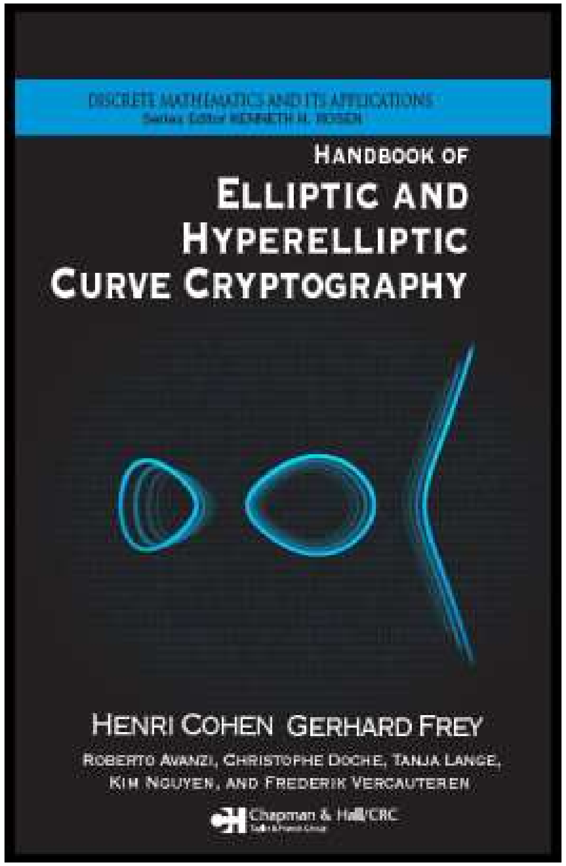
Halftime Advertising II



Handbook of Elliptic
and Hyperelliptic Curve Cryptography

R. Avanzi, H. Cohen, C. Doche,
G. Frey, T. Lange, K. Nguyen,
F. Vercauteren

CRC Press 2005

Sample chapter and more information now available at

`www.hyperelliptic.org/HEHCC`

Please notify us about typos and omissions!

Elliptic vs. hyperelliptic, part 2

Tanja Lange

Technische Universiteit Eindhoven

tanja@hyperelliptic.org

25.04.2007

Hyperelliptic curves

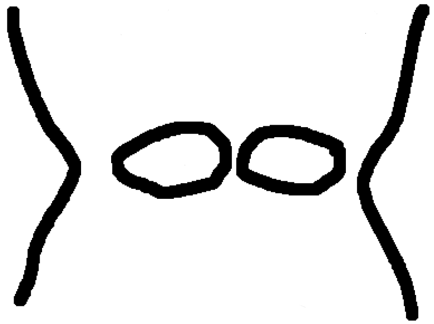
Affine equation of hyperelliptic curve of genus g (with $\mathbb{F}_q$-rational Weierstraß-point at infinity)

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

$h(x), f(x) \in \mathbb{F}_q[x]$, 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.



(There are also the real quadratic function fields with $\deg(f) = 2g + 2 \ldots$

Hyperelliptic curves

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

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

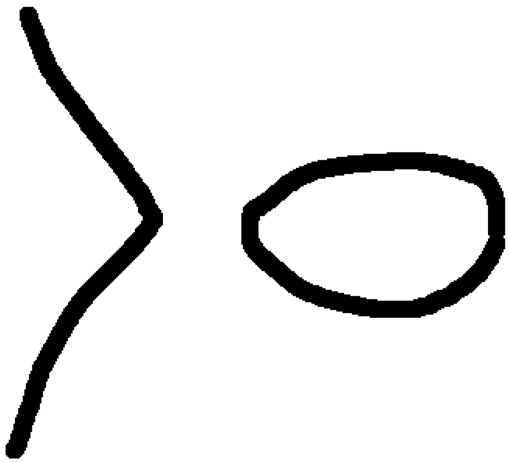
$h(x), f(x) \in \mathbb{F}_q[x]$, f monic, $\deg f = 2g + 1, \deg h \leq g$
non singular, i. e. in no point $(a, b) \in C/\overline{\mathbb{F}_q}$ over the algebraic closure $\overline{\mathbb{F}_q}$ we have both partial derivatives vanishing, i.e.
$2b + h(a) = 0$ and $h'(a)b - f'(a) = 0$.

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

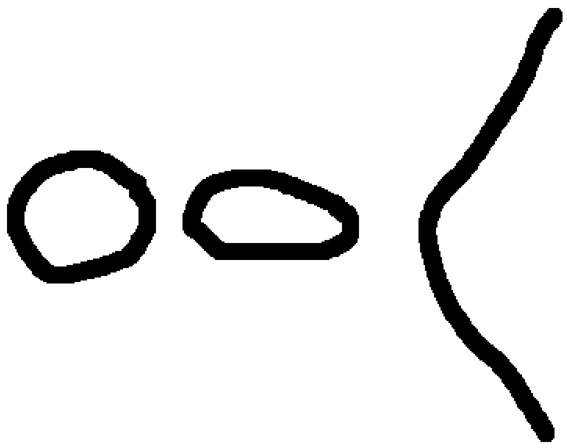
(There are also the real quadratic function fields with $\deg(f) = 2g + 2$ . . .
. . . the step-brothers . . .)

