arithmetic,
- exploits special $p$,
- on Intel (and AMD) processors: integer-based approach (with Montgomery arithmetic) is faster
- But: several architectures have much faster double-precision floating-point than integer arithmetic.

**Paper:** [http://cryptojedi.org/users/peter/#dclxvi](http://cryptojedi.org/users/peter/#dclxvi)

**Software:** [http://cryptojedi.org/crypto/#dclxvi](http://cryptojedi.org/crypto/#dclxvi)

(public domain)
Choose coordinate system for elliptic curve point operations and line function computation,
projective coordinates avoid inversions by doing more of the other operations.

Galbraith (2005): “One can use projective coordinates for the operations in $E(\mathbb{F}_q)$. The performance analysis depends on the relative costs of inversion to multiplication in $\mathbb{F}_q$.... and experiments show that affine coordinates are faster.”

Finite field inversion in prime field very expensive,
for plain ECC over $\mathbb{F}_p$: projective always better,
current speed records for pairings: projective formulas.
Extension field inversions

Quadratic extension:

- \( \mathbb{F}_{q^2} = \mathbb{F}_q(\alpha) \) with \( \alpha^2 = \omega \in \mathbb{F}_q^* \),

\[
\frac{1}{b_0 + b_1 \alpha} = \frac{b_0 - b_1 \alpha}{b_0^2 - b_1^2 \omega} = \frac{b_0}{b_0^2 - b_1^2 \omega} - \frac{b_1}{b_0^2 - b_1^2 \omega} \alpha,
\]

- \( b_0^2 - b_1^2 \omega = N(b_0 + b_1 \alpha) \in \mathbb{F}_q \),

- compute inversion in \( \mathbb{F}_{q^2} \) by inversion in \( \mathbb{F}_q \) and some other operations

\[
I_{q^2} \leq I_q + 2M_q + 2S_q + M(\omega) + \text{sub}_q + \text{neg}_q.
\]

- Assume \( M_{q^2} \geq 3M_q \) and get

\[
R_{q^2} = I_{q^2}/M_{q^2} \leq (I_q/3M_q) + 2 = \frac{R_q}{3} + 2.
\]
Extension field inversions

Degree-\(\ell\) extension:

- generalization of Itoh-Tsujii inversion,
- standard way to compute inverses in optimal extension fields,
- assume \(F_{q^\ell} = F_q(\alpha)\) with \(\alpha^\ell = \omega \in F_q^*\)
- with \(v = (q^\ell - 1)/(q - 1) = q^{\ell-1} + \cdots + q + 1\), compute
  \[
  \beta^{-1} = \beta^{v-1} \cdot \beta^{-v},
  \]
- for \(\beta \in F_{q^\ell}\), \(\beta^v = N(\beta) \in F_q\).

\[
R_{q^\ell} \leq R_q/M(\ell) + C(\ell)
\]

<table>
<thead>
<tr>
<th>(\ell)</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/M((\ell))</td>
<td>1/3</td>
<td>1/6</td>
<td>1/9</td>
<td>1/13</td>
<td>1/17</td>
<td>1/22</td>
</tr>
<tr>
<td>(C(\ell))</td>
<td>3.33</td>
<td>4.17</td>
<td>5.33</td>
<td>5.08</td>
<td>6.24</td>
<td>6.05</td>
</tr>
</tbody>
</table>
Simultaneous inversions
Montgomery’s $n$-th trick...

- Idea: To invert $a$ and $b$, compute $ab$, then $(ab)^{-1}$ and
  
  $$a^{-1} = b \cdot (ab)^{-1}, \quad b^{-1} = a \cdot (ab)^{-1},$$
  
  replace $2I$ by $1I + 3M$.

- In general for $s$ inversions at once: compute $c_i = a_1 \cdots a_i$ for $2 \leq i \leq s$, then $c_s^{-1}$ and
  
  $$c_{s-1}^{-1} = c_s^{-1} a_s, \quad a_{s-1}^{-1} = c_{s-2} c_{s-1}^{-1}, \quad \ldots$$
  
  replace $sI$ by $1I + 3(s - 1)M$.

