Part I: Efficient arithmetic on hyperelliptic curves over finite fields
–
Part II: Pairings

Tanja Lange

Department of Mathematics

Technical University of Denmark

t.lange@mat.dtu.dk

31.08.2006

Overview

- Arithmetic on hyperelliptic curves
  - Fast arithmetic for curves over odd characteristic fields.
  - Fast arithmetic for curves over even characteristic fields.
- Pairings
  - Protocols
  - Weil and Tate pairing on elliptic curves
  - Efficient computation of pairings
  - Supersingular and ordinary elliptic curves
  - Embedding degree
  - MNT curves
  - Outlook

Hyperelliptic curves – reminder

$\mathbb{F}_q$ is some finite field with q element.

Affine equation of hyperelliptic curve of genus g (with $\mathbb{F}_q$-rational Weierstraß-point at infinity)

$$C : y^2 + h(x)y = f(x)$$

$h(x), f(x) \in \mathbb{F}_q[x]$, f monic, $\deg f = 2g + 1, \deg h \leq g$
non singular, i. e. in no point $(a, b) \in C/\overline{\mathbb{F}_q}$ over the algebraic closure $\overline{\mathbb{F}_q}$ we have both partial derivatives vanishing, i.e.

$2b + h(a) = 0$ and $h'(a)b - f'(a) = 0$.

Otherwise (a, b) is called a singular point.

Mumford Representation

Group elements (divisor classes) $\bar{D}$ represented by two polynomials

$$\bar{D} = [u(x), v(x)]; u, v \in \mathbb{F}_q[x],$$

satisfying u monic, $\deg v < \deg u \leq g, \quad u | v^2 + vh - f.$

$\Rightarrow$ compact representation over ground field.

Arithmetic in general case can be stated in these representatives and thus uses polynomial arithmetic.

For relations between this representation and divisors given as sums of points see my talk yesterday.

Arithmetic on hyperelliptic curves

Composition & Reduction (Cantor/Koblitz)

`IN:` $\bar{D}_1 = [u_1, v_1], \bar{D}_2 = [u_2, v_2], C : y^2 + h(x)y = f(x)$
`OUT:` $\bar{D} = [u, v]$ reduced with $\bar{D} = \bar{D}_1 \oplus \bar{D}_2$

1. compute $d_1 = \gcd(u_1, u_2) = e_1 u_1 + e_2 u_2$
2. compute $d = \gcd(d_1, v_1 + v_2 + h) = c_1 d_1 + c_2(v_1 + v_2 + h)$
3. let $s_1 = c_1 e_1, s_2 = c_1 e_2, s_3 = c_2$
4. $u = \frac{u_1 u_2}{d^2} \quad v = \frac{s_1 u_1 v_2 + s_2 u_2 v_1 + s_3(v_1 v_2 + f)}{d} \mod u$

This result $[u, v]$ corresponds to a semireduced divisor in the resulting divisor class.

Arithmetic on hyperelliptic curves

Composition & Reduction (Cantor/Koblitz)

`IN:` $\bar{D}_1 = [u_1, v_1], \bar{D}_2 = [u_2, v_2], C : y^2 + h(x)y = f(x)$
`OUT:` $\bar{D} = [u, v]$ reduced with $\bar{D} = \bar{D}_1 \oplus \bar{D}_2$

1. compute $d_1 = \gcd(u_1, u_2) = e_1 u_1 + e_2 u_2$
2. compute $d = \gcd(d_1, v_1 + v_2 + h) = c_1 d_1 + c_2(v_1 + v_2 + h)$
3. let $s_1 = c_1 e_1, s_2 = c_1 e_2, s_3 = c_2$
4. $u = \frac{u_1 u_2}{d^2} \quad v = \frac{s_1 u_1 v_2 + s_2 u_2 v_1 + s_3(v_1 v_2 + f)}{d} \bmod u$
5. let $u' = \frac{f - vh - v^2}{u} \quad b' = (-h - v) \bmod u'$
6. if $\deg u' > g$ put $u := u', v := v'$ goto step 5
7. make u monic

   This result $[u, v]$ corresponds to a reduced divisor.

Hasse-Weil bounds

- Group order is $\sim q^g$ for curve of genus g over $\mathbb{F}_q$.
- Exactly the intervals are
  - Points on curve over $\mathbb{F}_q$

$$|\#C(\mathbb{F}_q) - q| \leq 2g\sqrt{q}.$$

  - Divisor classes over $\mathbb{F}_q$

$$(q^{1/2} - 1)^{2g} \leq |\mathrm{Pic}^0_C(\mathbb{F}_q)| \leq (q^{1/2} + 1)^{2g}.$$

  Same group size for smaller fields with increasing genus, e.g.
- $g = 1, q \sim 2^{256}$ like in Dan Bernstein's talk,
- corresponds to $g = 2, q \sim 2^{128}$, or to $g = 3, q \sim 2^{85}$.

Security of hyperelliptic curves I

- Divisor class group has generic DL algorithms (baby-step giant-step, Pollard rho, Pollard kangaroo) with running time
$$O(\sqrt{\ell}),$$
were ℓ is largest prime factor of group order.
- For elliptic curves and hyperelliptic curves of small genus we know no faster algorithms.
- Note, bad choices of curves are possible, e.g. anomalous, supersingular, small embedding degrees, composite degrees allowing Weil descent attacks.

