New software speed records for cryptographic pairings

Michael Naehrig

Microsoft Research
mnaehrig@microsoft.com

Montreal, 13 April 2010

joint work with Peter Schwabe (TU/e) and Ruben Niederhagen (TU/e, NTU)

Implementing pairings for crypto

For implementing pairings for use in pairing-based cryptographic protocols we usually use variants of the Tate pairing on elliptic curves.
We need:

- suitable curves, i.e. *pairing-friendly elliptic curves*,
- efficient algorithms to compute pairings *as fast as possible*.

Notation

Let E be an elliptic curve over $\mathbb{F}_p$ ($p > 3$ prime) with

- $n = \#E(\mathbb{F}_p) = p + 1 - t, \quad |t| \leq 2\sqrt{p}$,
- $r \mid n$ a large prime divisor of n ($r \neq p$, $r \geq \sqrt{p}$),
- and embedding degree $k > 1$.

The embedding degree of E w.r.t. r is the smallest integer k with $r \mid p^k - 1$.

- $G_1 = E(\mathbb{F}_p)[r]$,
- $G_2 = E(\mathbb{F}_{p^k})[r] \cap \ker(\phi_p - [p])$,
- ate pairing:

$$a_T : G_2 \times G_1 \to G_3,\ a_T(Q, P) = f_{T,Q}(P)^{(p^k-1)/r},$$

$T = t - 1$, $G_3 \subseteq \mathbb{F}_{p^k}^*$ group of r-th roots of unity.

Security and parameter size

- k should be small,
- DLPs must be hard in all three groups G_1, G_2, and G_3,
- for efficiency reasons balance the security.

Security level (bits)	Extension field size p^k (bits)	EC base point order r (bits)	ratio $\rho \cdot k$
	G_3	G_1, G_2	
80	1248	160	7.8
112	2432	224	10.9
128	3248	256	12.7
192	7936	384	20.7
256	15424	512	30.1

ECRYPT II recommendations (2009), $\rho = \log(p)/\log(r)$.

BN curves

(Barreto-N., 2005)

- Security requirements and key size recommendations fix optimal value for $\rho \cdot k$ for given security level.
- BN curves are (nearly) ideal for the 128-bit security level.
- If $u \in \mathbb{Z}$ such that

$$\begin{aligned} p = p(u) &= 36u^4 + 36u^3 + 24u^2 + 6u + 1, \\ n = n(u) &= 36u^4 + 36u^3 + 18u^2 + 6u + 1 \end{aligned}$$

  are both prime, then there exists an ordinary elliptic curve
- $E : y^2 = x^3 + b,\ b \in \mathbb{F}_p$ with
- $r = n = \#E(\mathbb{F}_p)$ prime, i. e. $\rho \approx 1$,
- and embedding degree $k = 12$.

An optimal ate pairing on BN curves $(u > 0)$

Input: $P \in G_1$, $Q \in G_2$, $6u + 2 = (1, m_{s-1}, \ldots, m_0)_2$.
Output: $a_{\text{opt}}(Q, P)$.

$R \leftarrow Q$, $f \leftarrow 1$
for $(i \leftarrow s - 1;\ i \geq 0;\ i - -)$ **do**
    $f \leftarrow f^2 \cdot l_{R,R}(P)$, $R \leftarrow [2]R$
    if $(m_i = 1)$ **then**
        $f \leftarrow f \cdot l_{R,Q}(P)$, $R \leftarrow R + Q$
    end if
end for
$Q_1 = \phi_p(Q)$, $Q_2 = \phi_{p^2}(Q)$
$f \leftarrow f \cdot l_{R,Q_1}(P)$, $R \leftarrow R + Q_1$
$f \leftarrow f \cdot l_{R,-Q_2}(P)$, $R \leftarrow R - Q_2$
$f \leftarrow f^{p^6-1}$
$f \leftarrow f^{p^2+1}$
$f \leftarrow f^{(p^4-p^2+1)/n}$
return f

Using twists of degree 6

There exists a twist $E'/\mathbb{F}_{p^2}$ of degree 6 with

- $n \mid E'(\mathbb{F}_{p^2})$,
- isomorphism

$$\psi : E' \to E, (x', y') \mapsto (\xi^{1/3} x', \xi^{1/2} y'),$$

  where $E' : y^2 = x^3 + b/\xi$.

