Fast Formulas for Computing Cryptographic Pairings

Craig Costello

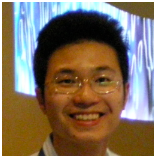
craig.costello@qut.edu.au
Queensland University of Technology

May 28, 2012

Thanks: supervisors and co-authors

Prof. Colin Boyd

Dr. Juanma Gonzalez Nieto



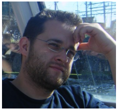
Dr. Kenneth Koon-Ho Wong

Prof. Alice Silverberg

(UC-Irvine)

Prof. Kristin Lauter

(Microsoft Research/UC-San Diego)



A. Prof. Huseyin Hisil

(QUT/Ismir Yasar)

Dr. Douglas Stebila

(QUT)

Prof. Tanja Lange

(T.U. Eindhoven)

Dr. Michael Naehrig

(T.U. Eindhoven/Microsoft)

Thanks: funding and institutes

QUT

Australian Government
Australian Research Council

FULBRIGHT
AUSTRALIAN–AMERICAN FULBRIGHT COMMISSION

Queensland Government

QUT isi Information Security Institute

THE UNIVERSITY OF CALIFORNIA IRVINE
LET THERE BE LIGHT

Microsoft Research

Associated Publications

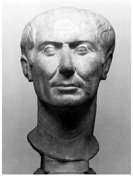
- **C. Costello, H. Hisil, C. Boyd, J. M. Gonzlez Nieto, and K. K. Wong. **Faster pairings on special Weierstrass curves**. *Pairing 2009 - Stanford, USA*.
- ———, T. Lange, and M. Naehrig. **Faster pairing computations on curves with high-degree twists**. *Public Key Cryptography 2010 - Paris, France*.
- ———, C. Boyd, J. M. Gonzlez Nieto, and K. K. Wong. **Avoiding full extension field arithmetic in pairing computations**. *AFRICACRYPT 2010 - Stellenbosch, South Africa*.
- ———, C. Boyd, J. M. Gonzlez Nieto, and K. K. Wong. **Delaying mismatched field multiplications in pairing computations**. *WAIFI 2010 - Istanbul, Turkey*.
- **———, and D. Stebila. **Fixed argument pairings**. *LATINCRYPT 2010 - Puebla, Mexico*.
- ———, K. Lauter and M. Naehrig. **Attractive subfamilies of BLS curves for implementing high-security pairings**. *INDOCRYPT 2011 - Chennai, India*.
- ———, and K. Lauter. **Group law computations on Jacobians of hyperelliptic curves**. *Selected Areas in Cryptography 2011 - Ontario, Canada*.
- ———. **Particularly friendly members of family trees**. *In submission*.

** significant improvements in thesis

1. Motivation
2. Pairings
3. Our work
   - Fast explicit formulas
   - Avoiding $\mathbb{F}_{q^k}$ arithmetic and fixed argument pairings
   - Attractive subfamilies of pairing-friendly curves
4. Summary

Private-key vs. Public-key cryptography

BC - WWII:



Caesar

Mary, Queen of Scots

German Enigma Code

must communicate beforehand

1970's:

Diffie-Hellman-Merkle

Rivest-Shamir-Adleman (RSA)

Cocks

HUGE BREAKTHROUGH: no need for prior communication!!!

Diffie-Hellman (Merkle): a toy example

Public values:

$q = 1000000000000061$ (prime), $g = 832022676086941$ (generator of $\mathbb{Z}_q$).

Secret values:

Alice's secret: a=4275315603725493

Alice computes (public key):

$g^a \bmod q = 9213047582249495$

Bob's secret: b=1083333300180813

Bob computes (public key):

$g^b \bmod q = 9893308140872135$

Bob can compute:

Alice can compute:

$$9893308140872135^a = 8817060794020263 = 9213047582249495^b$$

$$= g^{ab}$$

Secret keys safe as long as discrete log problem (DLP) is hard

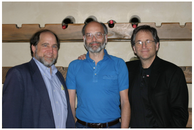
Joint secret safe as long as Diffie-Hellman problem is hard

Modulus (key) sizes: then and now

1970's:

$q =$ 1606938044258990275541962092341162602522202993782792835301301.
(200-bit prime)

NOW:



$q =$
179769313486231590772930519078902473361797697894230657273430081157732675805500963132708477322407536021120113879871393357658789768814416622492847430639474124377767893424865485276302219601246094119453082952085005768838150682342462881473913110540827237163350510684586298239947245938479716304835356329624224137111.
(1024-bit prime)

Elliptic curves cryptography (ECC)

mid 1980's:

Neal Koblitz

Victor Miller

Use elliptic curve (more abstract) groups instead!

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

Subexponential attacks on standard groups don't apply.

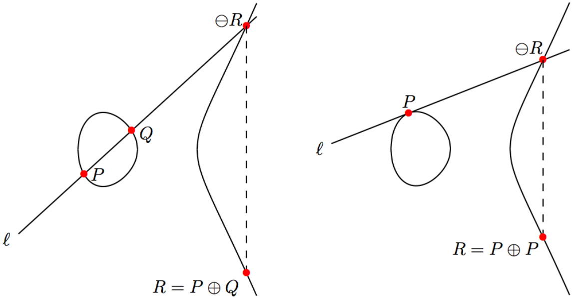
$q =$
179769313486231590772930519078902473361797697894230657273430081157732675805500963132708477322407536021 1
2011387987139335765878976881441662249284743063947412437776789342486548527630221960124609411945308295208
5005768838150682342462881473913110540827237163350510684586298239947245938479716304835356329624224137111.
(1024-bit prime)

vs.