Security of hyperelliptic curves II

- Index calculus attacks make curves of large genus weaker. The running time of the double large prime variation for fixed genus is

$$O(q^{2-\frac{2}{g}}).$$

  Compare this to $O(\sqrt{\ell}) \sim O(q^{g/2})$!
  E.g. for $g = 3$ this is $O(q^{4/3})$ which is smaller than $O(q^{3/2})$.

- Previous talks presented fast arithmetic for elliptic curves; we do the same for genus 2 and give references for $g = 3$.

Explicit Arithmetic $g = 2$ and $g = 3$

2000	2	Harley (odd char.)	
2001	2	L. (arbitrary char.)	$\Rightarrow$ 2 inversions
2001	2	Matsuo, Chao, Tsujii (faster)	
2002	2	Miyamoto, Doi, Matsuo, Chao, Tsujii	
2002	2	Takahashi	
2002	2	L. (arbitrary char.)	$\Rightarrow$ 1 inv.
2002	2	Sugizaki, Matsuo, Chao, Tsujii (even)	
2004	2	L., Stevens (even char., all cases)	
2002	3	Kuroki, Gonda, Matsuo, Chao, Tsujii	
2002	3	Pelzl	
2002	3	Guyot & Patankar	$\Rightarrow$ 1 inv.
2004	3	Fan, Wang (projective)	

General doubling, $g = 2$

Doubling, $\deg u = 2$			
Input	$[u, v], u = x^2 + u_1 x + u_0, v = v_1 x + v_0$		
Output	$[u', v'] = 2[u, v]$		
Step	Expression	odd	even
1	compute $\tilde{v} \equiv (h + 2v) \bmod u = \tilde{v}_1 x + \tilde{v}_0$: $\tilde{v}_1 = h_1 + 2v_1 - h_2 u_1, \tilde{v}_0 = h_0 + 2v_0 - h_2 u_0$;		
2	compute resultant $r = \mathrm{res}(\tilde{v}, u)$: $w_0 = v_1^2, w_1 = u_1^2, w_2 = \tilde{v}_1^2, w_3 = u_1 \tilde{v}_1, r = u_0 w_2 + \tilde{v}_0(\tilde{v}_0 - w_3)$;	2S, 3M $(w_2 = 4w_0)$	2S, 3M (see below)
3	compute almost inverse $inv' = invr$: $inv'_1 = -\tilde{v}_1, inv'_0 = \tilde{v}_0 - w_3$;		
4	compute $k' = (f - hv - v^2)/u \bmod u = k'_1 x + k'_0$: $w_3 = f_3 + w_1, w_4 = 2u_0, k'_1 = 2(w_1 - f_4 u_1) + w_3 - w_4 - h_2 v_1$; $k'_0 = u_1(2w_4 - w_3 + f_4 u_1 + h_2 v_1) + f_2 - w_0 - 2f_4 u_0 - h_1 v_1 - h_2 v_0$;	1M	2M (see below)
5	compute $s' = k' inv' \bmod u$: $w_0 = k'_0 inv'_0, w_1 = k'_1 inv'_1, s'_1 = (inv'_0 + inv'_1)(k'_0 + k'_1) - w_0 - w_1(1 + u_1), s'_0 = w_0 - u_0 w_1$;	5M	5M
6	compute $s'' = x + s_0/s_1$ and s_1: $w_1 = 1/(r s'_1)(= 1/r^2 s_1), w_2 = r w_1 (= 1/s'_1), w_3 = s'^2_1 w_1 (= s_1)$; $w_4 = r w_2 (= 1/s_1), w_5 = w_4^2, s''_0 = s'_0 w_2$;	I, 2S, 5M	I, 2S, 5M
7	compute $l' = s'' u = x^3 + l'_2 x^2 + l'_1 x + l'_0$: $l'_2 = u_1 + s''_0, l'_1 = u_1 s''_0 + u_0, l'_0 = u_0 s''_0$;	2M	2M
8	compute $u' = s^2 + (h + 2v)s/u + (v^2 + hv - f)/u^2$: $u'_0 = s''^2_0 + w_4(h_2(s''_0 - u_1) + 2v_1 + h_1) + w_5(2u_1 - f_4), u'_1 = 2s''_0 + h_2 w_4 - w_5$;	S, 2M	S, M
9	compute $v' \equiv -h - (l + v) \bmod u' = v'_1 x + v'_0$: $w_1 = l'_2 - u'_1, w_2 = u'_1 w_1 + u'_0 - l'_1, v'_1 = w_2 w_3 - v_1 - h_1 + h_2 u'_1$; $w_2 = u'_0 w_1 - l'_0, v'_0 = w_2 w_3 - v_0 - h_0 + h_2 u'_0$;	4M	4M
total		each I, 22M, 5S	

Elliptic curves vs. $g = 2$

- Doubling on elliptic curve needs $1I_q, 2M_q, 2S_q$.
- Doubling on genus 2 curve needs $1I_{q'}, 22M_{q'}, 5S_{q'}$.
- $\log q \sim 2 \log q'$ – naive Karatsuba field arithmetic $M_q = 3M_{q'}$ and $M = S \Rightarrow$ comparison depends on $1I_q <=> 1I_{q'} + 15M_{q'}$ (and on whether the naive assumptions hold).
- Theoretical analysis depends on inversion/multiplication (I/M)-ratio, S/M-ratio and ratio $M_q/M_{q'}$.
- Hard to take into account load instructions etc.
- How about precomputations?
- Comparison by implementation in general case (R. Avanzi, CHES'04) shows ECC faster.
- Remainder of talk: tunings to become faster than ECC.