The Opponents

Elliptic vs. Hyperelliptic



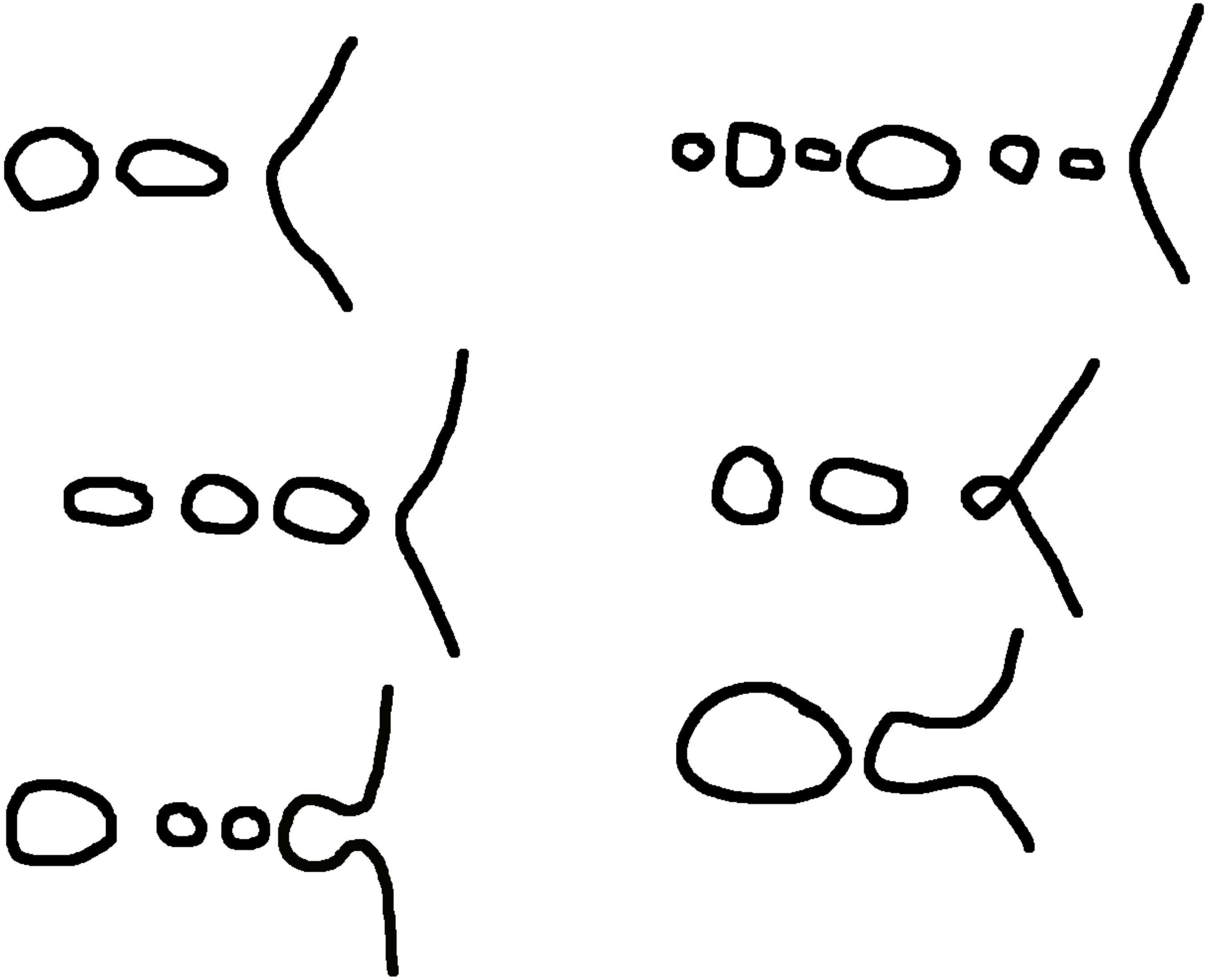
$g = 1$



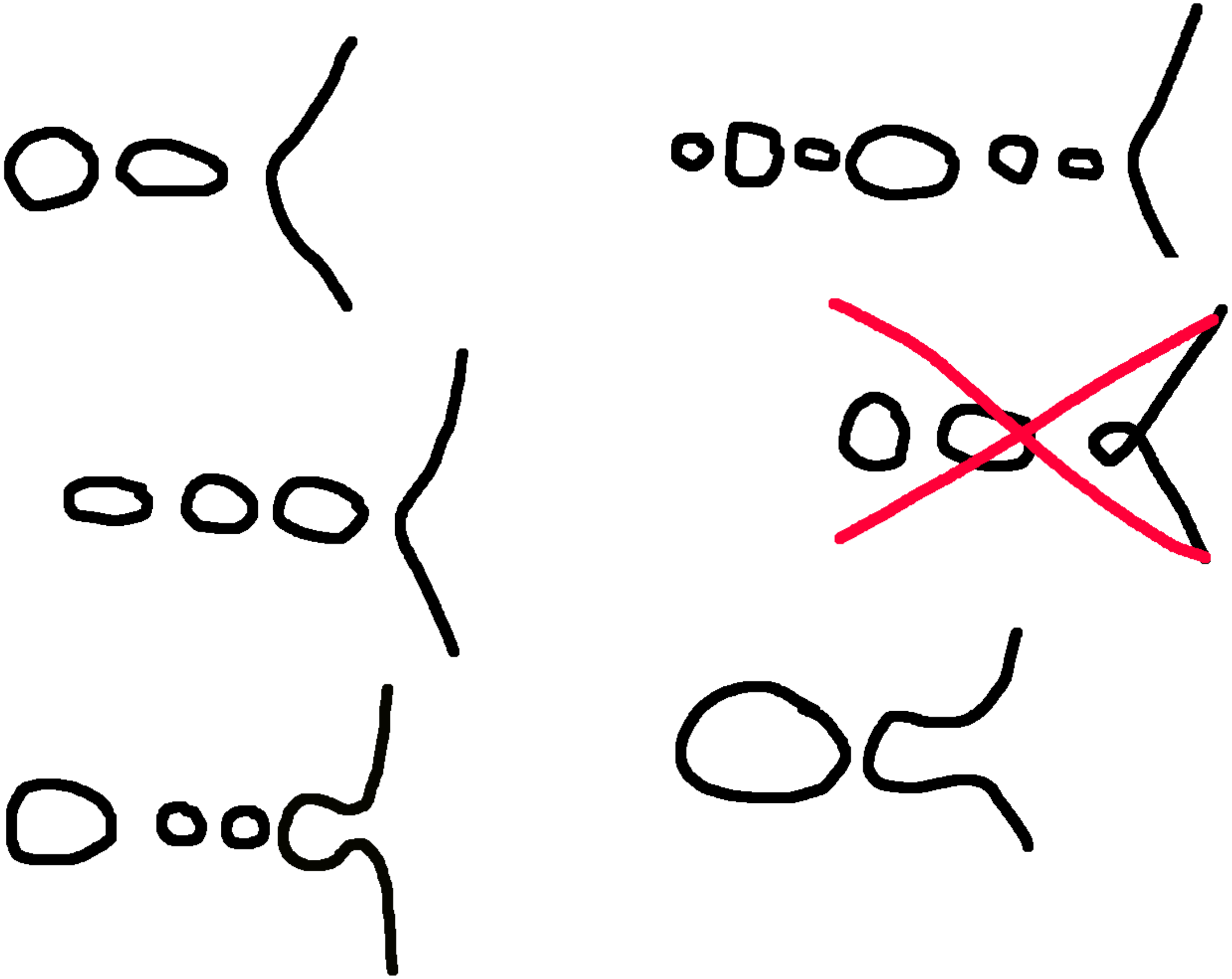
$g = 2$

(. . . already after some transformations . . .)

. . . and friends



. . . and friends



Divisor class group

For details see Peter Birkner's talk a few weeks ago, ... or THE book.
The divisor class group of degree zero is the factor group of the group of divisors of degree zero Div_C^0 modulo the principal divisors.

$$\mathrm{Pic}_C^0 = \mathrm{Div}_C^0/\mathrm{Princ}_C.$$

Each divisor class has a unique reduced representative

$$D = \sum_{\substack{i=1 \\ P_i \in C(\overline{\mathbb{F}_q}) \setminus \{P_\infty\}}}^{m} P_i - mP_\infty$$

with $P_i \neq P_\infty$, $P_i \neq -P_j$ for $i \neq j$ and $m \leq g$.

Representation

- Idea: use polynomials to represent divisors, ignore P_∞ – multiplicity dictated by affine part.
- Let D be semi-reduced $D = \sum_{i=1}^m P_i - mP_\infty$ with $P_i = (x_i, y_i)$. Put

  $$u(x) = \prod_{i=1}^m (x - x_i) \text{ and define } v \text{ by } v(x_i) = y_i \text{ with multiplicity,}$$

  i.e. if P_i appears n_i times then

  $$\left(\frac{d}{dx}\right)^j \left[v(x)^2 + v(x)h(x) - f(x)\right]_{|x=x_i} = 0, \quad \text{for } 0 \le j \le n_i - 1.$$

- Corresponds to ideal $\langle u(x), y - v(x) \rangle$ in coordinate ring of curve.