$q =$ 1461501637330902918203684832716283019655932542929
(160-bit prime)

The elliptic curve group law $\oplus$: chord-and-tangent rule

sub $\ell : y = \lambda x + \nu$ into $y^2 = x^3 + ax + b$



(Affine addition) $\lambda = \dfrac{y_Q - y_P}{x_Q - x_P}$; $\nu = y_P - \lambda x_P$;

$$(x_P, y_P) \oplus (x_Q, y_Q) = \left(\lambda^2 - x_P - x_Q, -(\lambda x_R + \nu)\right).$$

(Affine doubling) $\lambda = \dfrac{3x_P^2 + a}{2y_P}$; $\nu = y_P - \lambda x_P$;

$$[2](x_P, y_P) = (x_P, y_P) \oplus (x_P, y_P) = \left(\lambda^2 - 2x_P, -(\lambda x_R + \nu)\right).$$

The pairing explosion

- Pairings are an **extremely powerful primitive** that exist on elliptic curves (more generally abelian varieties)

Boneh - Franklin

- They have been used in the past decade to construct many new protocols / solve many cryptographic problems:
  - Identity-based encryption (IBE), predicate/attribute-based encryption (ABE), hierarchical encryption (HIBE)
  - group/short/ring signatures
  - (partially) homomorphic encryption
  - many many more...

The need for speed...

- First implementation [Menezes 1993]: a few minutes

hundreds of papers on faster pairing computation

- Current record-holding implementation [Aranha *et al.* 2010]: less than a millisecond

Pairings...

What's a pairing

A pairing is a **bilinear** map

$$\begin{aligned} e : \mathbb{G}_1 \times \mathbb{G}_2 &\to \mathbb{G}_T \\ P \times Q &\mapsto e(P, Q) \end{aligned}$$

- **Bilinear:** $\mathbf{e([a]P, [b]Q) = e(P, Q)^{ab} = e([b]P, [a]Q)}$
- $\mathbb{G}_1$ and $\mathbb{G}_2$ are (prime) order r groups on an elliptic curve $E/\mathbb{F}_q$ - also linearly independent
- $\mathbb{G}_T$ is the order r multiplicative group of the extension field $\mathbb{F}_{q^k}$
- All three discrete log problems need to be intractable
- r large, q^k much larger again
- The *embedding degree* $k \in \mathbb{Z}$ plays a vital role in pairing-based cryptography

Pairing-friendly curves

Definition: E is a pairing-friendly curve if...

- k is small (less than 50)
- the prime r dividing $\#E$ has $\rho = \frac{\log_2 q}{\log_2 r} \leq 2$

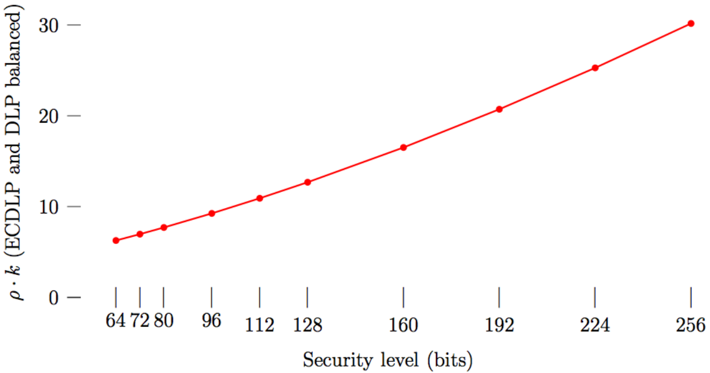
- Balasubramanian-Koblitz '98: $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ defined over $\mathbb{F}_{q^k}$ iff

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

- r huge prime, q huge prime $\rightarrow$ k huge in general
- k being small enough is extremely unlikely in general

- **Moral of the story: pairing-friendly curves are very rare**

Balancing security requires varied k



Example - a Barreto-Naehrig $k = 12$ ($\rho = 1$, $\#E = r$) curve:

$E/\mathbb{F}_q : y^2 = x^3 + 2$

q=2875788016482373728402120498006552346737721998351309856542751926351376964733335173

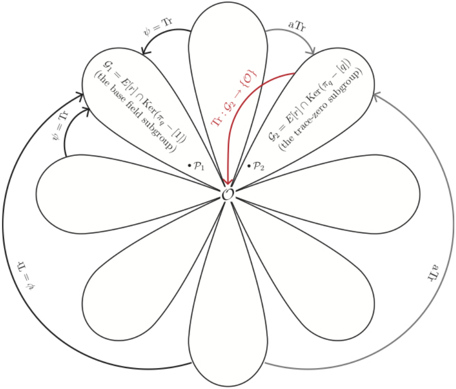
r=2875788016482373728402120498006552346737668371977047909896314898406560560716472109

Perfect for the 128-bit security level: $r \approx 256$-bits, $q^{12} \approx 3072$-bits.