Field of odd characteristic

Specialization $g = 2$, odd q

- If q is odd, transform $y \mapsto \tilde{y} - h(x)/2$

$$(\tilde{y} - h(x)/2)^2 + h(x)(\tilde{y} - h(x)/2) =$$
$$\tilde{y}^2 + (-h(x) + h(x))\tilde{y} + (1/4 - 1/2)h(x)^2 = f(x)$$

  and sort, to obtain curve equation of form

$$\tilde{y}^2 = \tilde{f}(x).$$

- $P = (a, b)$, then $-P = (a, -b)$
- ADD takes $1I, 22M, 3S$, DBL needs $1I, 22M, 5S$.
- Dan Bernstein says: inversions are expensive!!!
- So, avoid inversions.

How does it work?

- Get an idea:
  - Start with normal inputs, at some point one needs to invert.
  - Put the element to be inverted in Z.
  - Adjust all following steps to have denominator Z.
- Start with input having $Z \neq 1$.
  - For ADD this means a lot of adjustments because they have different Z-values.
  - DBL suffers less.
  - ADD might be mixed – one input has $Z = 1$, e.g. the base point in a DL system. Mixed ADD faster.
- Try different weights.

Different Coordinates $g = 2$, odd q

affine $\mathcal{A}$:

$$D \sim [x^2 + u_1 x + u_0, v_1 x + v_0] \sim [u_1, u_0, v_1, v_0]$$

projective $\mathcal{P}$: Miyamoto, Doi, Matsuo, Chao, and Tsujii; L.; all 2002

$$D \sim [U_1, U_0, V_1, V_0, Z] \sim [U_1/Z, U_0/Z, V_1/Z, V_0/Z]$$

new $\mathcal{N}$: L. 2002, contains also mixed coordinates

$$D \sim [U_1, U_1, V_1, V_0, Z_1, Z_2] \sim [U_1/Z_1^2, U_0/Z_1^2, V_1/(Z_1^3 Z_2), V_0/(Z_1^3 Z_2)]$$

$\mathcal{P}$ and $\mathcal{N}$: **no inversions**

Comparison - odd characteristic

Doubling		**Addition**	
operation	costs	operation	costs
$2\mathcal{N} = \mathcal{P}$	7S, 38M	$\mathcal{N} + \mathcal{N} = \mathcal{P}$	7S, 51M
$2\mathcal{P} = \mathcal{P}$	6S, 38M	$\mathcal{N} + \mathcal{P} = \mathcal{P}$	4S, 51M
$2\mathcal{N} = \mathcal{N}$	7S, 34M	$\mathcal{N} + \mathcal{N} = \mathcal{N}$	7S, 47M
$2\mathcal{P} = \mathcal{N}$	6S, 34M	$\mathcal{N} + \mathcal{P} = \mathcal{N}$	4S, 48M
		$\mathcal{P} + \mathcal{P} = \mathcal{P}$	4S, 47M
		$\mathcal{P} + \mathcal{P} = \mathcal{N}$	4S, 44M
		$\mathcal{A} + \mathcal{N} = \mathcal{P}$	5S, 40M
		$\mathcal{A} + \mathcal{P} = \mathcal{P}$	3S, 40M
		$\mathcal{A} + \mathcal{N} = \mathcal{N}$	5S, 36M
		$\mathcal{A} + \mathcal{P} = \mathcal{N}$	3S, 37M
$2\mathcal{A} = \mathcal{A}$	1I, 5S, 22M	$\mathcal{A} + \mathcal{A} = \mathcal{A}$	1I, 3S, 22M

Arithmetic a la Pierrick Gaudry

ePrint Report 2005/314
Fast genus 2 arithmetic based on Theta functions

- Needs full 2-torsion group, i. e. cofactor 16.
- Shows that approach valid over general fields.
- ADD + DBL = 25M – no inversion!!!
  (cf. affine ADD: 22M + 3S + 1I, DBL: 22M + 5S + 1I)
- is faster than elliptic curves in Montgomery form.
- Application of formulas is straightforward, only background is more involved.

Field of characteristic 2

Genus 2 curves over $\mathbb{F}_{2^n}$

$$C : y^2 + (h_2x^2 + h_1x + h_0)y = f(x), \deg(f) = 5.$$

Formulas stated earlier hold for general equation. Get higher efficiency by specializing to degree of h.

- $h = 0$ belongs to singular curves, so not hyperelliptic.
- $\deg h = 0$: the curve is supersingular (cf. Pairings in my next talk, choices of curves in last talk by Dan Bernstein).
- $\deg(h) = 1$: group has co-factor 2, no weaknesses up to our present knowledge.
- If $\deg h = 2$ and h has a root then one can achieve $h(x) = x^2 + h_1x$, otherwise $h(x) = x^2 + h_1x + h_0$. In the former case the group order is divisible by 4, in the latter by 2.

Isomorphic transformations

Consider e. g. $\deg h = 1$:
Changes of the form $y \to y' + ax'^2 + bx', x \to cx + d$ allow $f_4 = f_1 = 0$ and some more.