Thus we can represent G_2 by

$$G_2' = E'(\mathbb{F}_{p^2})[n]$$

and $\psi : G_2' \to G_2$ is a group isomorphism.

- Replace all points $R \in G_2$ by $R' \in G_2'$ via $R = \psi(R')$,
- points are much smaller,
- curve arithmetic over $\mathbb{F}_{p^2}$ instead of $\mathbb{F}_{p^{12}}$.

Modular multiplication

- 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 (2009) demonstrate this for hardware.
- More efficient because uses specially sized multipliers.
- What about software?

Using the polynomial representation

(Inspired by Bernstein's Curve25519 paper)

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

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

Then $P(1) = p$. Represent $f \in \mathbb{F}_p$ by a polynomial $F \in R$ as

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

such that $F(1) = f$.

$$f \leftrightarrow [f_0, f_1, f_2, f_3], \; f_i \in \mathbb{Z}$$

Polynomial multiplication and degree reduction

$$
\begin{aligned}
f &= f_0 + f_1 \cdot (6u)x + f_2 \cdot (6u^2)x^2 + f_3 \cdot (36u^3)x^3, \\
g &= g_0 + g_1 \cdot (6u)x + g_2 \cdot (6u^2)x^2 + g_3 \cdot (36u^3)x^3, \\
f \cdot g &= h_0 + h_1 \cdot (6u)x + h_2 \cdot (6u^2)x^2 + h_3 \cdot (36u^3)x^3 \\
&+ h_4 \cdot (36u^4)x^4 + h_5 \cdot (216u^5)x^5 + h_6 \cdot (216u^6)x^6
\end{aligned}
$$

Reduce modulo P:

$$
\begin{aligned}
(216u^6)x^6 &= -(216u^5)x^5 - 4(36u^4)x^4 - (36u^3)x^3 - (6u^2)x^2 \\
(216u^5)x^5 &= -6(36u^4)x^4 - 4(36u^3)x^3 - 6(6u^2)x^2 - (6u)x \\
(36u^4)x^4 &= -(36u^3)x^3 - 4(6u^2)x^2 - (6u)x - 1
\end{aligned}
$$

$$
\begin{bmatrix} h_0 \\ h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \end{bmatrix}
\rightarrow
\begin{bmatrix} h_0 \\ h_1 \\ h_2 - h_6 \\ h_3 - h_6 \\ h_4 - 4h_6 \\ h_5 - h_6 \\ 0 \end{bmatrix}
\rightarrow
\begin{bmatrix} h_0 \\ h_1 - (h_5 - h_6) \\ h_2 - h_6 - 6(h_5 - h_6) \\ h_3 - h_6 - 4(h_5 - h_6) \\ h_4 - 4h_6 - 6(h_5 - h_6) \\ 0 \\ 0 \end{bmatrix}
\cdots
\begin{bmatrix} h_0 - h_4 + 6h_5 - 2h_6 \\ h_1 - h_4 + 5h_5 - h_6 \\ h_2 - 4h_4 + 18h_5 - 3h_6 \\ h_3 - h_4 + 2h_5 + h_6 \\ 0 \\ 0 \\ 0 \end{bmatrix}
$$

Four coefficients are not enough

Using this for a 256-bit BN prime p:

- element in $\mathbb{F}_p$ is represented by 4 coefficients, some can be larger than 64 bits,
- only have $64 \times 64 \rightarrow 128$ multiplier on amd64 architecture.

Idea: more coefficients and use

- fast double precision floating point arithmetic,
- SIMD instructions (SSE, SSE2, SSE3) to do two 64-bit floating point multiplications or additions at once.

Represent elements in $\mathbb{F}_p$ with coefficients that fit into a 53-bit mantissa of a 64-bit floating point value (double precision).

