Fast bilinear maps from the Tate-Lichtenbaum pairing on hyperelliptic curves

joint work with Gerhard Frey, Essen

Tanja Lange

Department of Mathematics

Technical University of Denmark

t.lange@mat.dtu.dk

28.07.2006

Overview

- Pairings in Cryptography
- Arithmetic on hyperelliptic curves
- Special divisors
- Tate-Lichtenbaum pairing
- Applications
- Justification

Pairings

Pairings

Let $(G_1, +), (G_1', +)$ and $(G, \cdot)$ be groups and let

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

be a 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 $G_1 = G_1'$.

Pairing based protocols

- Joux, ANTS 2000, one round tripartite key exchange
- Sakai-Ohgishi-Kasahara, in 2000, ID-based key exchange
- Boneh and Franklin, Crypto 2001, ID-based cryptography
- Boneh, Lynn, and Shacham, Asiacrypt 2001, short signatures

Hyperelliptic curves

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. not both partial derivatives

$$2y + h(x) \text{ and } h'(x)y - f'(x)$$

vanish in any in $(a, b) \in C/\overline{\mathbb{F}_q}$

Representation of group elements

Riemann-Roch allows to find a unique reduced representation by means of a divisor of degree zero with $n \leq g$

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

and $P_i \neq \iota(P_j)$ for $i \neq j$, where ι is the hyperelliptic involution.

If $\bar{D}$ is defined over $\mathbb{F}_q$, the extension degree of the field of definition of the P_l is bounded; e. g. for $g = 2$ each point is defined over $\mathbb{F}_{q^2}$.

Mumford Representation

Divisor class group is isomorphic to ideal class group $\mathbf{Cl}(C/\mathbb{F}_q)$ of $\mathbb{F}_q[x,y]/(C)$.
Elements $\bar{D}$ represented by two polynomials

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

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

Isomorphism is explicit via

$$P_i = (x_i, y_i) \Leftrightarrow u(x_i) = 0, v(x_i) = y_i \text{ with multiplicity}$$

Cantor's algorithm works with this representation. More efficient using explicit formulae. Dependence on $\deg(u)$ in running time.

General addition, $g = 2$

Addition, $\deg u_1 = \deg u_2 = 2$	
Input	$[u_1, v_1], [u_2, v_2], u_i = x^2 + u_{i1}x + u_{i0}, v_i = v_{i1}x + v_{i0}$
Output	$[u', v'] = [u_1, v_1] + [u_2, v_2]$

Step	Expression	Operations
1	compute resultant r of u_1, u_2: $z_1 = u_{11} - u_{21}, z_2 = u_{20} - u_{10}, z_3 = u_{11}z_1 + z_2$; $r = z_2z_3 + z_1^2u_{10}$;	1S, 3M
2	compute almost inverse of u_2 modulo u_1 $(inv = r/u_2 \bmod u_1)$: $inv_1 = z_1, inv_0 = z_3$;	
3	compute $s' = rs \equiv (v_1 - v_2)inv \bmod u_1$: $w_0 = v_{10} - v_{20}, w_1 = v_{11} - v_{21}, w_2 = inv_0w_0, w_3 = inv_1w_1$; $s_1' = (inv_0 + inv_1)(w_0 + w_1) - w_2 - w_3(1 + u_{11}), s_0' = w_2 - u_{10}w_3$; if $s_1' = 0$ see below	5M
4	compute $s'' = x + s_0/s_1 = x + s_0'/s_1'$ and s_1: $w_1 = (rs_1')^{-1}(= 1/r^2s_1), w_2 = rw_1(= 1/s_1'), w_3 = s_1'^2w_1(= s_1)$; $w_4 = rw_2(= 1/s_1), w_5 = w_4^2, s_0'' = s_0'w_2$;	I, 2S, 5M
5	compute $l' = s''u_2 = x^3 + l_2'x^2 + l_1'x + l_0'$: $l_2' = u_{21} + s_0'', l_1' = u_{21}s_0'' + u_{20}, l_0' = u_{20}s_0''$	2M
6	compute $u' = (s(l + h + 2v_2) - k)/u_1 = x^2 + u_1'x + u_0'$: $u_0' = (s_0'' - u_{11})(s_0'' - z_1 + h_2w_4) - u_{10} + l_1' + (h_1 + 2v_{21})w_4 + (2u_{21} + z_1 - f_4)w_5$; $u_1' = 2s_0'' - z_1 + h_2w_4 - w_5$;	3M
7	compute $v' \equiv -h - (l + v_2) \bmod u' = v_1'x + v_0'$: $w_1 = l_2' - u_1', w_2 = u_1'w_1 + u_0' - l_1', v_1' = w_2w_3 - v_{21} - h_1 + h_2u_1'$; $w_2 = u_0'w_1 - l_0', v_0' = w_2w_3 - v_{20} - h_0 + h_2u_0'$;	4M
total		I, 3S, 22M

Special divisors

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 $u' = (k - s(l + h + 2v_2))/u_1 = x^2 + u'_1x + u'_0$: $u'_1 = k_2 - s_0^2 - s_0h_2 - u_{10}$; $u'_0 = k_1 - s_0(l_1 + h_1 + 2v_{21}) - u_{10}u'_1$;	S, 2M
7	compute $v' \equiv -h - (l + v_2) \bmod u' = v'_1x + v'_0$: $v'_1 = (h_2 + s_0)u'_1 - (h_1 + l_1 + v_{21})$; $v'_0 = (h_2 + s_0)u'_0 - (h_0 + l_0 + v_{20})$;	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

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

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

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 lite 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 $m \geq 1$.