Usual situation:

$$C : y^2 + h_1 xy = x^5 + f_3 x^3 + f_2 x^2 + f_0.$$

If $\mathbb{F}_q = \mathbb{F}_{2^n}$ with odd n one can additionally achieve $h_1 = 1, f_2 \in \mathbb{F}_2$.

In scalar multiplications DBL more important than ADD (use precomputation) $\Rightarrow$ speed up DBL.

Doubling, $g = 2$, $\deg h = 1$

Doubling $\deg h = 1$, $\deg u = 2$	
Input	$[u, v], u = x^2 + u_1 x + u_0, v = v_1 x + v_0; h_1^2, h_1^{-1}$
Output	$[u', v'] = 2[u, v]$

Step	Expression	$h_1 = 1$	h_1^{-1} small	h_1 arbitrary
1	compute rs_1: $z_0 = u_0^2,\ k_1' = u_1^2 + f_3;$ $w_0 = f_0 + v_0^2 (= rs_1'/h_1^3);$	3S	3S	3S
2	compute $1/s_1$ and s_0'': $w_1 = (1/w_0) z_0 (= h_1/s_1);$ $z_1 = k_1' w_1,\ s_0'' = z_1 + u_1;$	I, 2M	I, 2M	I, 2M
3	compute u': $w_2 = h_1^2 w_1,\ u_1' = w_2 w_1;$ $u_0' = s_0''^2 + w_2;$	2S	S, 2M	S, 2M
4	compute v': $w_3 = w_2 + k_1';$ $v_1' = h_1^{-1}(w_3 z_1 + w_2 u_1' + f_2 + v_1^2);$ $v_0' = h_1^{-1}(w_3 u_0' + f_1 + z_0);$	S, 3M	S, 3M	S, 5M
total		I, 6S, 5M	I, 5S, 7M	I, 5S, 9M

Results

Overview

	$h = X$	$h = h_1X$, h_1^{-1} small	$h = h_1X$	$h = X^2 + h_1X$, h_1^{-1} small
f_4 small	I, 6S, 5M	I, 5S, 7M	I, 5S, 9M	I, 6S, 10M
f_4 not small	n. a.	n. a.	n. a.	I, 6S, 12M

	$h = X^2 + h_1X$	$h = X^2 + h_1X + h_0$	all
f_4 small	I,5S, 15M	I, 6S, 21M	I, 5S, 22M
f_4 not small	I, 5S, 17M	n. a.	n. a.

Avoid inversions like in odd characteristic.

Different Coordinates $g = 2$, even q

affine $\mathcal{A}$:

$$D \sim [x^2 + u_1 x + u_0, v_1 x + v_0] \sim [u_1, u_0, v_1, v_0]$$

projective $\mathcal{P}$: L.; 2002

$$D \sim [U_1, U_0, V_1, V_0, Z] \sim [U_1/Z, U_0/Z, V_1/Z, V_0/Z]$$

new $\mathcal{N}$: L. 2002, contains also mixed coordinates

$$D \sim [U_1, U_1, V_1, V_0, Z_1, Z_2] \sim [U_1/Z_1^2, U_0/Z_1^2, V_1/(Z_1^3 Z_2), V_0/(Z_1^3 Z_2)]$$

recent $\mathcal{R}$: L. 2005, contains also mixed coordinates

$$D \sim [U_1, U_1, V_1, V_0, Z] \sim [U_1/Z, U_0/Z, V_1/Z^2, V_0/Z^2]$$

$\mathcal{P}$, $\mathcal{N}$, and $\mathcal{R}$: **no inversions**

Comparison – $h = x$

Doubling		Addition	
Operation	Costs	Operation	Costs
$2\mathcal{N} = \mathcal{N}$	$28\mathrm{M} + 5\mathrm{S}$	$\mathcal{R} + \mathcal{R} = \mathcal{R}$	$49\mathrm{M} + 8\mathrm{S}$
$2\mathcal{P} = \mathcal{P}$	$22\mathrm{M} + 6\mathrm{S}$	$\mathcal{P} + \mathcal{P} = \mathcal{P}$	$49\mathrm{M} + 4\mathrm{S}$
$2\mathcal{R} = \mathcal{R}$	$20\mathrm{M} + 8\mathrm{S}$	$\mathcal{N} + \mathcal{N} = \mathcal{N}$	$42\mathrm{M} + 6\mathrm{S}$
—	—	$\mathcal{A} + \mathcal{R} = \mathcal{R}$	$42\mathrm{M} + 7\mathrm{S}$
—	—	$\mathcal{A} + \mathcal{P} = \mathcal{P}$	$39\mathrm{M} + 4\mathrm{S}$
—	—	$\mathcal{A} + \mathcal{N} = \mathcal{N}$	$36\mathrm{M} + 6\mathrm{S}$
$2\mathcal{A} = \mathcal{A}$	$\mathrm{I} + 5\mathrm{M} + 6\mathrm{S}$	$\mathcal{A} + \mathcal{A} = \mathcal{A}$	$\mathrm{I} + 21\mathrm{M} + 3\mathrm{S}$

And more cases…

Halving

- Halving is operation invers to doubling.
- Needs trace, square-root, and half-trace computation like in ECC case.
- Kitamura, Katagi, Takagi: “A Complete Divisor Class Halving Algorithm for Hyperelliptic Curve Cryptosystems of Genus Two”, 2004.
  - Consider arbitrary binary $g = 2$ curves.
  - Much slower than doubling.
- Peter Birkner, “Efficient Divisor Class Halving on Genus Two Curves”, SAC 2006.
  - Uses L./Stevensen affine doubling to start $\Rightarrow$ much faster than [KKT].
  - Not competitive to DBL in polynomial basis; still slower in normal basis (from theoretical analysis).

