Exponentiating in Pairing Groups

Joppe W. Bos, Craig Costello, and Michael Naehrig

SAC 2013
Vancouver, Canada

Microsoft® Research

August 16, 2013

The pairing explosion

- The big (bilinear) bang: **[Jou00]**,**[SOK00]**,**[BF01]**. . .

. . .

. . .

. . .

PBC universe still expanding: . . . **[2013/413]**,**[2013/414]** . . .

- Secure bilinear maps would have been welcomed by cryptographers regardless of where they came from

Ben Lynn 2007:

". . . that pairings come from the realm of algebraic geometry (on curves) is a happy coincidence"

- Why so happy?
  - Already received a huge amount of optimization
  - Much more fun than traditional crypto. primitives
  - Discrete log problem on curves already under the microscope

ECC and PBC: a symbiotic relationship

$\longrightarrow\longrightarrow$ Many ECC optimisations quickly transferred to pairings $\longrightarrow\longrightarrow$

e.g.

- avoiding inversions
- projective space
- fast primes (supersingular curves)
- ...

$\longleftarrow\longleftarrow$ Pairings helped ECC too $\longleftarrow\longleftarrow$

e.g.

- 2008/117: Galbraith-Scott - fast exponentiation on pairing groups using $\psi = \phi\pi\hat{\phi}$
- i.e. Frobenius useful over extension fields
- 2008/194: Galbraith-Lin-Scott (GLS) - fast ECC over extension fields using ψ

Non-Weierstrass models for pairings. . . not so much

- A very successful ECC optimization: non-Weierstrass curves

  e.g. Montgomery, Hessian, Jacobi quartics, Jacobi intersections, Edwards, twisted Edwards, . . . (see EFD)

- Not so successful in PBC . . . why?

$$P + Q = R \quad , \quad \mathrm{div}(f) = (P) + (Q) - (R) - (\mathcal{O})$$

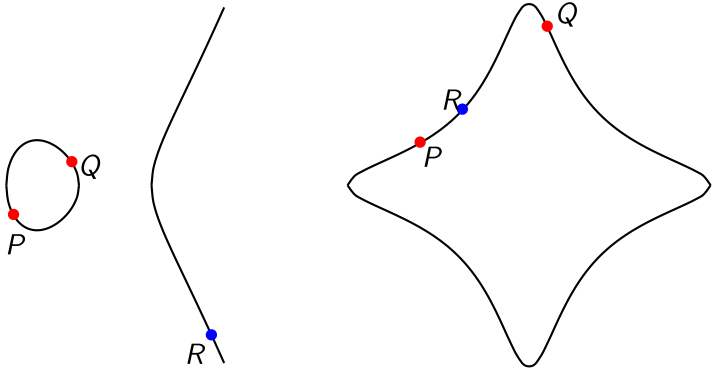
In ECC computations we only need points

get R as fast as possible

In pairing computations we need points *and* functions

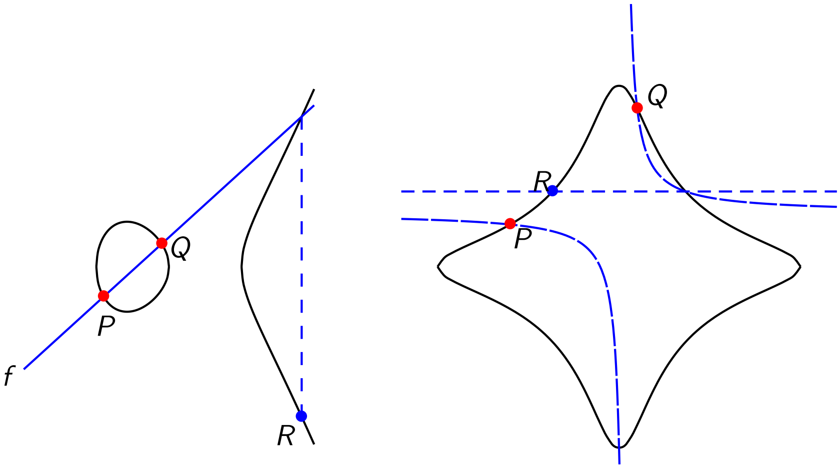
get R and f as fast as possible

Non-Weierstrass faster for ECC



Getting R from P and Q: much faster on Edwards (and others)

Weierstrass faster for pairings



Getting R, f from P and Q: Weierstrass preferable

This work: focus only on the scalar multiplications ...