Usual setting

- The usual setting restricts the first argument to $J_C(\mathbb{F}_q)$.
- To achieve unique value in $\mathbb{F}_{q^k}$ rather than class do exponentiation

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

- Often second argument $J_C(\mathbb{F}_{q^k})/\ell J_C(\mathbb{F}_{q^k})$ replaced by $J_C(\mathbb{F}_{q^k})[\ell]$ using isomorphisms.

Improvements

- For $k > 1$ all computations involving elements from $\mathbb{F}_{q^m}$ with $m < k$ can be skipped (lead to ℓ-th powers)
  use standard representative of $\bar{D}_2 = \sum_{i=1}^{n} P_i - nP_\infty$ and ignore $P_\infty \Rightarrow$ avoid divisions in $T_\ell(\bar{D}_1, \bar{D}_2)$.
- If $k = 2m$ is even, restrict first argument $u(x)$ of $\bar{D}_2$ to $\mathbb{F}_{q^m}$ if protocol allows.
- Do the exponentiation

$$T'_\ell(\bar{D}_1, \bar{D}_2) = T_\ell(\bar{D}_1, \bar{D}_2)^{(q^k-1)/\ell},$$

  only at end of protocol.

(These ideas are well-known and not specific to non-supersingular curves.)

Tate-Lichtenbaum pairing II

Definition of Tate-Lichtenbaum pairing:

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

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?

Supersingular

Clearly, this approach works just the same for supersingular curves, some arguments require $k > 1$.
Why this title for the talk?

Attacks:

- Just choose second argument this way – works.
- In this case this is also interesting for $k = 1$.

Protocols:

- Often operation with second argument required (scalar multiplication).
- For $k > g$ more interesting to work over ground field $\mathbb{F}_q$ and apply distortion map.
- Short signatures should be in $J_C(\mathbb{F}_q)$ (otherwise not really short).

Non-supersingular I

Disadvantages compared to supersingular curves:

- No distortion maps (trace can be used to map from G_2 to G_1 but inversion is hard.
- No reason why there should be a multiple of ℓ with low Hamming weight.
- Still no good instances for $g > 1$.

Non-supersingular II

Advantages:

- Can use prime fields $q = p$ for large embedding degree k (supersingular requires small characteristic)
- $\Rightarrow$ smaller p for same resulting field size p^k
- $\Rightarrow$ faster operations in $J_C(\mathbb{F}_p)$
- $\Rightarrow$ need to minimize operations over larger field.
- Avoid particular fast index calculus due to Coppersmith.
- More choices (in theory).

Applications

ID-based cryptography I

Idea: sender decides public key of receiver (i.e. using his email-address).

Consequences

- Advantage if recipient is not in system or sender wants to force use of a fresh key (other applications possible).
- Set-up requires a trusted authority (TA) which can compute the secret key for a given public key (all kinds of trouble).

Let $H : \{0,1\}^* \rightarrow G_1'$ be hash function.

Master secret key of TA is s, public key is $P_{pub} = sP$.

Secret key of identity ID is $sH(ID)$, computable by TA.

ID-based cryptography II

Encryption:

- Choose ID string ID and compute $H(ID) \in G_1'$.
- Choose random nonce k and compute $R = kP$.
- Compute $c = (e(P_{pub}, H(ID)))^k \oplus m$ and send (R, c).

Decryption

- Obtain secret key $S' = sH(ID) \in G_1'$ from TA.
- Compute $c \oplus e(R, S') = m$.

$$e(R, S') = e(kP, sH(ID)) = (e(P, H(ID)))^{ks} = (e(sP, H(ID)))^k$$

$$\Rightarrow c \oplus e(R, S') = c \oplus (e(P_{pub}, H(ID)))^k = m$$

Applications - Examples

- ID-based encryption:
  - use hash function

$$H : \{0,1\}^* \to C(\mathbb{F}_{p^k})$$

    to map IDs to points (like in original BF).
  - Advantages for sender (and TA).
- Tripartite key exchange:
  Works, advantage only when computing scalar multiple.
- Can be used in most protocols.

Justification

α-exceptional

Let α be a fixed real number with $0 < \alpha < 1$. Let G be a finite abelian group of order n.

Let S be a subset of G and $\langle S \rangle$ the smallest subgroup of G containing S.
Then S is called α-exceptional (in G) if

$$\frac{|G|}{|\langle S \rangle|} > n^{\alpha}.$$

Lemma 1 *Let G' be a subgroup of G of order $< n^{1-\alpha}$, and assume that S is not α-exceptional.*
Then $S \not\subseteq G'$.

Results

Proposition 1 *G_0 cyclic group of prime order $\ell > n^\alpha$,
$b : G_0 \times G \to G_1$ $\mathbb{Z}$-bilinear map, non-degenerate in the first variable.*

If $S \subset G$ not α-exceptional then there exists $Q \in S$ such that for all $P \in G_0 \setminus \{e_{G_0}\}$

$$b(P, Q) \text{ has order } \ell.$$

- Use this concept on $C(\mathbb{F}_{q^k})$ to show that $\langle\{P - P_\infty | P \in C(\mathbb{F}_{q^k})\}\rangle$ is not contained in the kernel, i.e. does not only consist of ℓ-folds.
- Show that pairing with restricted second argument set is non-degenerate – the above is a necessary but not sufficient criterion.

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.

The end

The END

Thanks!