The r-torsion: defining $\mathbb{G}_1$ and $\mathbb{G}_2$

- The entire r-torsion $E[r]$ is defined over $\mathbb{F}_{q^k}$: $E[r] \cong \mathbb{Z}_r \times \mathbb{Z}_r$
- $r+1$ cyclic subgroups of order r, e.g.

r=2875788016482373728402120498006552346737668371977047909896314898406560560716472109



- Different types of pairings depending on available isomorphisms
- Most useful/common/applicable/efficient is Type 3
- Type 3: $\mathbb{G}_1 = E[r] \cap \ker(\pi - [1])$ and $\mathbb{G}_2 = E[r] \cap \ker(\pi - [q])$

What do P, Q and $e(P,Q)$ look like?

- P comes from the "base field subgroup" $\mathbb{G}_1 = E[r] \cap \ker(\pi - [1]) \in E(\mathbb{F}_q)$, e.g.

$$P = (1745887308916193783695853478992570918900044933903060935686246526852538155858841119,$$
$$2444244693111337696007103313755478548273342873814694142226734613453281886385639873)$$

- The pairing $e(P,Q)$ lies in $\mathbb{F}_{q^k}$, e.g.

$$\begin{gathered}
1122570285626574733625914701807031010448154349202286639883584631206978153811383773x^{11} + \\
4238750239433959916714935464715121012491542484851282075784789295502018607201992 9x^{10} + \\
2866010132027332914446708786821217225743489902327108105957890839249244315037275 50x^{9} + \\
2826121779985733015532370468530527995448049975853993544853180851808025552470037363x^{8} + \\
5859399105028672129444237772855943370869839591493129446206447947362850770934475 46x^{7} + \\
1593819990721361493499507049271540572841789 8\ 10748368892500178582235149874536641 21x^{6} + \\
2162793654287391719830538560652021631287992137685407930166074984258729738616236955x^{5} + \\
1649455209892658948773609428850436697294892690168397487540630721684294395713680516x^{4} + \\
1412127150537720237052963479704313079517515741147521064181848925885801017302189791x^{3} + \\
1349010674299277690355298420667754315800686480025643148688438493412221809804707905x^{2} + \\
1769157390330880090254682143693135914705058362971636849346962038147236233875719007x + \\
2866082165939409165611602780404700532164525424945966135019824432074973528752245094
\end{gathered}$$

- Q comes from the "trace zero subgroup" $\mathbb{G}_2 = E[r] \cap \ker(\pi - [q]) \in E(\mathbb{F}_{q^k})$ - e.g. two coordinates of the above size

Using the twisted curve to represent $\mathbb{G}_2$

- Original curve is $E(\mathbb{F}_q) : y^2 = x^3 + ax + b$
- Fortunately, we can employ the use of the twisted curve $E'(\mathbb{F}_{q^{k/d}}) : y^2 = x^3 + ax\omega^4 + b\omega^6$ of $E(\mathbb{F}_q)$
- **Isomorphism** $\Psi : E' \to E; \quad (x', y') \mapsto (x'/\omega^2, y'/\omega^3)$
  $\Psi^{-1} : E \to E'; \quad (x, y) \mapsto (x\omega^2, y\omega^3)$
- Instead of working with $Q \in \mathbb{G}_2 = E(\mathbb{F}_{q^k})$, we can work with $Q' \in \mathbb{G}_2' = E'(\mathbb{F}_{q^{k/d}})$
- Possible degrees of twists $d = \{2, 3, 4, 6\}$ (the bigger the better)
- e.g. instead of working over $\mathbb{F}_{q^{12}}$, we can work over $\mathbb{F}_{q^2}$ using a $d = 6$ sextic twist.

$$363417854035217741503633045279457277830119473964329460726925678347621280259976326x + 5546573986720399889134095745348003378347355016593464532155265683768213902172836 69$$

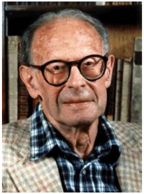
Summary so far

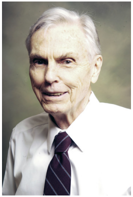
To compute the pairing $e(P, Q)$ of P and Q

- P is a point on E with coordinates in base field
- Q is a point on E with coordinates in extension field $\mathbb{F}_{q^k}$ (but use $Q' \in E'(\mathbb{F}_{q^{k/d}})$).
- $e(P, Q)$ is a value in $\mathbb{F}_{q^k}$

Now, how do we actually compute the pairing?

The Weil and Tate pairings





André Weil   John Tate

Weil pairing (in crypto): $e(P, Q) = \frac{f_{r,P}(Q)}{f_{r,Q}(P)}$;

Tate pairing (in crypto): $e(P, Q) = f_{r,P}(Q)^{(q^k-1)/r}$,

where $f_{r,P}$ is a (unique) degree r function: coefficients dependent on P, evaluated at Q...

e.g. $r = 2875788016482373728402120498006552346737668371977047909896314898406560560716472109$

impossible to compute/store explicitly for cryptographically useful instances!