Alternative models not faster for pairing, **but** *can they be used to enhance scalar multiplications in pairing groups???*

- maybe even bigger speedups for pairing exponentiations
- high dimensional GLV/GLS (# doublings $<$ # additions)
- additions is where Weierstrass sucks the most
- e.g. $y^2 = x^3 + b$ - Weierstrass add. $\approx 17\mathbf{m}$, Edwards $\approx 9\mathbf{m}$!!!
- curve models in pairings very minor improvement at best, but in scalar mulplications big savings possible!

Pairing-based protocols in practice

- pairing computation involves three groups $e\colon \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$
- often many more standalone operations in any or all of $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ than pairing(s) ... can be orders of magnitude more!

Utilizing non-Weierstrass models

- $\mathcal{J}$ = Jacobi quartic $\quad \mathcal{H}$ = Hessian $\quad \mathcal{E}$ = twisted Edwards
- We always have $j = 0$ in this work (e.g. $\mathcal{H}$ has $d = 0$)

Pairing on *Scalar mults on* *iff*

$$\begin{tikzcd}
 & \mathcal{J}\colon y^2 = dx^4 + 2ax^2 + 1 \arrow[dl, bend left, "\tau^{-1}"] & 2 \mid \#\mathcal{W} \\
\mathcal{W}\colon y^2 = x^3 + b \arrow[ur, bend left, "\tau"] \arrow[r, bend left, "\tau"] \arrow[dr, bend left, "\tau"] & \mathcal{H}\colon x^3 + y^3 + c = 0 \arrow[l, bend left, "\tau^{-1}"] & 3 \mid \#\mathcal{W} \\
 & \mathcal{E}\colon ax^2 + y^2 = 1 + dx^2y^2 \arrow[ul, bend left, "\tau^{-1}"] & 4 \mid \#\mathcal{W}^*
\end{tikzcd}$$

- Note $*$: field K has $\#K \equiv 1 \bmod 4$, then $4 \mid E$ is enough, otherwise need point of order 4 for $\mathcal{E}$ (cheers anon. reviewer)

The power of the sextic twist for $\mathbb{G}_2$

- Elements in $\mathbb{G}_2$ are points over the extension field $\subset E(\mathbb{F}_{p^k})$
  - k times larger to store
  - m times more costly to work over $\mathbb{F}_{p^k}$, where $k \ll m \leq k^2$!!!
- Can use group isomorphic to $\mathbb{G}_2$, which is on a different curve:

$$\mathbb{G}_2' \subseteq E'(\mathbb{F}_{p^{k/d}})$$

- E' is called the **twisted curve**
  - elements compressed by factor d
  - m times faster to work with, where $d \ll m \leq d^2$

Sextic twists: $d = 6$ is biggest possible for elliptic curves

- only possible if $6 \mid k$ and $j = 0$ (i.e. $y^2 = x^3 + b$)
- luckily all the best families with $6 \mid k$ have $y^2 = x^3 + b$
- $E'/\mathbb{F}_{p^{k/d}}$: $y^2 = x^3 + b'$, and $\Psi \colon E' \to E$ to map $\mathbb{G}_2' \leftrightarrow \mathbb{G}_2$