Representing integers with 12 coefficients

Now assume $u = v^3$ for some $v \in \mathbb{Z}$. Let $\delta = \sqrt[6]{6}$, then $(\delta v x)^3 = \sqrt{6} u x^3$. Consider $R = \mathbb{Z}[x] \cap \overline{\mathbb{Z}}[\delta v x]$, and

$$\begin{aligned} P &= 36u^4x^{12} + 36u^3x^9 + 24u^2x^6 + 6ux^3 + 1 \\ &= 36v^{12}x^{12} + 36v^9x^9 + 24v^6x^6 + 6v^3x^3 + 1 \\ &= (\delta v x)^{12} + \delta^3(\delta v x)^9 + 4(\delta v x)^6 + \delta^3(\delta v x)^3 + 1. \end{aligned}$$

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

$$\begin{aligned} F = \; & f_0 + f_1(6v)x + f_2(6v^2)x^2 + f_3(6v^3)x^3 \\ & + f_4(6v^4)x^4 + f_5(6v^5)x^5 + f_6(6v^6)x^6 + f_7(36v^7)x^7 \\ & + f_8(36v^8)x^8 + f_9(36v^9)x^9 + f_{10}(36v^{10})x^{10} + f_{11}(36v^{11})x^{11} \end{aligned}$$

such that $F(1) = f$.

$$f \leftrightarrow [f_0, f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}, f_{11}]$$

Multiplication and degree reduction

Multiplication of two elements

$$\begin{array}{rcl} f & \leftrightarrow & [f_0, f_1, f_2, f_3, f_4, f_5, f_6, f_7, f_8, f_9, f_{10}, f_{11}] \\ g & \leftrightarrow & [g_0, g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8, g_9, g_{10}, g_{11}] \end{array}$$

gives 23 coefficients. Reduce the degree of the polynomial as before via

$$\begin{array}{rcl} (\delta v x)^{12} & = & -\delta^3(\delta v x)^9 - 4(\delta v x)^6 - \delta^3(\delta v x)^3 - 1 \\ (36v^{12})x^{12} & = & -(36v^9)x^9 - 4(6v^6)x^6 - (6v^3)x^3 - 1 \end{array}$$

By multiplications, additions, reduction etc. the absolute values of the coefficients grow. Need to reduce them once in a while.

Coefficient reduction

$$F = f_0 + f_1(6v)x + f_2(6v^2)x^2 + \dots$$

- replace f_0 by $(f_0 \mod 6v)$ and add quotient to f_1,
- use rounding $r = \mathrm{round}(f_0/(6v))$, then

$$f_0 \leftarrow f_0 - r \cdot (6v),\ f_1 \leftarrow f_1 + r,$$

- $r = \mathrm{round}(f_1/v)$, $f_1 \leftarrow f_1 - r \cdot v$, $f_2 \leftarrow f_2 + r$,
- gives $f_0 \in [-3v, 3v]$, $f_1 \in [-v/2, v/2]$,
- ...
- carry from f_{11} goes to f_0, f_3, f_6, f_9.

Reduced representation and comparison

An element $f \in \mathbb{F}_p$ with representation $[f_0, f_1, \ldots, f_{11}]$ is *reduced* if

$$|f_0|, |f_6| \leq 3v,\ |f_i| \leq v/2,\ i \neq 0, 6.$$

- product of two reduced elements is (almost) reduced after degree and coefficient reduction,
- $[0, 0, \ldots, 0]$ is the unique reduced representation for 0,
- it is even a unique representation for 0 among elements with

$$|f_0|, |f_6| < 6v,\ |f_i| < v,\ i \neq 0, 6.$$

- For comparing two $\mathbb{F}_p$-elements, subtract them and reduce the result.

The curve

- We need u to be a third power and $6u+2$ to have low Hamming weight.
- There are about $12\,000$ primes p that lead to BN curves s.t. u is a third power and p has 256 or 257 bits.
- Lowest Hamming weight $(h(6u+2)=9)$ for

$$\begin{aligned} v &= 1966080 \text{ } (21 \text{ bits}) \\ u=v^3 &= 7599824371187712000 \text{ } (63 \text{ bits}) \\ 6u+2 &= 45598946227126272002 \text{ } (66 \text{ bits}) \\ p &= 36u^4+36u^3+24u^2+6u+1 \text{ } (257 \text{ bits}) \end{aligned}$$

- Curve equation: $E: y^2 = x^3 + 17$ over $\mathbb{F}_p$,
- $\mathbb{F}_{p^2} = \mathbb{F}_p(i)$, $i^2 = -7$,
- Twist: $E' : y^2 = x^3 + 17/\xi$ over $\mathbb{F}_{p^2}$, $\xi = i + 6$.

Arithmetic in $\mathbb{F}_{p^2}$

The (optimal) ate pairing needs fast $\mathbb{F}_{p^2}$-arithmetic.