Other curves

- Similar studies for $g = 3$.
  - In characteristic 2 also $h \in \mathbb{F}_q$ for $g = 3$ possible (not supersingular).
  - affine:
    Kuroki, Gonda, Matsuo, Chao, Tsujii (2002),
    Pelzl, Wollinger (2002), Guyot, Patankar (2003)
  - projective:
    X. Fan and Y. Wang (2004)
- There are works on non-hyperelliptic curves – so far not competitive, DLP weaker than on hyperelliptic curves.
- Larger genus looks insecure for crypto. Still interesting since arithmetic can be used in attacks (Weil descent), cf. works by Florian Hess and Kamal Khuri-Makdisi.

Pairings

Protocols

Diffie-Hellman Key exchange

Alice

1. secretly generates $a < |\langle P \rangle|$
2. computes $Q_1 = [a]P$
3. transmits Q_1
4. computes $[a]Q_2$

Bob

1. secretly generates $b < |\langle P \rangle|$
2. computes $Q_2 = [b]P$
3. transmits Q_2
4. computes $[b] \cdot Q_1$

$$[a]Q_2 = [ab]P = [b] \cdot Q_1$$

Common Key: the group element $k = [(ab)]P \in G$ can be used in symmetric encryption.

ElGamal Encryption

Public parameters:
Group G, generator P, ord$(P) = l$, some invertible embedding function $H : m \to G$.
Receiver has secret key s_A and public key $P_A = [s_A]P$.

- Encrypt message m
  - choose random integer k
  - compute $R = [k]P$ and $c = H(m) + [k]P_A$
- Decrypt ciphertext (R, c)
  - compute $S = [s_A]R$
  - obtain $m = H^{-1}(c - S)$

  (This gives m as $S = [s_A]R = [ks_A]P = [k]P_A$).

DL systems

These systems assume that the Discrete Logarithm Problem (DLP) is hard to solve, i.e.

- given P and $P_A = [s_A]P$
- it is hard to find s_A.

The Computational Diffie-Hellman Problem (CDHP) is the problem

- given P, $P_A = [s_A]P$, and $P_B = [s_B]P$
- compute $[s_A s_B]P$.

The Decisional Diffie-Hellman Problem (DDHP) is the problem

- given P, $P_A = [s_A]P$, $P_B = [s_B]P$ and $R = [r]P$
- decide whether $R = [s_A s_B]P$.

Pairings

Let $(G_1, +), (G'_1, +)$ and $(G, \cdot)$ be cyclic groups of prime order ℓ and let

$$e : G_1 \times G'_1 \to G$$

be a an efficiently computable map satisfying

- $e(P + Q, R') = e(P, R')e(Q, R')$
- $e(P, R' + S') = e(P, R')e(P, S')$
- The map is non-degenerate in the first argument, i.e. if $e(P, R') = 1$ for all $R' \in G'_1$ for some P then P is the identity in G_1

Then e is called a bilinear map or pairing.

In protocol papers often $G_1 = G'_1$ (keyword “distortion map”).

Consequences I

Assume that $G_1 = G_1'$ and hence

$$e(P,P) \neq 1.$$

Then for all triples $(P_1, P_2, P_3) \in \langle P \rangle^3, P_i = [a_i]P$ one can decide efficiently whether

$$P_3 = [a_1 a_2]P, \text{ i.e. } \log_P(P_3) = \log_P(P_1)\log_P(P_2)$$

by comparing $e(P_1, P_2) \stackrel{?}{=} e(P, P_3)$, i.e.

$$e(P_1, P_2) = e([a_1]P, [a_2]P) = e(P, [a_1 a_2]P) = e(P, P_3).$$

Thus the DDHP is easy! CDHP can still be hard!
Keyword: gap groups.

Consequences II

Even for $G_1 \neq G_1'$ can transfer the DLP in G_1 to a DLP in G.

Let $P' \in G_1'$ be such that

$$e(P, P') \neq 1.$$

Given DLP $P, P_A = [s_A]P$ compute

$g = e(P, P')$ and $h = e(P_A, P') = e([s_A]P, P') = e(P, P')^{s_A} = g^{s_A}$.

This gives DLP in G with $g, h = g^{s_A}$ related by the same scalar s_A as original system.

DL system G_1 is at most as secure as the system G!

Pairing based protocols I

Joux, ANTS 2000, one round tripartite key exchange

Let P, P' be generators of G_1 and G_1' respectively.
Users A, B and C compute joint secret from their secret contributions a, b, c as follows (A's perspective)

- Compute and send $[a]P, [a]P'$.
- Upon receipt of $[b]P$ and $[c]P'$ put $k = (e([b]P, [c]P'))^a$

The resulting element k is the same for each participant as

$$k = (e([b]P, [c]P'))^a = (e(P, P'))^{abc} = (e([a]P, [c]P'))^b = (e([a]P, [b]P'))^c.$$

- Obvious saving in first step if $G_1 = G_1'$.
- Only one user needs to do both computations.