Mapping back and forth to $\mathcal{W}$

Galbraith-Scott'08

- $\mathbb{G}_1 \subseteq E(\mathbb{F}_p) : y^2 = x^3 + b$
  - $\phi : (x, y) \mapsto (\zeta x, y)$, $\zeta^3 = 1 \in \mathbb{F}_p$
  - gives 2-dimensional (GLV) decomposition on $\mathbb{G}_1$
- $\mathbb{G}_2' \subseteq E'(\mathbb{F}_{p^e}) : y^2 = x^3 + b'$
  - $\psi = \Psi \cdot \pi_p \cdot \Psi^{-1}$
  - gives $\varphi(k)$-dimensional (GLS) decomposition on $\mathbb{G}_2$

- $[k]P$ starts by computing $\phi(P)$ or $\psi^i(P)$ for $1 \leq i \leq d-1$
- ideally we'd define (elements of) $\mathbb{G}_1$ or $\mathbb{G}_2$ on fastest model
- requires endomorphisms to transfer favorably to other model, but only GLV morphism ϕ on $\mathcal{H} : x^3 + y^3 + c = 0$ does ☹

The general strategy

We apply ϕ or ψ (repeatedly) on $\mathcal{W}$, map across to $\mathcal{J}$, $\mathcal{H}$ or $\mathcal{E}$ for the rest of the routine, and come back to $\mathcal{W}$ at the end

Our goal

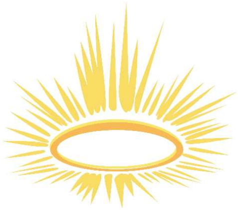
sec. level	family-k	pairing e	exp. in $\mathbb{G}_1$	exp. in $\mathbb{G}_2$	exp. in $\mathbb{G}_T$
128-bit	BN-12	?	**??**	**??**	?
192-bit	BLS-12	?	**??**	**??**	?
	KSS-18	?	**??**	**??**	?
256-bit	BLS-24	?	**??**	**??**	?

- to fill in the above table using all of the state of the art techniques for pairings/exponentiations
- give protocol designers a good idea of the ratios

$$e : \mathbb{G}_1 : \mathbb{G}_2 : \mathbb{G}_T$$

- not speed records (no assembly), but ratios should remain $\approx$ same
- find optimal curve models in all **??** cases

$k = 12$ Barreto-Naehrig (BN) curves



$$p(x) = 36x^4 + 36x^3 + 24x^2 + 18x + 1$$
$$n(x) = 36x^4 + 36x^3 + 18x^2 + 18x + 1$$

- BN curves are so good: for our purposes, they are too good
- they were meant to be prime - can't even force small cofactor

Prop 1. *Let $E/\mathbb{F}_p$ be a BN curve with sextic twist $E'/\mathbb{F}_{p^2}$. The groups $E(\mathbb{F}_p)$ and $E'(\mathbb{F}_{p^2})$ do not contain points of order 2, 3 or 4.*

...but for the other popular families ...

Prop 2. *For* $p \equiv 3 \bmod 4$, *let* $E/\mathbb{F}_p$ *be a* $k = 12$ *BLS curve with sextic twist* $E'/\mathbb{F}_{p^2}$. *The group* $E(\mathbb{F}_p)$ *contains a point of order* 3 *and can contain a point of order 2, but not 4, while the group* $E'(\mathbb{F}_{p^2})$ *does not contain a point of order 2, 3 or 4.*

Prop 3. *Let* $E/\mathbb{F}_p$ *be a* $k = 18$ *KSS curve with sextic twist* $E'/\mathbb{F}_{p^3}$. *The group* $E(\mathbb{F}_p)$ *does not contain a point of order 2, 3 or 4, while the group* $E'(\mathbb{F}_{p^3})$ *contains a point of order 3 but does not contain a point of order 2 or 4.*

Prop 4. *For* $p \equiv 3 \bmod 4$, *let* $E/\mathbb{F}_p$ *be a* $k = 24$ *BLS curve and sextic twist* $E'/\mathbb{F}_{p^4}$. *The group* $E(\mathbb{F}_p)$ *can contain points of order 2 or 3 (although not simultaneously), but not 4, while the group* $E'(\mathbb{F}_{p^4})$ *can contain a point of order 2, but does not contain a point of order 3 or 4.*