- Mainly optimize computations in $\mathbb{F}_{p^2}$,
- use SIMD instructions addpd, mulpd,
- can do one mulpd and one addpd in one cycle, i.e. 4 floating point operations,
- only do full reductions when absolutely necessary,
- often short coefficient reduction is sufficient.

High-level implementation

- Field extensions: $\mathbb{F}_{p^{12}}$ is built as a tower on $\mathbb{F}_{p^2}$ as

$$\mathbb{F}_{p^6} = \mathbb{F}_{p^2}(\tau),\ \tau^3 = \xi, \quad \mathbb{F}_{p^{12}} = \mathbb{F}_{p^6}(\omega),\ \omega^2 = \tau.$$

- Miller loop:
  - Jacobian coordinates on twist for curve arithmetic,
  - explicit formulas for line function computation,
  - special multiplication of $\mathbb{F}_{p^{12}}$-element with sparse line function value.
- Final exponentiation:
  - uses method from Scott et al. (2009),
  - hard part done with 3 exponentiations to the power u, and addition-chain to build special exponent (polynomial parametrization),
  - special squaring functions for elements in the cyclotomic subgroup (Granger, Scott, 2009).

Timings

- Optimal ate pairing on a single core of a 2.4 GHz Core 2 Quad Q6600 in less than 4,500,000 cycles (< 2 ms).

no function	63
$\mathbb{F}_{p^2} \times \mathbb{F}_{p^2}$ multiplication	693
$\mathbb{F}_{p^2}$ squaring	531
$\mathbb{F}_{p^2} \times \mathbb{F}_{p}$ multiplication	432
$\mathbb{F}_{p^2}$ short coefficient reduction	135
$\mathbb{F}_{p^2}$ inversion	127,152
Miller loop	2,267,343
optimal ate pairing	4,455,954

- Previous fastest published timings of an implementation by Mike Scott: 10,000,000 cycles on some Core 2 for the R-ate pairing (Hankerson, Menezes, Scott, 2008),
- Mike's implementation now: 7,850,000 cycles on a Core 2 T5500.

Thanks for your attention

- For more details see:
  M. N., Ruben Niederhagen, Peter Schwabe,
  New software speed records for cryptographic pairings
  `http://eprint.iacr.org/2010/186`
- Implementation (Niederhagen/Schwabe):
  `http://www.cryptojedi.org/crypto/#dclxvi`

mnaehrig@microsoft.com

Coefficient reduction

Input: Coefficient vector $(h_0, h_1, \ldots, h_{11}) \in \mathbb{Z}^{12}$.
Output: Reduced coefficient vector $(h'_0, h'_1, \ldots, h'_{11})$.

for $(i \in \{1, 4, 7\})$ **do**
    $r \leftarrow \mathrm{round}(h_i/v), h_i \leftarrow h_i - rv, h_{i+1} \leftarrow h_{i+1} + r$
    $r \leftarrow \mathrm{round}(h_{i+1}/v), h_{i+1} \leftarrow h_{i+1} - rv, h_{i+2} \leftarrow h_{i+2} + r$
end for
$r \leftarrow \mathrm{round}(h_{10}/v), h_{10} \leftarrow h_{10} - rv, h_{11} \leftarrow h_{11} + r$
$r \leftarrow \mathrm{round}(h_{11}/v), h_{11} \leftarrow h_{11} - rv$
$h_9 \leftarrow h_9 - r$, $h_6 \leftarrow h_6 - 4r$, $h_3 \leftarrow h_3 - r$, $h_0 \leftarrow h_0 - r$
$r \leftarrow \mathrm{round}(h_0/(6v)), h_0 \leftarrow h_0 - r \cdot 6v, h_1 \leftarrow h_1 + r$
$r \leftarrow \mathrm{round}(h_3/v), h_3 \leftarrow h_3 - rv, h_4 \leftarrow h_4 + r$
$r \leftarrow \mathrm{round}(h_6/(6v)), h_6 \leftarrow h_6 - r \cdot 6v, h_7 \leftarrow h_7 + r$
$r \leftarrow \mathrm{round}(h_9/v), h_9 \leftarrow h_9 - rv, h_{10} \leftarrow h_{10} + r$
for $(i \in \{1, 4, 7, 10\})$ **do**
    $r \leftarrow \mathrm{round}(h_i/v), h_i \leftarrow h_i - rv, h_{i+1} \leftarrow h_{i+1} + r$
end for
return $(h'_0, h'_1, \ldots, h'_{11})$.