- Average $I/M$ is
  
  $$\frac{(sI)}{(sM)} = \frac{I}{(sM)} + \frac{3(s - 1)}{s} \leq \frac{R}{s + 3}.$$
Affine coordinates for pairings

Affine coordinates can be better than projective

- if the used implementation has small \( I/M \),
- for ate pairings on curves with larger embedding degree, i.e. at high security levels (the ate pairing needs arithmetic in \( E'\left(\mathbb{F}_{q^{k/d}}\right) \), \( I/M \) gets smaller in larger extensions),
- when high-degree twists are not being used (s.t. \( k/d \) is large),
- for computing several pairings (or products of several pairings) at once on independent point pairs.
Pairings based on Microsoft’s bignum
optimal ate pairing on a 256-bit BN curve

Use MS bignum for

- base field arithmetic ($\mathbb{F}_p$) with Montgomery multiplication,
- 256-bit integers are split into 4 pieces of 64 bits,
- extension fields based on MS bignum field extensions, with inversions based on norm trick.

MS bignum + pairings

- is a C implementation (w/ little bit of assembly for mod mul on AMD64),
- not restricted to specific security level, curves, or processors,
- works under 32-bit and 64-bit Windows.
Fields over 256-bit BN prime field with
\[ p \equiv 3 \pmod{4}, \text{i.e. } \mathbb{F}_{p^2} = \mathbb{F}_p(i), \ i^2 = -1. \]

Timings on a 3.16 GHz Intel Core 2 Duo E8500, 64-bit Windows 7

<table>
<thead>
<tr>
<th></th>
<th>( \mathbb{F}_p )</th>
<th>( \mathbb{F}_{p^2} )</th>
<th>( \mathbb{F}_{p^6} )</th>
<th>( \mathbb{F}_{p^{12}} )</th>
<th>( \mathbb{I} )</th>
<th>( \mathbb{I}/\mathbb{M} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>cyc</td>
<td>414</td>
<td>2122</td>
<td>18544</td>
<td>60967</td>
<td>9469</td>
<td>22.87</td>
</tr>
<tr>
<td>( \mu s )</td>
<td>0.13</td>
<td>0.67</td>
<td>5.81</td>
<td>19.17</td>
<td>2.98</td>
<td></td>
</tr>
<tr>
<td>cyc</td>
<td>414</td>
<td>1328</td>
<td>12929</td>
<td>43081</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \mu s )</td>
<td>0.13</td>
<td>0.42</td>
<td>4.05</td>
<td>13.57</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cyc</td>
<td></td>
<td>11426</td>
<td>40201</td>
<td>103659</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \mu s )</td>
<td></td>
<td>3.65</td>
<td>12.66</td>
<td>32.88</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cyc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>22.87</td>
<td></td>
</tr>
<tr>
<td>( \mu s )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.38</td>
<td></td>
</tr>
<tr>
<td>cyc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.38</td>
</tr>
<tr>
<td>( \mu s )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.38</td>
<td></td>
</tr>
<tr>
<td>cyc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \mu s )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Pairings based on Microsoft’s bignum

Pairings on a 256-bit BN curve with
  - sparse parameter $u$ (HW 7), sparse $6u + 2$ (HW 8).

Timings on a 3.16 GHz Intel Core 2 Duo E8500, 64-bit Windows 7

<table>
<thead>
<tr>
<th>operation</th>
<th>CPU cycles</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Miller loop</td>
<td>7,572,000</td>
<td>2.36 ms</td>
</tr>
<tr>
<td>optimal ate pairing</td>
<td>14,838,000</td>
<td>4.64 ms</td>
</tr>
<tr>
<td>20 opt. ate at once</td>
<td>14,443,000</td>
<td>4.53 ms</td>
</tr>
<tr>
<td>product of 20 opt. ate</td>
<td>4,833,000</td>
<td>1.52 ms</td>
</tr>
<tr>
<td>EC scalar mult in $G_1$</td>
<td>2,071,000</td>
<td>0.64 ms</td>
</tr>
<tr>
<td>EC scalar mult in $G_2$</td>
<td>8,761,000</td>
<td>2.74 ms</td>
</tr>
</tbody>
</table>