Arithmetic on hyperelliptic curves

Composition & Reduction (Cantor/Koblitz)

`IN:` $D_1 = [u_1, v_1], D_2 = [u_2, v_2], C : y^2 + h(x)y = f(x)$
`OUT:` $D = [u, v]$ reduced with $D \sim D_1 + D_2$

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. $u = \frac{u_1 u_2}{d^2} \quad v = \frac{s_1 u_1 v_2 + s_2 u_2 v_1 + s_3(v_1 v_2 + f)}{d} \bmod u$

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

Arithmetic on hyperelliptic curves

Composition & Reduction (Cantor/Koblitz)

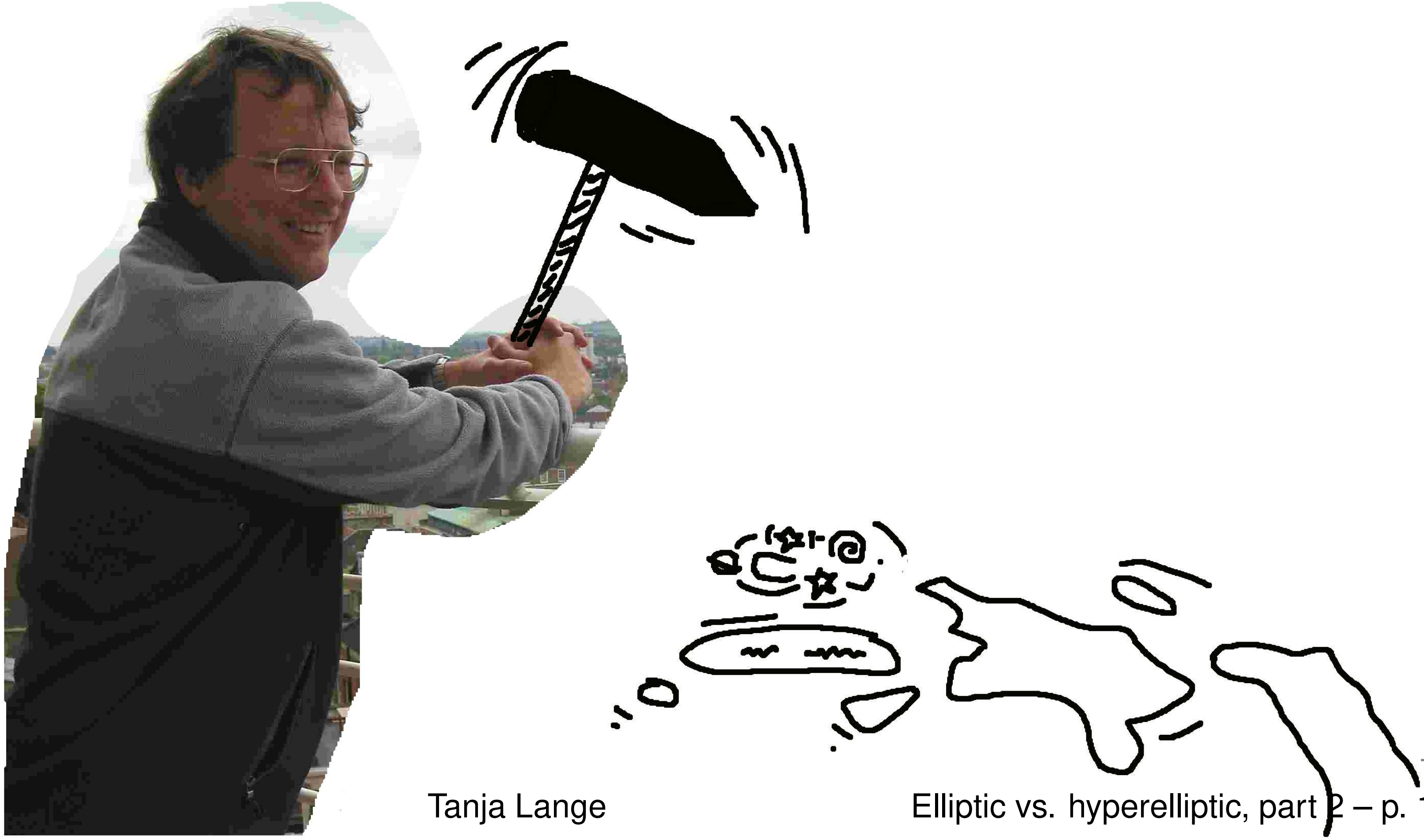
`IN:` $D_1 = [u_1, v_1], D_2 = [u_2, v_2], C : y^2 + h(x)y = f(x)$
`OUT:` $D = [u, v]$ reduced with $D \sim D_1 + D_2$

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 $u' = \frac{f - vh - v^2}{u} \quad v' = (-h - v) \bmod u'$
6. if $\deg u' > g$ put $u := u', v := v'$ goto step 5
7. make u monic

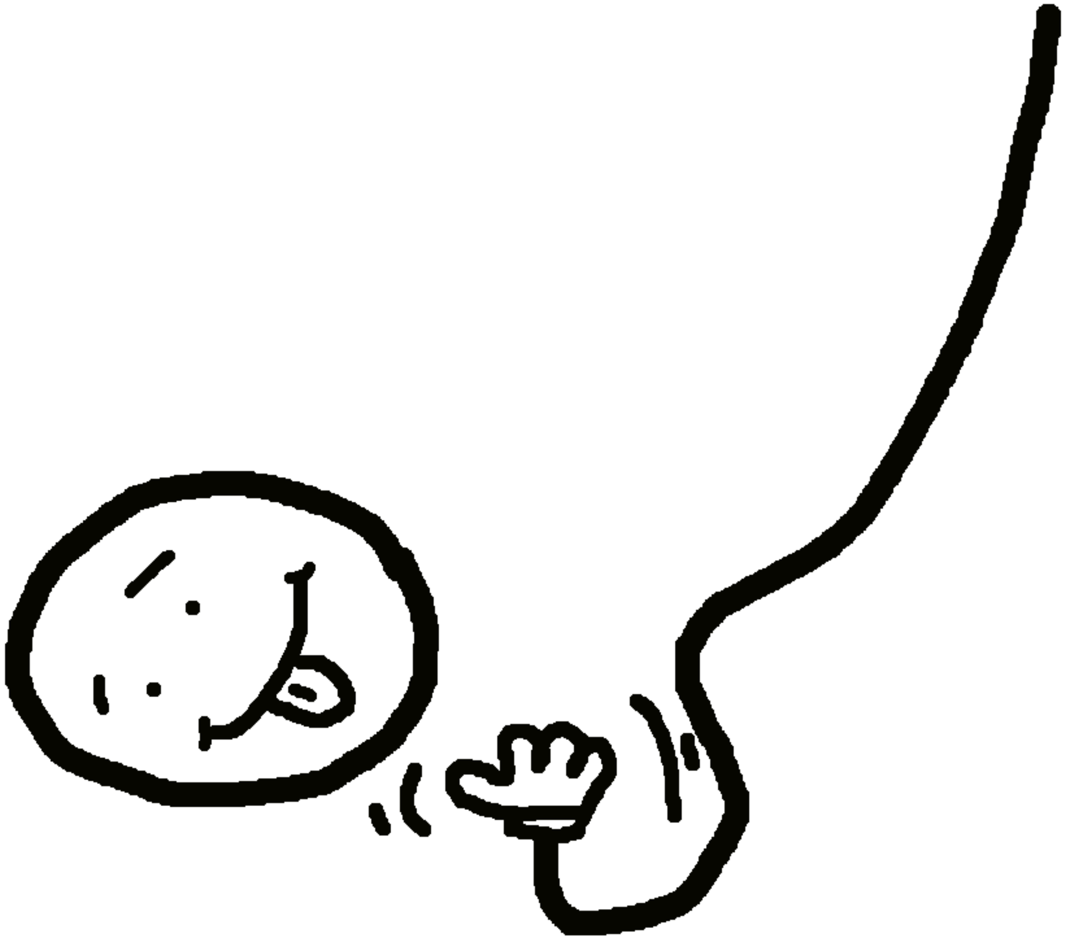
This result $[u, v]$ corresponds to a reduced divisor.

First implementation results

Nigel P. Smart: On the Performance of Hyperelliptic Cryptosystems. EUROCRYPT 1999: 165-175:
Performance of hyperelliptic curves is really bad!



Spotlight on elliptic curve



More destruction results . . .