Pairing based protocols II

- Sakai-Ohgishi-Kasahara, 2000 ID-based key exchange
- Boneh and Franklin, Crypto 2001, ID-based cryptography

Idea: user's identity defines his public key.

Consequences

- Advantage in ID-based crypto if recipient is not in system or sender wants to force use of a fresh key (other applications possible).
- No need for PKI, can avoid need for authentication.
- Set-up requires a trusted authority (TA) which can compute the secret key for a given public key.

ID-based cryptography II

- Let $H : \{0,1\}^* \to G_1'$ be hash function.
- Master secret key of TA is s, public key is $P_{pub} = [s]P$.
- Public key of ID represented by string ID is $H(ID) \in G_1'$.
- Secret key $S' = [s]H(ID) \in G_1'$ computable only by TA.
- Idea is that from

  $$P_{pub}, H(ID) \text{ and } k$$

  or from

  $$[k]P \text{ and } [s]H(ID)$$

  both parties can compute

  $$e([k]P, [s]H(ID)) = (e(P, H(ID)))^{ks} = (e([s]P, H(ID)))^k.$$

Security assumptions

(Just a few from the huge zoo.)

- Clearly these systems require that the DLP is hard in the groups.
- Additionally we define the following computational and decisional problems. To ease notation let $G_1 = G_1'$ and $g = e(P, P)$.
  - Computational Bilinear Diffie-Hellman Problem (CBDHP):
    Compute $g^{s_A s_B s_C}$ given $[s_A]P, [s_B]P, [s_C]P$, and P
  - Decisional Bilinear Diffie-Hellman Problem (DBDHP):
    Given $P, [s_A]P, [s_B]P, [s_C]P$ and g^r decide whether $g^r = g^{s_A s_B s_C}$.

Pairings on (hyper)elliptic curves

Prerequisites

We want to define pairings

$$G_1 \times G_1' \to G$$

preserving the group structure.

The pairings we use map to the multiplicative group of a finite extension field $\mathbb{F}_{q^k}$. To embed the points of order ℓ the embedding degree k has to be chosen large enough so that the ℓ-th roots of unity are in $\mathbb{F}_{q^k}^*$.

So choose

$$k \text{ minimal with } \ell \mid q^k - 1.$$

Tate-Lichtenbaum pairing I

$\mathrm{Pic}^0_C(\mathbb{F}_{q^k})[\ell]$: divisor classes on C of order ℓ defined over $\mathbb{F}_{q^k}$.

$\bar{D}_1 \in \mathrm{Pic}^0_C(\mathbb{F}_{q^k})[\ell] \Rightarrow \exists F_{D_1}$ such that $\ell D_1 \sim \mathrm{div}(F_{D_1})$, where D_1 represents the class $\bar{D}_1$.

Let $\bar{D}_2 \in \mathrm{Pic}^0_C(\mathbb{F}_{q^k})$ be represented by D_2 with

$$\mathrm{support}(D_2) \cap \mathrm{support}(D_1) = \emptyset.$$

Tate-Lichtenbaum pairing

$$T_\ell(\bar{D}_1, \bar{D}_2) = F_{D_1}(D_2) = \frac{\prod_{i=1}^n F_{D_1}(P_i)}{\prod_{j=1}^n F_{D_1}(Q_j)}$$

for $D_2 = \sum_{i=1}^n P_i - \sum_{j=1}^n Q_j$.

Tate-Lichtenbaum pairing II

This

$$T_\ell(\bar{D}_1, \bar{D}_2) = F_{D_1}(D_2)$$

defines a bilinear and non-degenerate map

$$T_\ell : \mathrm{Pic}^0_C(\mathbb{F}_{q^k})[\ell] \times \mathrm{Pic}^0_C(\mathbb{F}_{q^k})/\ell\mathrm{Pic}^0_C(\mathbb{F}_{q^k}) \longrightarrow \mathbb{F}^*_{q^k}/\mathbb{F}^{*\ell}_{q^k}$$

as ℓ-folds are in the kernel of T_ℓ.
Namely, if $\bar{D}_2 = [\ell]\bar{D}_3$ then

$$F_{D_1}(D_2) = F_{D_1}(D_3)^\ell = 1.$$

To achieve unique value in $\mathbb{F}_{q^k}$ rather than class do final exponentiation

$$\tilde{T}_\ell = T_\ell(\bar{D}_1, \bar{D}_2)^{(q^k-1)/\ell}.$$

Tate-Lichtenbaum pairing III

For elliptic curves use isomorphism

$$\mathrm{Pic}^0_E(\mathbb{F}_{q^k}) \cong E(\mathbb{F}_{q^k})$$

to define pairing on points $T_\ell(P, Q)$, with

$$D_1 = P - P_\infty, D_2 = (Q \oplus R) - R$$

for some R.

Build F iteratively by Miller's algorithm (double-and-add).

