Exponentiating in Pairing Groups

Michael Naehrig

Microsoft Research
mnaehrig@microsoft.com

joint work with
Joppe W. Bos and Craig Costello

SIAM AG13, Fort Collins
MS5 - Cryptography and Number Theory
1 August 2013

Pairings

$$e : G_1 \times G_2 \to G_T$$

- G_1 and G_2 are groups (of points on an elliptic curve),
- G_T is a (multiplicative) group (of finite field elements),
- all groups have prime order r,
- e is bilinear, non-degenerate, efficiently computable

Pairing groups

$$e : G_1 \times G_2 \to G_T$$

- $G_1 = E(\mathbb{F}_p)[r]$, $G_2 \subseteq E(\mathbb{F}_{p^k})[r]$,
- $E/\mathbb{F}_p$: elliptic curve, e.g. in short Weierstrass form

$$E : y^2 = x^3 + ax + b,$$

- r prime, $r \mid \#E(\mathbb{F}_p)$, $\mathrm{char}(\mathbb{F}_p) > 3$,
- with small (even) embedding degree $k > 1$,

$$r \mid p^k - 1, \quad r \nmid p^i - 1 \text{ for } i < k,$$

- $G_T = \mu_r \subseteq \mathbb{F}_{p^k}^*$ group of r-th roots of unity,

Optimal ate pairings

Typical setting at higher security levels:

$$e : G_2' \times G_1 \to G_T, \quad (Q', P) \mapsto g_{Q'}(P)^{\frac{p^k-1}{r}}$$

- $G_1 = E(\mathbb{F}_p)[r]$, $G_2' = E'(\mathbb{F}_{p^e})[r]$, $G_T = \mu_r \subseteq \mathbb{F}_{p^k}^*$,
- $E'/\mathbb{F}_{p^e}$: twist of E of degree $d \mid k$, $e = k/d$, $r \mid \#E'(\mathbb{F}_{p^e})$,
- $g_{Q'}$: function depending on Q' with coefficients in $\mathbb{F}_{p^k}^*$.

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

Many ECC optimisations quickly transferred to pairings, e.g.

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

Pairings helped ECC too, e.g.

- Galbraith-Scott 2008: fast exponentiation on pairing groups using efficiently computable endomorphisms
- i.e. Frobenius useful over extension fields
- Galbraith-Lin-Scott (GLS) 2008: fast ECC over extension fields using eff. comp. endomorph.

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?

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

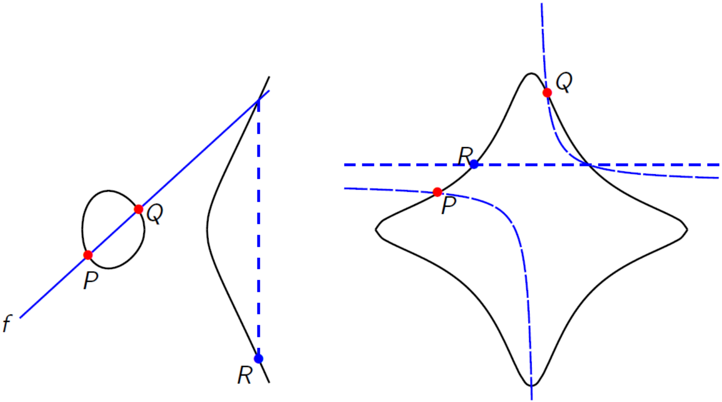
In ECC computations we only need points

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. . . not for PBC



Getting *R* from *P* and *Q*: much faster on Edwards (and others)
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)
- for additions, Weierstrass coordinates suck 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!

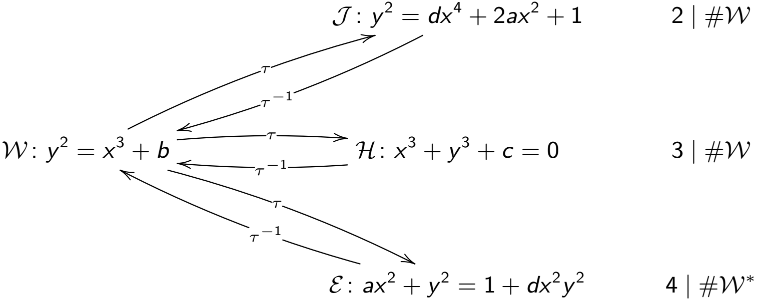
Pairing-based protocols in practice

- 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*



- 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 $y^2 = x^3 + b$
- $E'/\mathbb{F}_{p^{k/d}} \colon y^2 = x^3 + b'$, and $\Psi \colon E' \to E$ to map $\mathbb{G}_2' \leftrightarrow \mathbb{G}_2$