Index calculus attacks on HECC

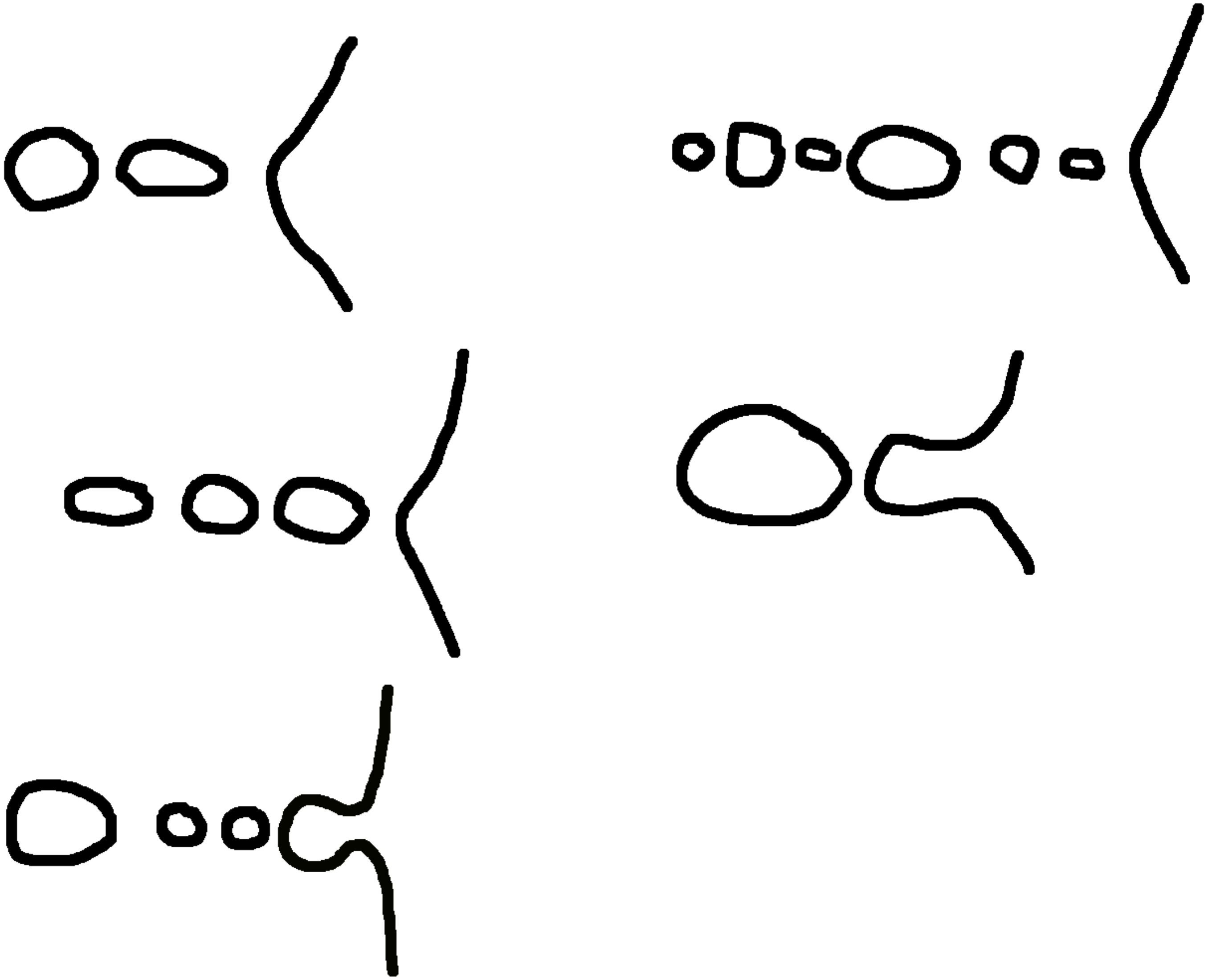
- Adleman, De Marrais, and Huang, A subexponential algorithm over hyperelliptic curves of large genus over GF(q), 1999.
- Gaudry, An algorithm for solving the discrete logarithm problem on hyperelliptic curves, 2000.
- Enge, Computing discrete logarithms in high-genus hyperelliptic Jacobians in provably subexponential time, 2003.
- Enge and Gaudry, A general framework for subexponential discrete logarithm algorithms, 2002.

. . . and more . . .

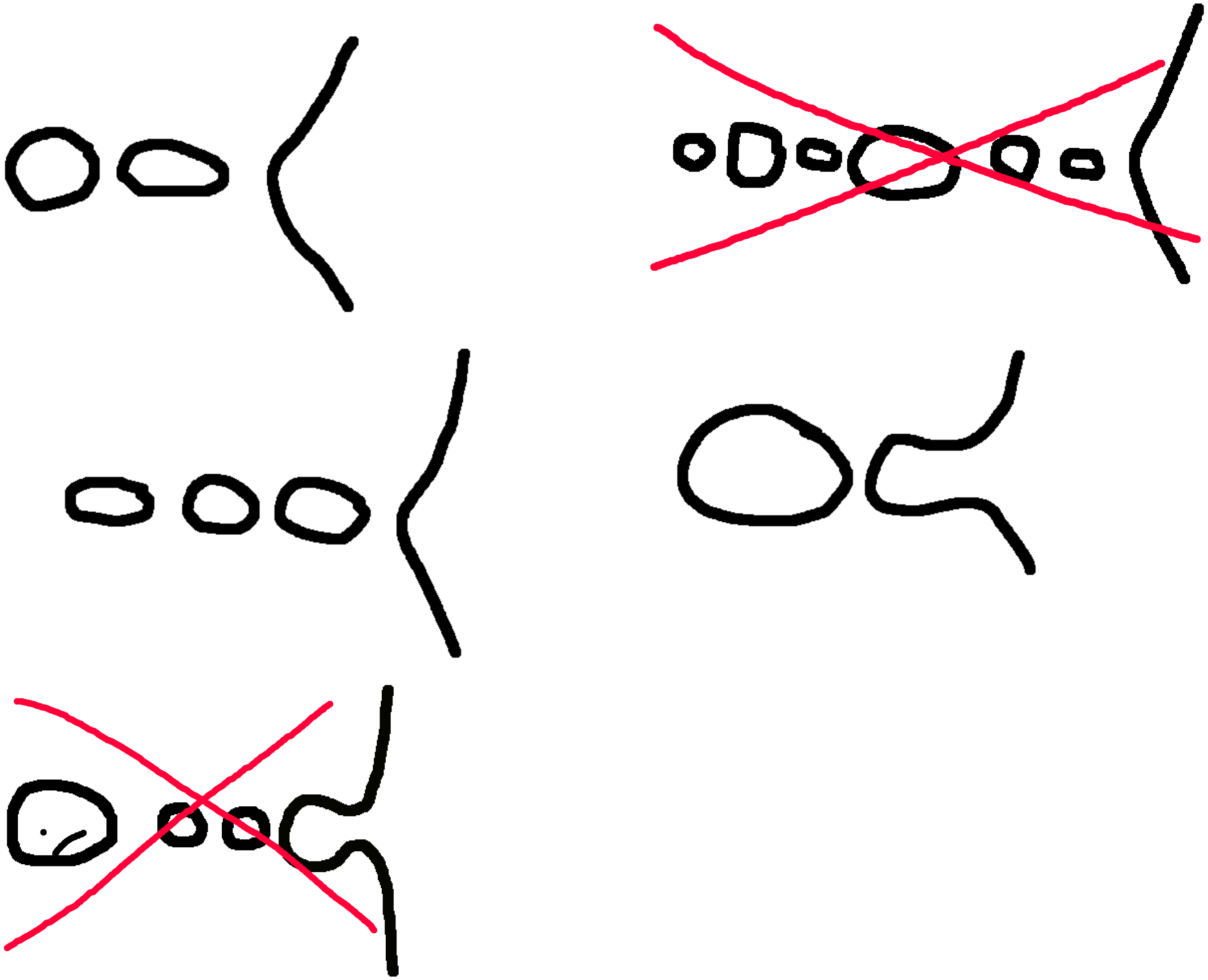
- Thériault. Index calculus attack for hyperelliptic curves of small genus. 2003.
- Gaudry, Thériault, and Thomé. A double large prime variation for small genus hyperelliptic index calculus. 2005.
- Diem. Index calculus with double large prime variation in class groups of plane curves of small degree, 2006.
- Gaudry, Thomé, Thériault, and Diem. A double large prime variation for small genus hyperelliptic index calculus. (accepted).