Often

$$T_\ell : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_{q^k})/\ell E(\mathbb{F}_{q^k}) \rightarrow \mathbb{F}^*_{q^k}/\mathbb{F}^{*\ell}_{q^k}.$$

Miller's algorithm

In: $\ell = \sum_{i=0}^{n-1} \ell_i 2^i$, $P, Q \oplus R, R$
Out: $T_\ell(P, Q)$

1. $T \leftarrow P$, $F \leftarrow 1$
2. for $i = n-2$ downto 0 do
   (a) Calculate lines l and v in doubling $[2]T$
   $T \leftarrow [2]T$
   $F \leftarrow F^2 \cdot l(Q \oplus R)v(R)/(l(R)v(Q \oplus R))$
   (b) if $\ell_i = 1$ then
   Calculate lines l and v in addition $T \oplus P$
   $T \leftarrow T \oplus P$
   $F \leftarrow F \cdot l(Q \oplus R)v(R)/(l(R)v(Q \oplus R))$
3. return F

Efficient computation
(for much more ask Paulo!!)

k minimal

Observation:
k is minimal with $\ell \mid q^k - 1$, so final exponentiation $T_\ell(\bar{D}_1, \bar{D}_2)^{(q^k-1)/\ell}$ removes all terms from subfields.

- Choose $R \in E(\mathbb{F}_q)$, computation of $l(R)$ and $v(R)$ not needed.
- Choose $R = P_\infty$,

$$\tilde{T}_\ell(P, Q) = F_{P-P_\infty}(Q)^{(q^k-1)/\ell}.$$

- If protocol allows, choose $Q = (x_Q, y_Q) \in \mathbb{F}_{q^k}$ with x in subfield, this avoids computation of $v(Q)$. This means no extra division!

Improvements to Miller

Observation:
Let $N = c\ell$ for some c, then

$$\tilde{T}_\ell(P,Q) = T_\ell(P,Q)^{(q^k-1)/\ell} = T_N(P,Q)^{(q^k-1)/N}.$$

- Other addition chains can be used in Miller's algorithm, e.g. NAF.
  Other bases might be good depending on characteristic of the field.
- Use multiple N with low Hamming weight (in p-adic representation).
  For supersingular curves choosing N as number of points gives fast final exponentiation.

Special curves

Duursma-Lee and Barreto-Galbraith-O hEigeartaigh-Scott obtain further speed-up by choosing supersingular curves of genus $(p-1)/2$ over $\mathbb{F}_{p^r}$.

Main ideas are to use p as basis in addition chain $\Rightarrow$ much shorter chain and take into account that each order divides p^r+1.

Key words are “eta-pairing” and “eta-T-pairing”.

Recent paper ePrint 2006/110 by Hess, Smart, and Vercauteren introduces ate-pairing.

Special divisors

joint work with G. Frey.

- Useful if no map $J_C(\mathbb{F}_q) \mapsto J_C(\mathbb{F}_{q^k})$ given.
- Katagi, Kitamura, Akishita, and Takagi suggested to use special divisors $\bar{D} = P - P_\infty$ as basis for DL system, also for $g > 1$
  $\Rightarrow$ additions involving special divisor much cheaper.
- Use subset of special divisors in $J_C(\mathbb{F}_{q^k})$ as second argument of the pairing – or even restricted to points with x-coordinate in subfield.
- Evaluation of functions only in one point much faster.
- Speed-up grows with genus!
- For non-degeneracy etc. see ANTS 2006 proceedings.

Supersingular and ordinary

Definition

Let E be an elliptic curve defined over $\mathbb{F}_q, q = p^r$.
E is supersingular if

- $E[p^s](\overline{\mathbb{F}}_q) = \{P_\infty\}$.
- $|E(\mathbb{F}_q)| = q - t + 1$ with $t \equiv 0 \bmod p$.
- End_E is order in quaternion algebra.

Otherwise it is ordinary and one has $E[p^s](\overline{\mathbb{F}}_q) = \mathbb{Z}/p^s\mathbb{Z}$.

These statements hold for all s if they hold for one.
End_E order in quaternion algebra means that there are maps which are linearly independent of the Frobenius endomorphism. They are called distortion maps.

Example

Consider

$$y^2 + y = x^3 + a_4 x + a_6 \text{ over } \mathbb{F}_{2^r},$$

so $q = 2^r$.

Negative of $P = (a, b)$ is $-P = (a, b + 1)$,
$\Rightarrow$ no affine point with $P = -P$ since $b \neq b + 1$,
$\Rightarrow$ even number of affine points, one point P_∞,

$\Rightarrow |E(\mathbb{F}_q)| = q - t + 1 = 2^r - t + 1$ is odd, so t is even.

This curve is supersingular (using the second criterion).

Distortion map I

For supersingular curves it is possible to find maps $\phi : E(\mathbb{F}_q) \to E(\mathbb{F}_{q^k})$ that map to a linearly independent subgroup, i.e.

$$T'_\ell(P,P) \neq 1 \text{ for } T'_\ell(P,P) = T_\ell(P,\phi(P)).$$

(This needs that there are independent endomorphisms, so no chance for ordinary curves).
Examples:

- $y^2 = x^3 + a_4 x$, for $p \equiv 3 \pmod 4$.
  Distortion map $(x,y) \mapsto (-x, iy)$ with $i^2 = -1$
- $y^2 = x^3 + a_6$, for $p \equiv 2 \pmod 3$.
  Distortion map $(x,y) \mapsto (jx, y)$ with $j^3 = 1, j \neq 1$,

In both cases, $\#E(\mathbb{F}_p) = p + 1$, $k = 2$.