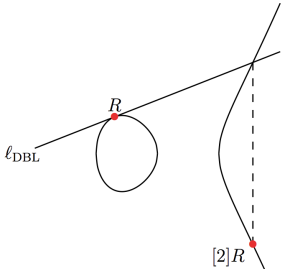
Miller's algorithm

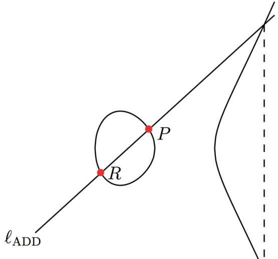
- **Miller 1986**: build the function $f_{r,P}$ by successively squaring (and multiply), but evaluate as you build...

Miller's algorithm for the Tate pairing $f_{r,P}(Q)^{(q^k-1)/r}$

$r = (r_{l-1}, \ldots, r_1, r_0)_2$ initialize: $R = P$, $f = 1$

1. for $i = l-2$ to 0 do //(Miller loop)
   a. i. Compute ℓ_{DBL} in the doubling of R
      ii. $R \leftarrow [2]R$ //(DBL)
      iii. $f \leftarrow f^2 \cdot \ell_{\mathrm{DBL}}(Q)$
   b. if $m_i = 1$ then
      i. Compute ℓ_{ADD} in the addition of $R + P$
      ii. $R \leftarrow R + P$ //(ADD)
      iii. $f \leftarrow f \cdot \ell_{\mathrm{ADD}}(Q)$
2. $f \leftarrow f^{(q^k-1)/r}$. //(Final exponentiation)





Miller's algorithm for the Tate pairing $f_{r,P}(Q)^{(q^k-1)/r}$

$r = (r_{l-1}, \ldots, r_1, r_0)_2$ initialize: $R = P$, $f = 1$

1. for $i = l-2$ to 0 do //(Miller loop)
   a. i. Compute ℓ_{DBL} in the doubling of R
      ii. $R \leftarrow [2]R$ //(DBL)
      iii. $f \leftarrow f^2 \cdot \ell_{\mathrm{DBL}}(Q)$
   b. if $m_i = 1$ then
      i. Compute ℓ_{ADD} in the addition of $R + P$
      ii. $R \leftarrow R + P$ //(ADD)
      iii. $f \leftarrow f \cdot \ell_{\mathrm{ADD}}(Q)$
2. $f \leftarrow f^{(q^k-1)/r}$. //(Final exponentiation)

r=2875788016482373728402120498006552346737668371977047909896314898406560560716472109

r=1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1,
1, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1,
1, 1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1,
1, 0, 0, 1, 0, 1, 1, 0, 1

Miller's algorithm for the optimal ate pairing: $\pi(Q) = [q]Q$

$m = (m_{l-1}, \ldots, m_1, m_0)_2$ initialize: $R = Q$, $f = 1$

1. for $i = l - 2$ to 0 do //(Miller loop)
   a. i. Compute ℓ_{DBL} in the doubling of R
      ii. $R \leftarrow [2]R$ //(DBL)
      iii. $f \leftarrow f^2 \cdot \ell_{\mathrm{DBL}}(P)$
   b. if $m_i = 1$ then
      i. Compute ℓ_{ADD} in the addition of $R + Q$
      ii. $R \leftarrow R + Q$ //(ADD)
      iii. $f \leftarrow f \cdot \ell_{\mathrm{ADD}}(P)$
2. $f \leftarrow f^{(q^k-1)/r}$. //(Final exponentiation)

m=27886288892678111236, i.e. m=(1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0)

Hess

Smart

Vercauteren

$\rightarrow \log_2 m \approx \log_2 r/\varphi(k)$

Our work

Unify and optimise curve operations in Miller loop

a. i. Compute ℓ_{DBL} in the doubling of R

ii. $R \leftarrow [2]R$ //(DBL)

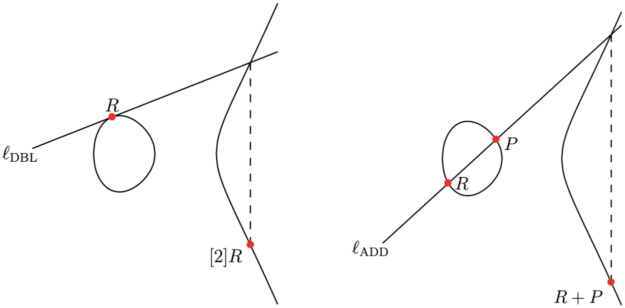
iii. $f \leftarrow f^2 \cdot \ell_{\mathrm{DBL}}(P)$

b. if $m_i = 1$ then

i. Compute ℓ_{ADD} in the addition of $R + Q$

ii. $R \leftarrow R + Q$ //(ADD)

iii. $f \leftarrow f \cdot \ell_{\mathrm{ADD}}(P)$



Do everything on the twisted curve

- ate pairing computation involves moving back between E and E' (different curves)
- problematic for deriving unified formulas for all steps

Theorem (C-Lange-Naehrig-2010)