This rules out genus $g \geq 4$ and scratches $g = 3$.

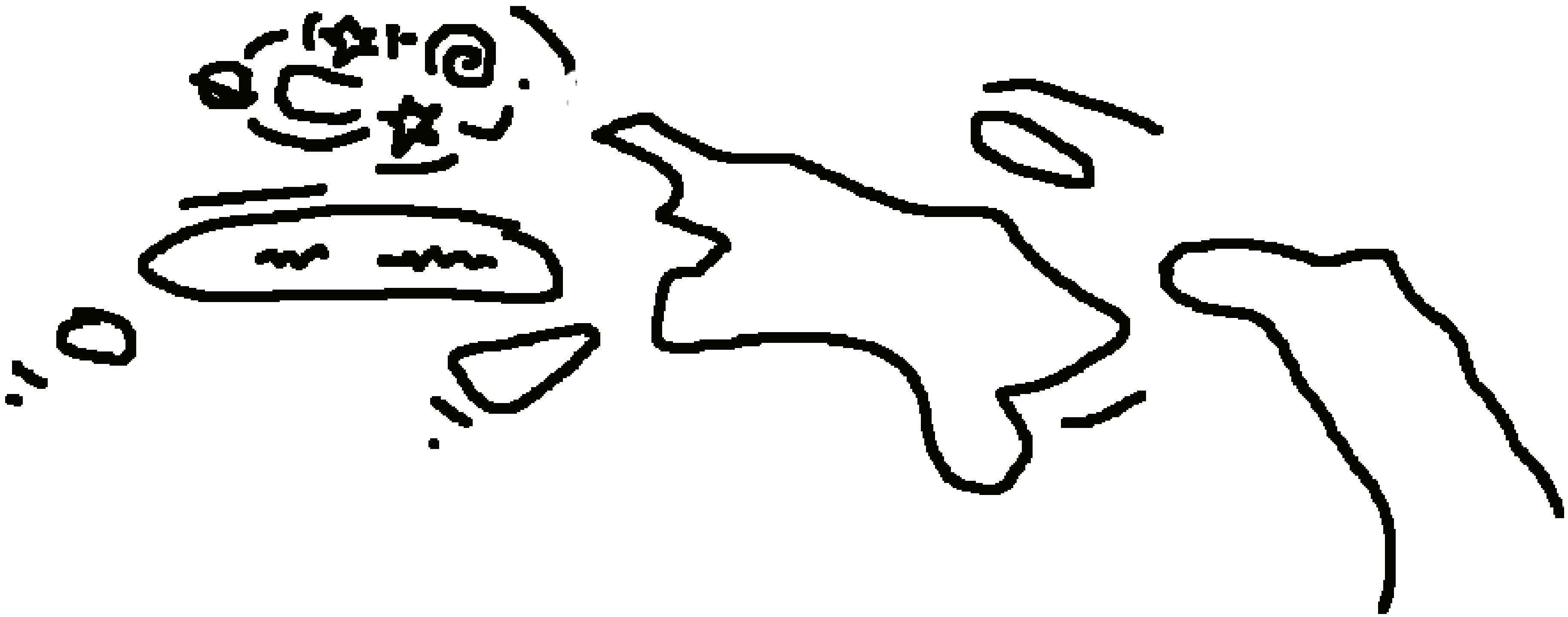
Fewer friends



Fewer friends

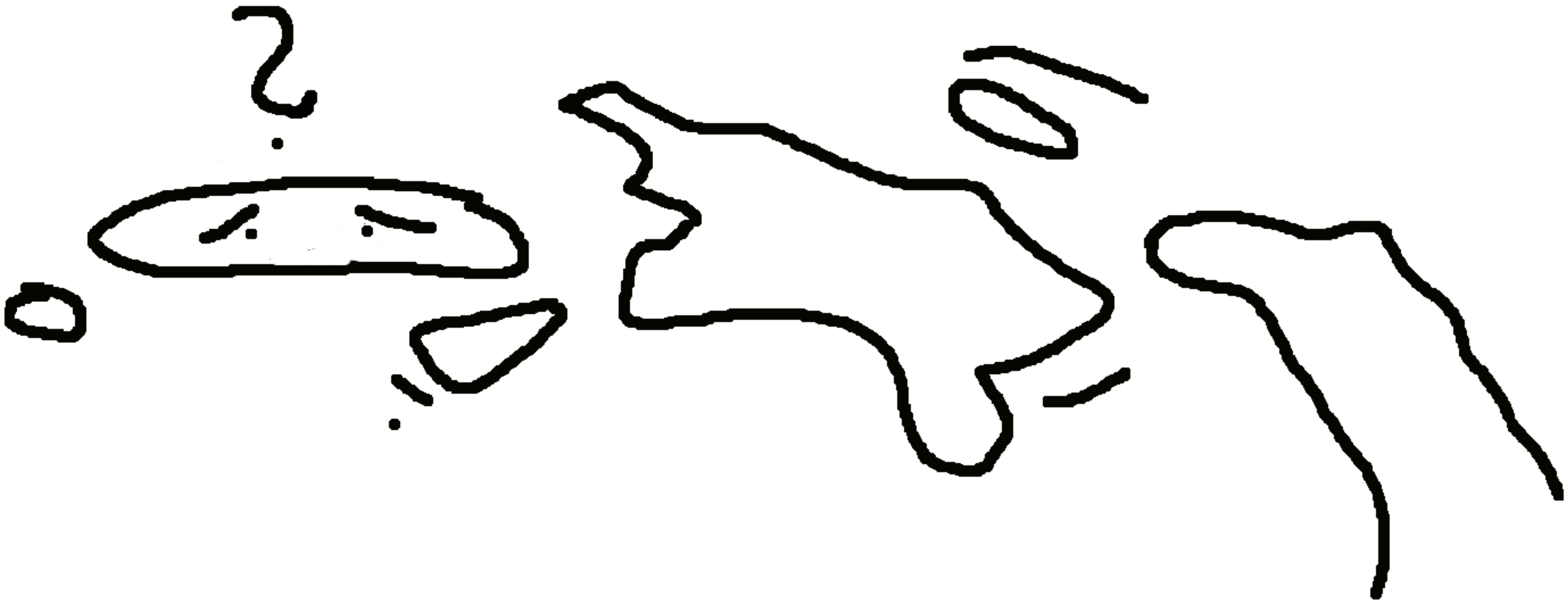


Game over?



. . . slow, most friends insecure, . . .

Any hope?



Never fear!

- 

  Robert Harley (while counting points with Pierrick Gaudry) is annoyed about slow $g = 2$ operations
  We don't use Cantor for $g = 1$ why should we for $g = 2$?
- Develops first "explicit formulae" for genus 2 curves, close to no explanation.
- Main speed-ups by avoiding computation of unused coefficients.
- Explicit use of Karatsuba multiplication, reduction and resultants for inversion modulo polynomials.
- Requires case distinction.