Distortion maps II

- Over $\mathbb{F}_{2^d}$ consider
$$y^2 + y = x^3 + x + a_6, \text{ with } a_6 = 0 \text{ or } 1$$
and distortion map
$$(x,y) \mapsto (x+s^2, y+sx+t),\ s,t \in \mathbb{F}_{2^{4d}},\ s^4+s=0,\ t^2+t+s^6+s^2=0.$$
$$\#E(\mathbb{F}_{2^d}) = 2^d + 1 \pm 2^{(d+1)/2},\ k = 4.$$
- Over $\mathbb{F}_{3^d}$ consider
$$y^2 = x^3 + x + a_6, \text{ with } a_6 = \pm 1$$
and distortion map
$$(x,y) \mapsto (-x+s, iy) \text{ with } s^3 + 2s + 2a_6 = 0 \text{ and } i^2 = -1.$$
$$\#E(\mathbb{F}_{3^d}) = 3^d + 1 \pm 3^{(d+1)/2},\ k = 6.$$

Application of distortion maps

- Distortion maps allow to state the protocols as using bilinear map
$$G_1 \times G_1 \to G.$$
- Scalar multiplications of input point can be performed over $\mathbb{F}_q$ and then mapped into $E(\mathbb{F}_{q^k})$, this saves computation time
$\Rightarrow$ obvious computational savings in the above protocols.
- Can be used to map on points with x-coordinate in subfield.
- Systems are easier to state.
- Really short signatures (see next slide).

Short signatures

Boneh, Lynn, and Shacham, Asiacrypt 2001

Requires hash function $H : \{0,1\}^* \to G_1$.

User secret key s, public key $[s]P$.

Signature of message m:
Compute and send $S = [s]H(m)$.

Verification:
Check if

$$e([s]P, H(m)) = e(P, S).$$

Point compression to the x-coordinate achieves signatures given by one field element, i.e. of half the length of ECDSA.

Embedding degree

Embedding degrees I

Let E be supersingular and $q = p \geq 5$, i.e $p > 2\sqrt{p}$.

Hasse's Theorem states $|t| \leq 2\sqrt{p}$ and E supersingular implies $t \equiv 0 \bmod p$, so $t = 0$ and

$$|E(\mathbb{F}_p)| = p + 1.$$

Obviously

$$(p+1) \mid p^2 - 1 = (p+1)(p-1)$$

so $k \leq 2$ for supersingular curves over prime fields.

Embedding degrees II

Let E be supersingular.
Menezes, Okamoto, and Vanstone show:

- in characteristic 2 we have $k \leq 4$,
- in characteristic 3 we have $k \leq 6$,
- over prime fields $\mathbb{F}_p$ with $p \geq 5$ we have $k \leq 2$,

and these bounds are attained.

Now or soon, larger k become necessary, no chance with supersingular elliptic curves, little hope for larger genus hyperelliptic (cf. following slide).

Parameter sizes

For DLP in groups one has square-root attacks, for finite fields index calculus methods are more efficient. A common requirement is

- $\#E(\mathbb{F}_q) \sim q$: 160 bits
- q^k: 1024 bits
- optimal embedding degree would be

$$k = 1024/160 = 6.4$$

However, 6 is only attained for small characteristic and there Coppersmith methods require larger fields, cf. Page, Smart, and Vercauteren.
Difference between sizes grows $\Rightarrow$ larger k becomes optimal; might be interesting to revive $g = 4$ particularly in light of Frey/L. 2006.

MNT curves

Find ordinary curves with small embedding degree! Thus force

$$q + 1 - t \equiv 0 \pmod{\ell} \text{ and } q^k - 1 \equiv 0 \pmod{\ell}.$$

- Original proposal by Miyaji, Nakabayashi, and Takano.
- Dupont, Enge, Morain: small k by CM
- Barreto, Lynn, Scott; Brezing, Weng;
  Curves for $k = 3, 4$ and 6.
- Generalization to co-factors, $k = 5, 10, 12$
  Galbraith, McKee, Valença.
- Family of curves with $k = 12$ by Barreto and Naehrig.
- Family of curves with $k = 10$ by D. Freeman.

Supersingular vs. ordinary

Supersingular:

- pro: known, speed ups by choosing $N = c\ell$, eta-T-pairing, distortion maps
- con: only small k, far too large field size for prime fields, small number of curves

Ordinary:

- pro: can be used over prime fields, more and larger embedding degrees possible, more (but also finite) choice
- con: no distortion maps, no reason why good multiple N should exist, no loop-shortening

Outlook

Pairings are a very active area of research, e.g. on the following tasks

- find more protocols / solve posed problems,
- speed up the computation,
- find more MNT curves,
- consider other curves, e.g. hyperelliptic curves.

Great collection at

P. S. L. M. Barreto, “The Pairing-Based Crypto Lounge”

`http://planeta.terra.com.br/informatica/paulobarreto/pblounge.html`

- Break system???

Literature

- P. S. Barreto, S. Galbraith, C. O hEigeartaigh, M. Scott
  "Efficient Pairing Computation on Supersingular Abelian Varieties"
- R. Dutta, R. Barua, and P. Sakar
  "Pairing-based cryptography: A survey"
- D. Hankerson and A. J. Menezes and S. A. Vanstone
  "Guide to elliptic curve cryptography"
- Chapter in
  "Advances in Elliptic Curve Cryptography",
  by I. F. Blake, G. Seroussi, and N. P. Smart (Editors)
- Chapters in
  "Handbook of Elliptic and Hyperelliptic Curve Cryptography", by H. Cohen and G. Frey (Editors)