Available models...

family-k	$\mathbb{G}_1$ algorithm	$\mathbb{G}_1$ models avail.	$\mathbb{G}_2$ algorithm	$\mathbb{G}_2$ models avail.
BN-12	2-GLV	$\mathcal{W}$	4-GLS	$\mathcal{W}$
BLS-12	2-GLV	$\mathcal{H},\mathcal{J},\mathcal{W}$	4-GLS	$\mathcal{W}$
KSS-18	2-GLV	$\mathcal{W}$	6-GLS	$\mathcal{H},\mathcal{W}$
BLS-24	2-GLV	$\mathcal{H},\mathcal{J},\mathcal{W}$	8-GLS	$\mathcal{E},\mathcal{J},\mathcal{W}$

model	`DBL` cost	`ADD` cost	`MIX` cost	`AFF` cost
Weierstrass - $\mathcal{W}$	**7**	**16**	**11**	**6**
Jacobi-quartic - $\mathcal{J}$	**9**	**13**	**12**	**11**
Hessian - $\mathcal{H}$	**7**	**12**	**10**	**8**
twisted Edwards - $\mathcal{E}$	**9**	**10**	**9**	**8**

- operation counts don't/can't assume small constants like ECC

Best models. . .

family-k	$\mathbb{G}_1$ algorithm	$\mathbb{G}_1$ models avail.	$\mathbb{G}_2$ algorithm	$\mathbb{G}_2$ models avail.
BN-12	2-GLV	$\mathcal{W}$	4-GLS	$\mathcal{W}$
BLS-12	2-GLV	Hessian (1.23x)	4-GLS	$\mathcal{W}$
KSS-18	2-GLV	$\mathcal{W}$	6-GLS	Hessian (1.11x)
BLS-24	2-GLV	Hessian (1.19x)	8-GLS	twisted Edwards (1.16x)

model/ coords	`DBL` cost	`ADD` cost	`MIX` cost	`AFF` cost
$\mathcal{W}$ / Jac.	**7**	**16**	**11**	**6**
$\mathcal{J}$ / ext.	**9**	**13**	**12**	**11**
$\mathcal{H}$ / proj.	**7**	**12**	**10**	**8**
$\mathcal{E}$ / ext.	**9**	**10**	**9**	**8**

- for BLS $k=12$ and BLS $k=24$, define $\mathbb{G}_1 \subset \mathcal{H}/\mathbb{F}_p$ (modify pairing to include initial conversion to $\mathcal{W}$)
- for KSS $k=18$ and BLS $k=24$, $\mathbb{G}_2 \subset \mathcal{W}/\mathbb{F}_p$, but τ to $\mathcal{H}, \mathcal{E}$ after ψ's are computed, and τ^{-1} to come back to $\mathcal{W}$ at end

Results

Benchmark results (in millions (M) of clock cycles Intel Core i7-3520M).

sec. level	family-k	pairing e	exp. in $\mathbb{G}_1$	exp. in $\mathbb{G}_2$	exp. in $\mathbb{G}_T$
128-bit	BN-12	7.0	0.9	1.8	3.1
192-bit	BLS-12	47.2	4.4	10.9	17.5
	KSS-18	63.3	3.5	9.8	15.7
256-bit	BLS-24	115.0	5.2	27.6	47.1

- state-of-the-art algorithms (optimal ate, lazy reduction, cyclotomic squarings, etc.)
- not rivalling speed records, but $e\colon \mathbb{G}_1\colon \mathbb{G}_2\colon \mathbb{G}_T$ ratios should stay similar
- should give protocol designers a good idea of ratios
- what's best for 192-bit security (match protocol to family)
- for BN ratios at hardcore level, see: http://sandia.cs.cinvestav.mx/index.php?n=Site.CPABE

(Zavattoni, Dominguez Perez, Mitsunari, Sanchez, Teruya, Rodriguez-Henriquez)