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

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

Some more support

- Gaudry, Harley. Counting points on hyperelliptic curves over finite fields. 2000.
- Hess, Seroussi, and Smart, Two topics in hyperelliptic cryptography, 2000.
  Gives efficient compression method.
- Weng. Konstruktion kryptographisch geeigneter Kurven mit komplexer Multiplikation. 2001.
  One can efficiently construct curves with the CM method over large prime fields.
- Gaudry, Schost. Construction of secure random curves of genus 2 over prime fields. 2004.
  Reaches prime fields of 80 bits & proposes some curves with almost prime order divisor class group.

General doubling, $g = 2$

Doubling, $\deg u = 2$			
Input Output	$[u, v], u = x^2 + u_1 x + u_0, v = v_1 x + v_0$ $[u', v'] = 2[u, v]$		
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 =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 $inv' = inv r$: $inv_1' = -\tilde{v}_1, inv_0' = \tilde{v}_0 - w_3$;		
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: $w_1 = 1/(r s_1')(= 1/r^2 s_1), w_2 = r w_1 (= 1/s_1'), w_3 = s_1'^2 w_1 (= s_1)$; $w_4 = r w_2 (= 1/s_1), w_5 = w_4^2, s_0'' = s_0' w_2$;	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 $u' = s^2 + (h + 2v)s/u + (v^2 + hv - f)/u^2$: $u_0' = s_0''^2 + w_4(h_2(s_0'' - u_1) + 2v_1 + h_1) + w_5(2u_1 - f_4), u_1' = 2s_0'' + h_2 w_4 - w_5$;	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	

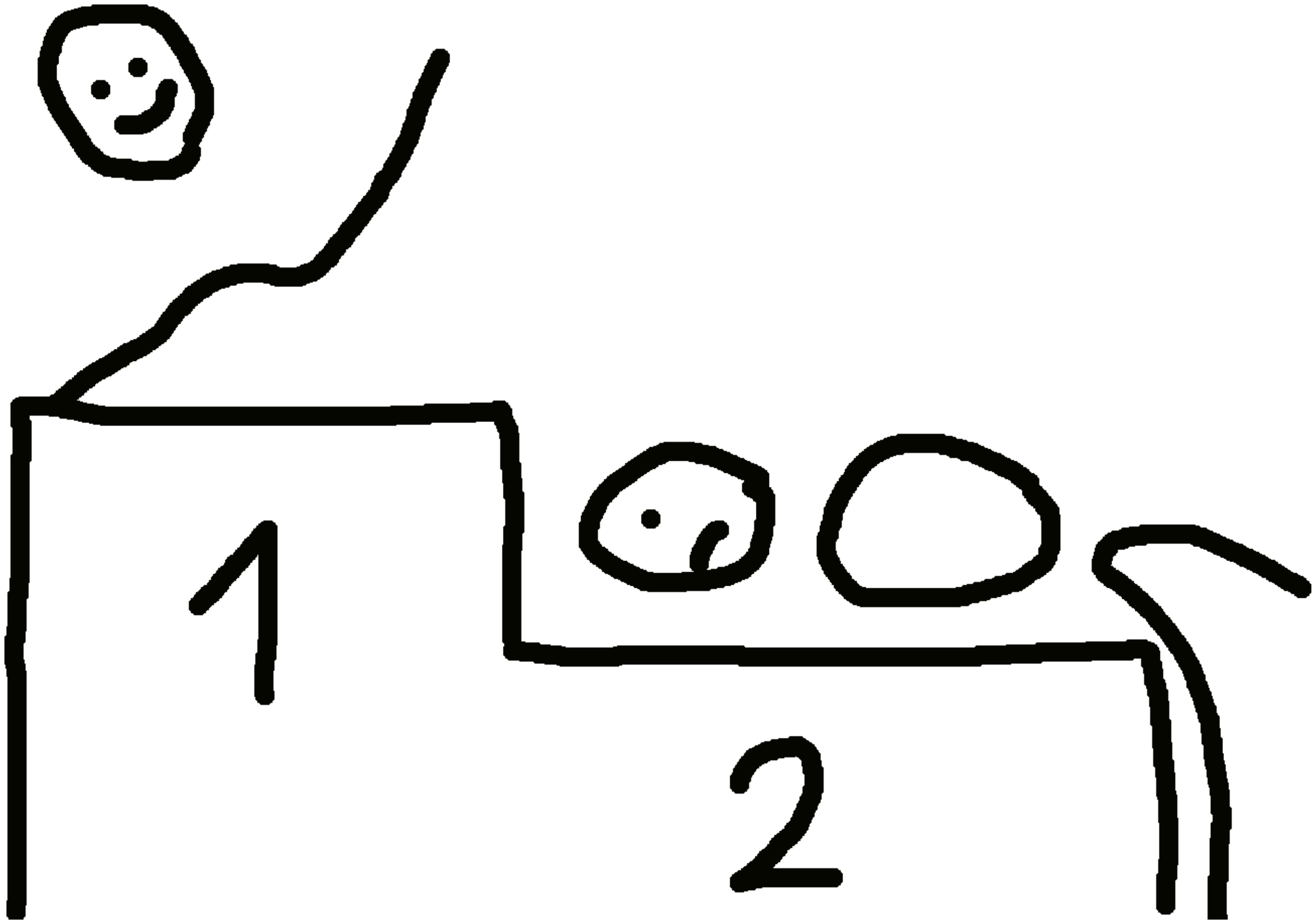
Elliptic curves vs. $g = 2$

- Let's look at the numbers first – sorry Dan!
- 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 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.
- My GMP and NTL implementation didn't look too promising for HEC ...
- ...which could be explained by too general software (one could hope).

Roberto Avanzi, CHES 2004

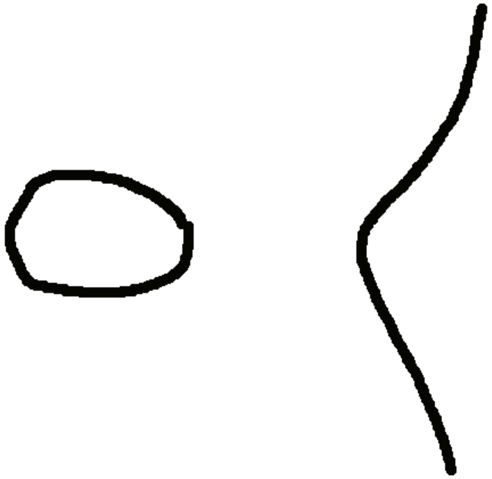
"We present an implementation of elliptic curves and of hyperelliptic curves of genus 2 and 3 over prime fields. To achieve a fair comparison between the different types of groups, we developed an ad-hoc arithmetic library, designed to remove most of the overheads that penalize implementations of curve-based cryptography over prime fields. These overheads get worse for smaller fields, and thus for larger genera for a fixed group size. We also use techniques for delaying modular reductions to reduce the amount of modular reductions in the formulae for the group operations. The result is that the performance of hyperelliptic curves of genus 2 over prime fields is much closer to the performance of elliptic curves than previously thought. For groups of 192 and 256 bits the difference is about 14% and 15% respectively.