Let $E/\mathbb{F}_q : y^2 = x^3 + ax + b$ and let $E'/\mathbb{F}_{q^{k/d}} : y^2 = x^3 + a\omega^4 x + b\omega^6$, a degree-$d$ twist of E. Let Ψ be the associated twist isomorphism $\Psi : E' \to E : (x', y') \to (x'/\omega^2, y'/\omega^3)$. Let $P \in G_1$, $Q \in G_2$, and let $Q' = \Psi^{-1}(Q)$ and $P' = \Psi^{-1}(P)$. Let $a_T(Q, P)$ be the ate pairing of Q and P. Then

$$\mathbf{a_T(Q, P)^{\gcd(d,6)} = a_T(Q', P')^{\gcd(d,6)}},$$

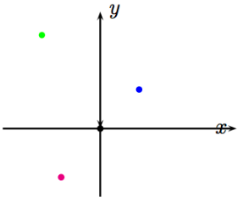
where $a_T(Q', P') = f_{T,Q'}(P')^{(q^k-1)/r}$ uses the same loop parameter as $a_T(Q, P)$ on E, but takes the two twisted points Q' and P' as inputs, instead of Q and P.

Corollary

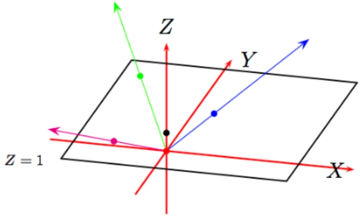
If $a_T(Q, P)$ is bilinear and non-degenerate, then so is $a_T(Q', P')$.

Weierstrass curves for fast pairings

- We found Weierstrass curves $y^2 = x^3 + ax + b$ to perform fastest for pairings (geometric 'chord-and-tangent' description so simple)
- Focussed on tailor-made formulas for curves with high-degree twists
- Different projective spaces perform fastest for different curves



Three points in $\mathbb{A}^2(K)$.



Three lines in $\mathbb{P}^2(K)$.

- Z coordinate sweeps the denominators and avoids inversions

Example: $y^2 = x^3 + b$

- Homogeneous projective coordinates are best: substitute $x = X/Z$ and $y = Y/Z$ to work on $Y^2Z = X^3 + bZ^3$
- Compute $(X_3 : Y_3 : Z_3) = [2](X_1 : Y_1 : Z_1)$ as

$$\begin{aligned} X_3 &= 2X_1Y_1(Y_1^2 - 9bZ_1^2), \\ Y_3 &= Y_1^4 + 18bY_1^2Z_1^2 - 27b^2Z_1^4, \\ Z_3 &= 8Y_1^3Z_1. \end{aligned}$$

- Compute the line function as

$$\ell = 3X_1^2 \cdot x_{P'} - 2Y_1Z_1 \cdot y_{P'} + 3bZ_1^2 - Y_1^2.$$

- A lot of “magic” goes into the above simplification (Gröbner basis reductions, etc)
- Exploit overlaps - altogether costs: **2m + 7s**

Comparisons with previous best formulas

Curve Curve order Twist deg.	**Record**	DBL ADD mADD	**Prev.** **Record**	DBL ADD mADD
$y^2 = x^3 + ax$ - $d = 2, 4$	C-Lange- Naehrig'10 $\mathcal{W}_{(1,2)}$	**2m** + **8s** **12m** + 7**s** **9m** + 5**s**	Ionica-Joux Arene *et al.* $\mathcal{J}$	**1m** + **11s** **10m** + 6**s** **7m** + 6**s**
$y^2 = x^3 + c^2$ $3 \mid \#E$ $d = 2, 6$	C-Hisil-Boyd- Gonzalez Nieto-Wong'09 $\mathcal{P}$	**3m** + **5s** **14m** + 2**s** **10m** + 2**s**	Arene *et al.* $\mathcal{P}$	**3m** + **8s** **10m** + 6**s** **7m** + 6**s**
$y^2 = x^3 + b$ $3 \nmid \#E$ $d = 2, 6$	C-Lange- Naehrig'10 $\mathcal{P}$	**2m** + **7s** **14m** + 2**s** **10m** + 2**s**	Arene *et al.* $\mathcal{J}$	**3m** + **8s** **10m** + 6**s** **7m** + 6**s**
$y^2 = x^3 + b$ - $d = 3$	C-Lange- Naehrig'10 $\mathcal{P}$	**6m** + **7s** **16m** + 3**s** **13m** + 3**s**	El Mrabet *et al.* $\mathcal{P}$	**8m** + **9s** *ADD/mADD* *not reported*

Avoiding extension field arithmetic

a. i. Compute ℓ_{DBL} in the doubling of R
ii. $R \leftarrow [2]R$ //(DBL)
iii. $f \leftarrow f^2 \cdot \ell_{\mathrm{DBL}}(Q)$

b. if $m_i = 1$ then
i. Compute ℓ_{ADD} in the addition of $R + P$
ii. $R \leftarrow R + P$ //(ADD)
iii. $f \leftarrow f \cdot \ell_{\mathrm{ADD}}(Q)$

- $\mathbb{F}_{q^k}$ operations most costly in the Miller loop
- Our prior work saved operations that occur in $\mathbb{F}_{q^{k/d}} \subset \mathbb{F}_{q^k}$
- C-Boyd-Gonzalez Nieto-Wong (two papers) looked at avoiding the arithmetic that hurts most...
- Return to Tate setting (for now): $P \in \mathbb{F}_q$ is first argument