GLV/GLS

Galbraith-Scott 2008

- $\mathbb{G}_1 \subseteq E(\mathbb{F}_p) : y^2 = x^3 + b$
  - $\phi : (x, y) \mapsto (\zeta x, y)$, $\zeta^3 = 1 \in \mathbb{F}_p$
  - $\phi(P) = [\lambda_\phi]P$ for $\lambda_\phi^2 + \lambda_\phi + 1 \equiv 0 \bmod r$
  - gives 2-dimensional (GLV) decomposition on $\mathbb{G}_1$
- $\mathbb{G}_2' \subseteq E'(\mathbb{F}_{p^e}) : y^2 = x^3 + b'$
  - $\psi = \Psi \cdot \pi_p \cdot \Psi^{-1}$
  - $\psi(P) = [\lambda_\psi]P$ for $\Phi_k(\lambda_\psi) \equiv 0 \bmod r$
  - gives $\varphi(k)$-dimensional (GLS) decomposition on $\mathbb{G}_2'$

GLV/GLS

- $[s]P$ starts by computing $\phi(P)$ or $\psi^i(P)$ for $1 \leq i \leq \varphi(k) - 1$
- decompose $[s]P = \sum_{i=0}^{\varphi(k)-1}[s_i]P_i$ by finding a vector close to $(s, 0)$ or $(s, 0, \ldots, 0)$ in the GLV/GLS lattices

$$B_\phi = \begin{pmatrix} r & 0 \\ -\lambda_\phi & 1 \end{pmatrix}; \qquad B_\psi = \begin{pmatrix} r & 0 & \ldots & 0 \\ -\lambda_\psi & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -\lambda_\psi^{\varphi(k)-1} & 0 & \ldots & 1 \end{pmatrix}.$$

- all s_i are much shorter than s
- compute $[s]P = \sum_{i=0}^{\varphi(k)-1}[s_i]P_i$ by multi-exponentiation

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

- 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	?	**??**	**??**	?

- fill in the above table using state-of-the-art techniques for exponentiations and pairings
- give protocol designers a good idea of the ratios of exponentiation costs in

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

- no speed records (no assembly)
- find optimal curve models in all **??** cases

Points of small order

Prop 1. *BN ($k = 12$):* *$E(\mathbb{F}_p)$ and $E'(\mathbb{F}_{p^2})$ do not contain points of order 2, 3 or 4.*

Prop 2. *BLS ($k = 12$):* *If $p \equiv 3 \bmod 4$, $E(\mathbb{F}_p)$ contains a point of order 3 and can contain a point of order 2, but not 4.*
$E'(\mathbb{F}_{p^2})$ does not contain a point of order 2, 3 or 4.

Prop 3. *KSS ($k = 18$):* *$E(\mathbb{F}_p)$ does not contain a point of order 2, 3 or 4.*
$E'(\mathbb{F}_{p^3})$ contains a point of order 3 but none of order 2 or 4.

Prop 4. *BLS ($k = 24$):* *If $p \equiv 3 \bmod 4$, $E(\mathbb{F}_p)$ can contain points of order 2 or 3 (although not simultaneously), but not 4.*
$E'(\mathbb{F}_{p^4})$ can contain a point of order 2, but none 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/ coords	DBL cost	ADD cost	MIX cost	AFF cost
$\mathcal{W}$ / Jac.	$\mathbf{7}_{2,5,0,14}$	$\mathbf{16}_{11,5,0,13}$	$\mathbf{11}_{7,4,0,14}$	$\mathbf{6}_{4,2,0,12}$
$\mathcal{J}$ / ext.	$\mathbf{9}_{1,7,1,12}$	$\mathbf{13}_{7,3,3,19}$	$\mathbf{12}_{6,3,3,18}$	$\mathbf{11}_{5,3,3,18}$
$\mathcal{H}$ / proj.	$\mathbf{7}_{6,1,0,11}$	$\mathbf{12}_{12,0,0,3}$	$\mathbf{10}_{10,0,0,3}$	$\mathbf{8}_{8,0,0,3}$
$\mathcal{E}$ / ext.	$\mathbf{9}_{4,4,1,7}$	$\mathbf{10}_{9,0,1,7}$	$\mathbf{9}_{8,1,0,7}$	$\mathbf{8}_{7,0,1,7}$

- 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.	$\mathbf{7}_{2,5,0,14}$	$\mathbf{16}_{11,5,0,13}$	$\mathbf{11}_{7,4,0,14}$	$\mathbf{6}_{4,2,0,12}$
$\mathcal{J}$ / ext.	$\mathbf{9}_{1,7,1,12}$	$\mathbf{13}_{7,3,3,19}$	$\mathbf{12}_{6,3,3,18}$	$\mathbf{11}_{5,3,3,18}$
$\mathcal{H}$ / proj.	$\mathbf{7}_{6,1,0,11}$	$\mathbf{12}_{12,0,0,3}$	$\mathbf{10}_{10,0,0,3}$	$\mathbf{8}_{8,0,0,3}$
$\mathcal{E}$ / ext.	$\mathbf{9}_{4,4,1,7}$	$\mathbf{10}_{9,0,1,7}$	$\mathbf{9}_{8,1,0,7}$	$\mathbf{8}_{7,0,1,7}$

- 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 rivaling speed records, but hope that $\mathbb{G}_1 : \mathbb{G}_2 : \mathbb{G}_T : e$ ratios stay similar
- should give protocol designers a good idea of ratios
- what's best for 192-bit security (match protocol to family)