End of Round 1



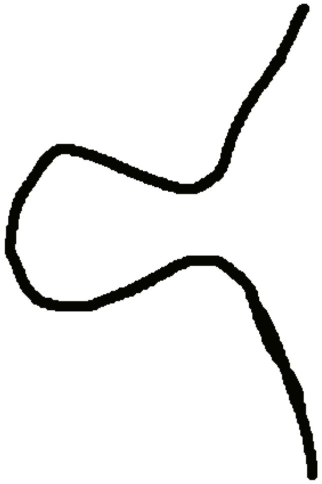
Round 2

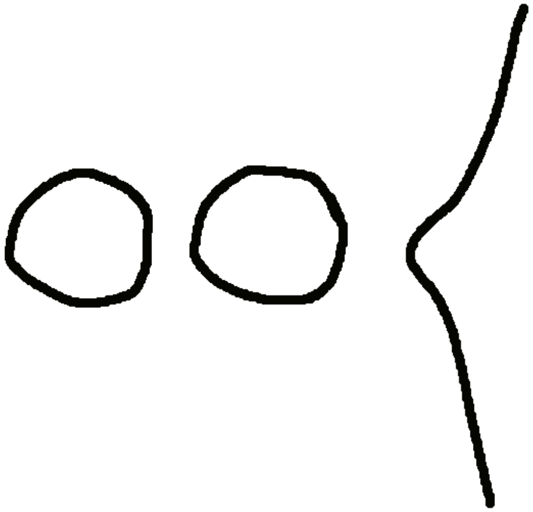
Let’s have a closer look at the opponents . . .

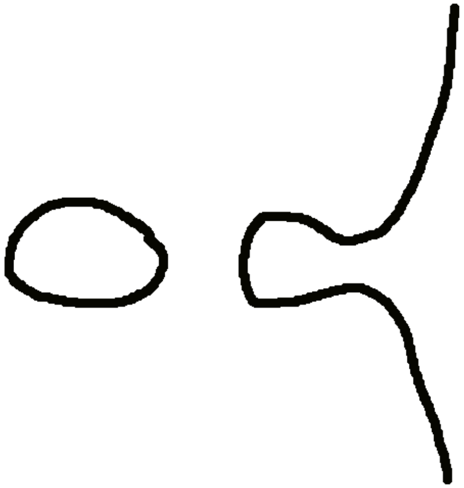


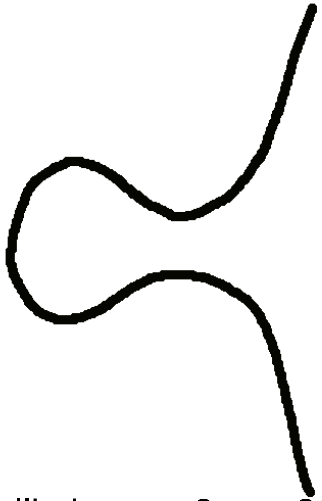
Three types for $g = 1$ (the one with no root cannot be drawn).

Four types for $g = 2$.









Zoom on the candidates

- Larger genus curves have more parameters and thus allow more choices.
- Hope to find efficient and secure families of curves.
- Special family of HEC might be faster than comparable family of EC – or there might be no similar family.
- Just seen such an example:
  Dan reported success for $g = 2$ on the

  Montgomery vs. Chudnovsky^2 & Gaudry

  part of the fight.
- Story for $g = 3$ still unwritten but one can hope for even larger speed-up in the Montgomery approach.

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

- Peter reported on genus 2 curves with $\deg(h) = 1$.
- This family of curves has 2-rank 1, is thus neither supersingular nor ordinary.
- For elliptic curves we do not have anything intermediate.
- Scalar multiplication much faster on genus-2 curves than on EC (time for precomputations is included, each with optimal window size).
- Avanzi, Thériault, and Wang, Rethinking Low Genus Hyperelliptic . . . , 2006. Gives study of field and explicit curve arithmetic for genera 3 and 4. Concentrates on special cases

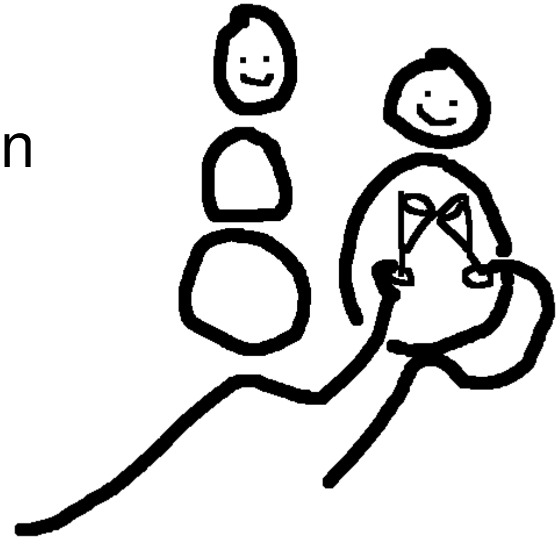
$$\begin{aligned} y^2 + y &= x^7 + f_5 x^5 \cdots \\ y^2 + y &= x^9 + f_7 x^7 \cdots, f_7 \neq 0. \end{aligned}$$

Avanzi, Thériault, and Wang

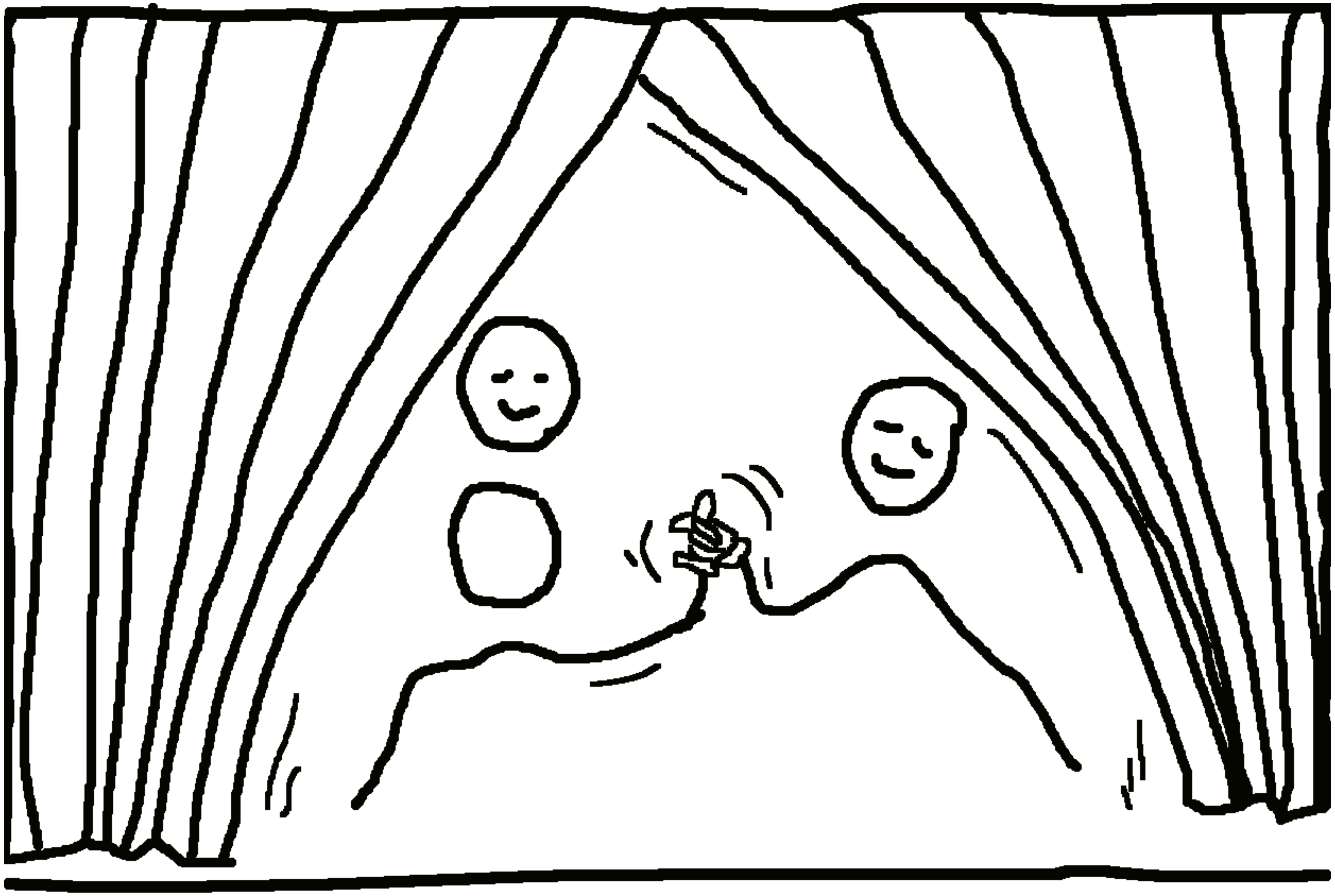
- Careful implementation of binary field arithmetic.
- Timings for $g = 1, 2, 3$, and 4, each with best coordinate system available.
- Performance of curves of genus four is comparable to that of EC often better.
- Curves of genus two and three similar performance.
- Genus three wins in some cases.
  Depends mainly on relation between
  finite field sizes in relation to
  word size.
- Genus $g = 4$ performance similar to elliptic curves.