Avoiding extension field arithmetic

iterate through...

i. $f \leftarrow f^2 \cdot \ell_{\mathrm{DBL}}(Q)$

ii. *if* $m_i = 1$ *then* $f \leftarrow f \cdot \ell_{\mathrm{ADD}}(Q)$

- The "updates" look like $\ell : \ell_x \cdot x_Q + \ell_y \cdot y_Q + \ell_0$
- The $\ell_x, \ell_y, \ell_0 \in \mathbb{F}_q$, whilst $x_Q, y_Q \in \mathbb{F}_{q^k}$
- Consider leaving ℓ unevaluated, and operating on it before touching Q (a lot more operations in $\mathbb{F}_q$, but saves operations in $\mathbb{F}_{q^k}$)
- e.g. $k = 12$, $\mathbb{F}_{q^k}$ mul costs 54 $\mathbb{F}_q$ muls
- e.g. merging two consecutive doublings:

$$
\begin{aligned}
(\ell_x \cdot x_Q + \ell_y \cdot y_Q + \ell_0)^2 \cdot (\ell'_x \cdot x_Q + \ell'_y \cdot y_Q + \ell'_0) \\
= \hat{\ell_{3,0}} \cdot x_Q^3 + \hat{\ell_{2,0}} x_Q^2 + \hat{\ell_{1,0}} x_Q + \hat{\ell_{2,1}} x_Q^2 y_Q + \hat{\ell_{1,2}} x_Q y_Q^2 \\
+ \hat{\ell_{1,1}} x_Q y_Q + \hat{\ell_{0,3}} x_Q^3 + \hat{\ell_{0,2}} x_Q^2 + \hat{\ell_{0,1}} x_Q + \hat{\ell_{0,0}}
\end{aligned}
$$

- actually turns out so much better

$$\ell_{2,0} x_Q^2 + \ell_{1,1} x_Q y_Q + \ell_{1,0} x_Q + \ell_{0,1} y_Q + \ell_{0,0}$$

Quadruple-and-add on $y^2 = x^3 + b$

- Compute $(X_3 : Y_3 : Z_3) = [4](X_1 : Y_1 : Z_1)$ on $Y^2Z = X^3 + bZ^3$ and the update function is...

$$\ell_{2,0}x_Q^2 + \ell_{1,1}x_Qy_Q + \ell_{1,0}x_Q + \ell_{0,1}y_Q + \ell_{0,0}$$

$$\ell_{2,0} = -6X_1^2Z_1(5Y_1^4 + 54bY_1^2Z_1^2 - 27b^2Z_1^4);$$
$$\ell_{0,1} = 8X_1Y_1Z_1(5Y_1^4 + 27b^2Z_1^4);$$
$$\ell_{1,1} = 8Y_1Z_1^2(Y_1^4 + 18bY_1^2Z_1^2 - 27b^2Z_1^4);$$
$$\ell_{0,0} = 2X_1(Y_1^6 - 75bY_1^4Z_1^2 + 27b^2Y_1^2Z_1^4 - 81b^3Z_1^6);$$
$$\ell_{1,0} = -4Z_1(5Y_1^6 - 75bZ_1^2Y_1^4 + 135Y_1^2b^2Z_1^4 - 81b^3Z_1^6).$$

- Quadrupling cost $14\mathbf{m} + 16\mathbf{s}$ in $\mathbb{F}_q$ (vs. two doublings: $4\mathbf{m} + 14\mathbf{s}$ in $\mathbb{F}_q$)
- We suffer an extra $10\mathbf{m} + 2\mathbf{s}$ in $\mathbb{F}_q$, but we save a much more costly $\mathbb{F}_{q^k}$ multiplication
- Speed ups for Tate, doesn't work for ate... however...

Fixed argument pairings

- Very common scenario: in the pairing $e(P,Q)$, one of the arguments is fixed as a long term secret key (or constant public param, etc)
- We can exploit this and perform precomputations
- C-Stebila'10 - merging iterations in this scenario is much more powerful (thanks to "anonymous" reviewer of previous work)

	128-bit optimal pairing $k=12$ BN curve $\mathbb{F}_q = 254$ bits		256-bit optimal pairing $k=24$ BLS curve $\mathbb{F}_q = 639$ bits	
precomp method	Miller loop cost	$\approx$ storage required (bits)	Miller loop cost	$\approx$ storage required (bits)
none	6469 $\mathbf{m}_1$	-	19069 $\mathbf{m}_1$	-
Scott '05	5017 $\mathbf{m}_1$	70,000	14794 $\mathbf{m}_1$	340,000
quadrupling	4446 $\mathbf{m}_1$	75,000	12898 $\mathbf{m}_1$	368,000
octupling	4053 $\mathbf{m}_1$	100,000	11673 $\mathbf{m}_1$	510,000