Avanzi, Thériault, and Wang

- Careful implementation of binary field arithmetic.
- Timings for $g = 1, 2, 3$, and 4, each with best coordinate system available.
- Performance of curves of genus four is comparable to that of EC often better.
- Curves of genus two and three similar performance.
- Genus three wins in some cases. Depends mainly on relation between finite field sizes in relation to word size.
- Genus $g = 4$ performance similar to elliptic curves.



Special $g = 2$ families are very fast



Round 3

–

more special choices and applications

Special addition, $g = 2$

Addition, $\deg u_1 = 1, \deg u_2 = 2$

Input	$[u_1, v_1], [u_2, v_2], u_1 = x + u_{10}, u_2 = x^2 + u_{21}x + u_{20}, v_1 = v_{10}, v_2 = v_{21}x + v_{20}$
Output	$[u', v'] = [u_1, v_1] + [u_2, v_2]$

Step	Expression	Operations
1	compute $r \equiv u_2 \bmod u_1$: $r = u_{20} - (u_{21} - u_{10})u_{10}$;	M
2	compute inverse of u_2 modulo u_1: $inv = 1/r$	I
3	compute $s = (v_1 - v_2)inv \bmod u_1$: $s_0 = inv(v_{10} - v_{20} - v_{21}u_{10})$;	2M
4	compute $l = su_2 = s_0x^2 + l_1x + l_0$: $l_1 = s_0u_{21}, l_0 = s_0u_{20}$;	2M
5	compute $k = (f - v_2h - v_2^2)/u_2 = x^3 + k_2x^2 + k_1x + k_0$: $k_2 = f_4 - u_{21}, k_1 = f_3 - (f_4 - u_{21})u_{21} - v_{21}h_2 - u_{20}$;	M
6	compute $u' = (k - s(l + h + 2v_2))/u_1 = x^2 + u_1'x + u_0'$: $u_1' = k_2 - s_0^2 - s_0h_2 - u_{10}$; $u_0' = k_1 - s_0(l_1 + h_1 + 2v_{21}) - u_{10}u_1'$;	S, 2M
7	compute $v' \equiv -h - (l + v_2) \bmod u' = v_1'x + v_0'$: $v_1' = (h_2 + s_0)u_1' - (h_1 + l_1 + v_{21})$; $v_0' = (h_2 + s_0)u_0' - (h_0 + l_0 + v_{20})$;	2M
total		I, S, 10 M

Special base divisor

- Operations involving divisors of lower degree cheaper than general
- Result usually has full degree
- Use in scalar multiplication
  - Can be used if base divisor class has low degree representative
  - No use for special doublings but in mixed additions

Use of classes represented by

$$D = P - P_\infty, P \in C(\mathbb{F}_q)$$

was suggested by Katagi, Kitamura, Akishita, and Takagi for general DL systems.

Tate-Lichtenbaum pairing

$J_C(\mathbb{F}_q)[\ell]$: points on J_C of order ℓ defined over $\mathbb{F}_q$, ℓ is prime. Let k be minimal with $\ell \mid q^k - 1$.
Pairing is defined by

$$T_\ell : J_C(\mathbb{F}_{q^k})[\ell] \times J_C(\mathbb{F}_{q^k})/\ell J_C(\mathbb{F}_{q^k}) \to \mathbb{F}_{q^k}^*/\mathbb{F}_{q^k}^{*\ell}$$

via:
$\bar{D}_1 \in J_C(\mathbb{F}_{q^k})[\ell] \Rightarrow \exists F_{D_1}$ such that $\ell D_1 \sim \mathrm{div}(F_{D_1})$, where D_1 represents the class $\bar{D}_1$.
Let $\bar{D}_2 \in J_C(\mathbb{F}_{q^k})$ be represented by D_2 with

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

Then

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

Miller's algorithm for HECC

To evaluate use

$$F_{D_1}(D_2) = \frac{\prod_{i=1}^n F_{D_1}(P_i)}{\prod_{j=1}^n F_{D_1}(Q_j)}$$

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

Build F iteratively by Miller's algorithm (double-and-add): each operation results in polynomial of degree g, use Horner's rule to evaluate at the at most g points in the support of D_2. Do this for numerator and denominator.

Note:
$\bar{D}_2$ defined over $\mathbb{F}_{q^k} \Rightarrow$ individual points P_i, Q_j defined over some $\mathbb{F}_{q^{km}}$ for $m \geq 1$.

Tate-Lichtenbaum pairing II

Definition of Tate-Lichtenbaum pairing:

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

Idea:
use representatives of $J_C(\mathbb{F}_{q^k})/\ell J_C(\mathbb{F}_{q^k})$ that speed up computations, namely divisor classes with representatives of $\deg 1$
$\Rightarrow$ ensures $n = 1$, only 1 instead of g evaluations.

(This is used already in Duursma-Lee and subsequent papers.)

Problems & Solutions

- No idea how to find such a representative in classes of $J_C(\mathbb{F}_{q^k})/\ell J_C(\mathbb{F}_{q^k})$.
- No reason why there should exist one in each class.

Idea:

change protocols such that second argument is chosen in $C(\mathbb{F}_{q^k})$.

- How to ensure that pairing does not become trivial?
- How to apply this?
- Can we reach enough classes? Security issues?
- Frey, Lange, Fast bilinear maps from the Tate-Lichtenbaum pairing on hyperelliptic curves, 2006. Gives applications and proofs.

Parameter choices

Security level 2^{80}, assumed to correspond to 160-bit ECC and 1024-bit DL in finite fields.

genus	$g=1$	$g=2$	$g=3$	$g=4$
Pollard's rho	$q^{1/2}$	q	$q^{3/2}$	q^2
Double large prime	–	–	$q^{4/3}$	$q^{3/2}$
$\log_2 q$	160	80	60	54
k	6	13	17	20

- Speed-up grows with genus $\Rightarrow g=4$ might be interesting.
- Hard to construct MNT curves for $g=4$
- Arithmetic not well developed.

Wanted!

- Montgomery for $g = 3$.
- Family of hyperelliptic MNT curves.
- CM theory for $g = 4$.
- Security analysis of $g = 4$ for pairings.
  (Nigel's talk addressed the use in protocol, I mean the attacks on the cryptographic primitive.)
- Point counting for larger genus – genus 2 seems on the road, so far not much on imaginary quadratic hyperelliptic curves.
-
-

Wanted!

- Montgomery for $g = 3$.
- Family of hyperelliptic MNT curves.
- CM theory for $g = 4$.
- Security analysis of $g = 4$ for pairings.
  (Nigel's talk addressed the use in protocol, I mean the attacks on the cryptographic primitive.)
- Point counting for larger genus – genus 2 seems on the road, so far not much on imaginary quadratic hyperelliptic curves.
-
-
- ...coffee!

The end