Fixed argument pairings - applications

	# pairings	fixed arguments	# pairings	fixed arguments
Public key encryption	**Encryption**		**Decryption**	
Boyen-Mei-Waters '05	0		1	2^{nd}
ID-based encryption	**Encryption**		**Decryption**	
Boneh-Franklin '03	1	2^{nd}	1	1^{st}
Boneh-Boyen '03	0		1	2^{nd}
Waters '05	0		2	both in 2^{nd}
Attribute-based encr.	**Encryption**		**Decryption**	
GPSW '06	0		$\leq$ #attr.	all in 1^{st}
LOSTW '10	0		$\leq 2 \cdot$ #attr.	all in 2^{nd}
ID-based signatures	**Signing**		**Verification**	
Waters '05	0		2	1 in 2^{nd}
ID-based key exchange	**Initiator**		**Responder**	
Smart-1 '02	2	1 in 1^{st}, 1 in 2^{nd}	2	1 in 1^{st} , 1 in 2^{nd}
Chen-Kudla '03	1	1^{st}	1	2^{nd}
McCullagh-Barreto '05	1	2^{nd}	1	2^{nd}

Table: A few of the protocols that can employ/enjoy our precomputation technique.

Towered extension field arithmetic

- Koblitz-Menezes'05: for $k = 2^i3^j$, build extension field as a sequence of quadratic and cubic subextensions (preferably binomials)
    - Karatsuba-like tricks make arithmetic much faster
    - easier to implement
    - twisted subfields constructed inherently
- e.g. a $k = 12$ tower

$$\mathbb{F}_q \xrightarrow{\beta^2-\alpha} \mathbb{F}_{q^2} \xrightarrow{\gamma^3-\beta} \mathbb{F}_{q^6} \xrightarrow{\delta^2-\gamma} \mathbb{F}_{q^{12}}.$$

- Instead of $\mathbb{F}_{q^{12}}$ multiplications costing 144 $\mathbb{F}_q$ multiplications, they cost $3 \cdot 3 \cdot 6 = 54$ $\mathbb{F}_q$ multiplications
- **Finding a nice tower is not always possible**

Parameterised families of pairing-friendly curves

- All of the best constructions of pairing-friendly curves come from parameterised families
- Implementors now able to gather suitable curves in bulk



Barreto Lynn Scott

- e.g. Barreto-Lynn-Scott (among many other contributions) gave curves with $k = 24$:

$$q(x) = (x-1)^2(x^8 - x^4 + 1)/3 + x;$$
$$n(x) = (x-1)^2(x^8 - x^4 + 1)/3;$$
$$r(x) = x^8 - x^4 + 1;$$

- when $q = q(x)$, $r = r(x)$ are prime, guaranteed a curve $E/\mathbb{F}_q : y^2 = x^3 + b$ with $r \mid n = \#E$.

Example: searching for BLS curves

$$q(x) = (x-1)^2(x^8 - x^4 + 1)/3 + x; \qquad r(x) = x^8 - x^4 + 1.$$

- Kick-start with $x = 2^{64} = 18446744073709551616$ (targeting 256-bit security): $x \equiv 1 \bmod 3$, $x \leftarrow x + 3$
- soon enough $x = 18446744073709563373$
  $q =$
  152081353920740809892727066524584946339781036330218959281772305234001103872205207355200355585054305961029358887567446121016058918174051639618221302567689792185243234190430804646778679690996022 1
- soon after $x = 18446744073709568134$
  $q =$
  1520813539207412024060742043441878459074161652061485145425476810606768714457121717514068260675858946726622675208621738650395266513452695828995492519266950330867144614888025492087559518474496777
- **moral: thousands/millions/billions... of possible curves to choose from...**

Attractive subfamilies of BLS curves for high-security

Theorem (C-Lauter-Naehrig'11)

Instead of $x \equiv 1 \bmod 3$, *take*

x_0 mod 72	$p(x_0)$ mod 72	$n(x_0)$ mod 72	*efficient tower*	*curve* E	*correct twist* E'
7	*19*	*12*	✓	$y^2 = x^3 + 1$	$y^2 = x^3 \pm 1/v$
16	*19*	*3*	✓	$y^2 = x^3 + 4$	$y^2 = x^3 \pm 4v$
31	*43*	*12*	✓	$y^2 = x^3 + 1$	$y^2 = x^3 \pm v$
64	*19*	*27*	✓	$y^2 = x^3 - 2$	$y^2 = x^3 \pm 2/v$

$$\begin{tikzcd}
 & & & \mathbb{F}_{q^8} \arrow[r, "z^3+w"] & \mathbb{F}_{q^{24}}; & T_1, \\
\mathbb{F}_q \arrow[r, "u^2+1"] & \mathbb{F}_{q^2} \arrow[r, "v^2+u+1"] & \mathbb{F}_{q^4} \arrow[ur, "w^2+v"] \arrow[rr, "z^6+v"] \arrow[dr, "w^3+v"'] & & \mathbb{F}_{q^{24}}; & T_2, \\
 & & & \mathbb{F}_{q^{12}} \arrow[r, "z^2+w"'] & \mathbb{F}_{q^{24}}; & T_3.
\end{tikzcd}$$

Particularly friendly members of family trees

- Most recent work took this "subfamilies" idea further
- Thoroughly explored the other 8 families of interest: BW $k = 8$, BLS $k = 12$, KSS $k = 16$, KSS $k = 18$, BLS $k = 27$, KSS $k = 32$, KSS $k = 36$, BLS $k = 48$
- Two (quartic/sextic) twist types: type M and type D (type D previously preferred - untwisting isomorphism)... but **CLN'10 Theorem remedies this!**: there is no preference
- Also give compact generators for many of the favoured subfamilies

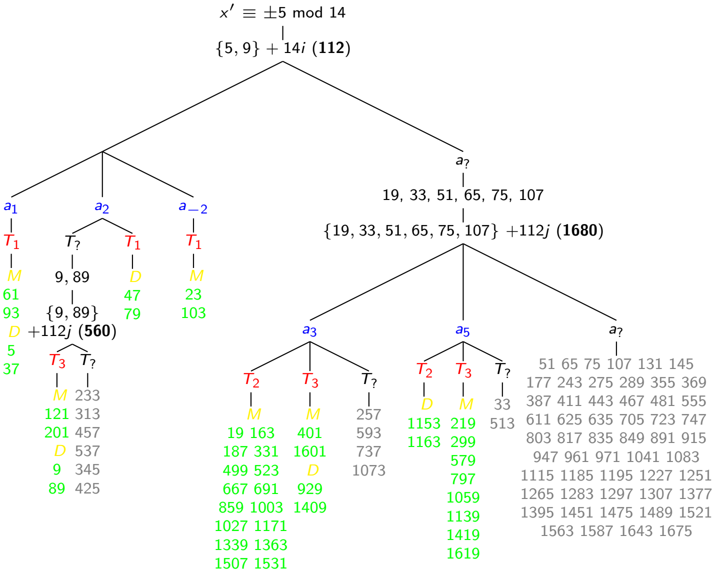


Figure: Example: The $k = 16$ KSS family tree.

Picking fruits from the trees

rating	equiv. class for x' ($x' = x/5$)	tower	a	twist type	$\mathbb{G}_1$ gen. $[h](\cdot,\cdot)$	$\mathbb{G}_2'$ gen. $[h'](\cdot,\cdot)$	%
$\star\star\star\star\star$	$61, 93 \bmod 112$	T_1	1	M	-	$\left(v-1, \sqrt{(v-1)^3 + v(v-1)}\right)$	12.2
	$5, 37 \bmod 112$	T_1	1	D	-	$\left(-v, \sqrt{-v^3-1}\right)$	12.7
	$47, 79 \bmod 112$	T_1	2	D	-	$\left(2/v, \sqrt{\frac{8}{v^3}+\frac{4}{v^2}}\right)$	12.1
	$23, 103 \bmod 112$	T_1	-2	M	$(1, \sqrt{-1})$	-	13.1
$\star\star\star\star$	$\{19, \ldots, 1531\}_{16} \bmod 1680$	T_2	3	M	$(1, 2)$	$\left(3/v, \sqrt{\frac{27}{v^3}+\frac{9}{v^2}}\right)$	7.9
$\star\star\star$	$1153, 1633 \bmod 1680$	T_2	5	D	$\left(2, 2\sqrt{3}\right)$	-	0.9

Table: Our favourite picks from the $k = 16$ KSS tree.

- Implementors can use our trees to tailor-make searches that target specific parameter combinations/preferences
- ... or choose from our extensive list (low hamming-weight examples)
- e.g. $x = -(1 + 2^2 + 2^4 + 2^{16} + 2^{26} + 2^{50})$ corresponds to 47 mod 112, gives $q(x)$ and $r(x)$ as prime, so KSS curve is $y^2 = x^3 + x$, tower is T_1, correct twist is type D.

Summary

Summary

- **Chapter 3:** C-Hisil-Boyd-Gonzalez Nieto-Wong'09 and C-Lange-Naehrig'10: fastest explicit formulas for pairing arithmetic across all elliptic curve models
- **Chapter 4:** C-Boyd-Gonzalez Nieto-Wong'10 (parts I and II): merging Miller iterations to give faster Tate pairing in some scenarios, particularly larger embedding degrees
- **Chapter 5:** C-Stebila'10: applies merging technique to fixed argument scenario for significant improvement in the Miller loop for state-of-the-art pairings
- **Chapter 6:** C-Lauter-Naehrig'11: attractive subfamilies of BLS curves for high-security pairings
- **Chapter 7:** C'12: attractive subfamilies for all other families covering all possible security levels
- **Chapter 8:** C-Lauter'11: new algorithm for arbitrary genus hyperelliptic curve arithmetic, records for genus 2 hyperelliptic curves.

Future work

- **Next 3-4 months:** still fast pairing-based cryptography
    - fixed arguments on hyperelliptic curves
    - fixed argument Weil pairing
    - faster non-pairing operations
    - attacking pairings

- **Next few years:** lattice-based cryptography?
    - efficient fully homomorphic encryption: the holy grail

- **Rest of life:** curves/abelian varieties/arithmetic geometry/computational number theory

Questions?

- Recommended questions include
  - "what's the cost of a pairing compared to [*insert related operation*]"
  - "tell us about your [*insert kind adjective*] work on arbitrary genus arithmetic"
  - "tell us more about the *magic* involved in deriving the faster formulas in Chapters 3 and 4"
  - "why didn't your high-security pairings timings break the software speed record at the 256-bit level?"
  - "show us a pairing in Magma"
  - ...