

A survey of hyperelliptic pairings

Steven Galbraith

Royal Holloway, University of London



<http://www.isg.rhul.ac.uk/~sdg/>

History

- ▶ The first papers on pairings in cryptography (Sakai-Ohgishi-Kasahara, Mitsunari-Sakai-Kasahara, Joux, Verheul, Boneh-Franklin) all used pairings on supersingular elliptic curves.
- ▶ I suggested using supersingular hyperelliptic curves, to get larger embedding degrees.
- ▶ We can ask: Was this a good suggestion?
- ▶ After 6 years of research, the answer is: Yes and No.
- ▶ Yes: Generated lots of interesting research and open problems.
- ▶ No: Hyperelliptic curves usually less practical for pairings than elliptic curves.

Executive summary

With current knowledge on pairing implementation, I recommend using elliptic curves for pairing-based cryptography.

Plan of talk

Joint work with Florian Hess and Frederik Vercauteren

- ▶ Brief introduction to hyperelliptic curves.
- ▶ Ate pairings and pairing implementation.
- ▶ Comparison between elliptic and hyperelliptic curves.
- ▶ Rubin-Silverberg compression.
- ▶ Torsion structure.
- ▶ Conclusions and open problems.
- ▶ If time: Pairing inversion.

Hyperelliptic curves

A **hyperelliptic curve** over a field \mathbb{F}_q is the curve associated with a non-singular equation of the form

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

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

Usually: $\deg(f(x)) = 2g + 1$ and $\deg(h(x)) \leq g$ in which case there is a single point ∞ and the curve has genus g .

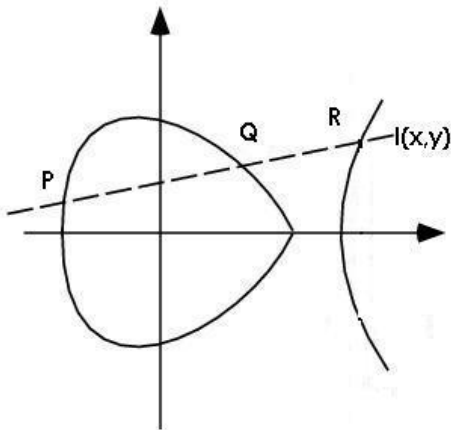
If the genus is 1 then we call the curve **elliptic**.

Example: $y^2 = x^5 + 1$ has genus 2.

Elliptic curve group law

For elliptic curves there is a group law on points given by a geometric procedure.

The rule is that $P + Q + R = 0$ if P , Q and R are the points of intersection (counting multiplicities) of the curve with a line.



Hyperelliptic group law

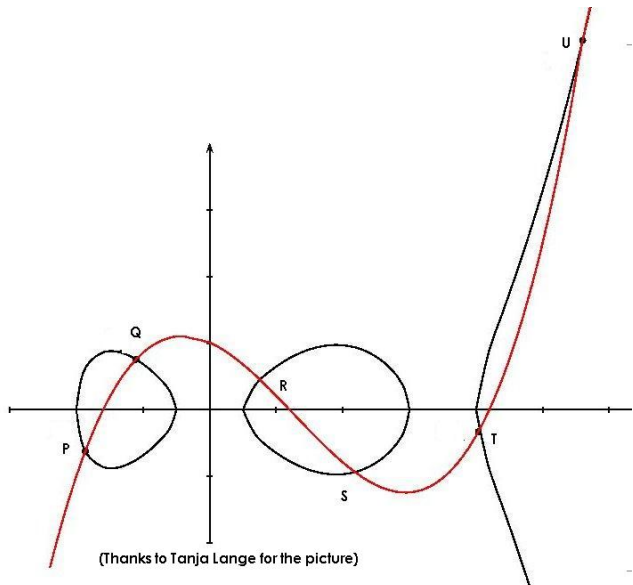
For genus $g \geq 2$ there is not a group law on points, but one can obtain a geometric group law on sets of points.

Let D_1, D_2, D_3 be (multi-)sets of points.

Then $D_1 + D_2 + D_3 = 0$ if there is curve F such that the intersection of C and F (counting multiplicities) is exactly the (multi-)set $D_1 \cup D_2 \cup D_3$.

Hyperelliptic group law

$$\{P, Q\} + \{R, S\} + \{T, U\} = 0$$



Divisor class group of a curve

The precise definitions use the language of divisors. See the *Handbook of Elliptic and Hyperelliptic Cryptography* for background.

Get divisor class group $\text{Pic}_{\mathbb{F}_q}^0(C)$.

In the case of genus 1, $\text{Pic}_{\mathbb{F}_q}^0(E) = E(\mathbb{F}_q)$.

Mumford representation for divisors

Each divisor class has a **reduced representative**

$$E - d(\infty)$$

where $d \leq g$ and $E = (P_1) + (P_2) + \cdots + (P_d)$ where $P_i = (x_i, y_i) \in C(\overline{\mathbb{F}}_q)$ and $P_i \neq -P_j$ for $i \neq j$.

Such a divisor is represented as a pair $u_E(x), v_E(x) \in \mathbb{F}_q[x]$ where

$$u_E(x) = \prod_{i=1}^d (x - x_i),$$

$$v_E(x_i) = y_i$$

and

$$v_E(x)^2 + h(x)v_E(x) - f(x) \equiv 0 \pmod{u_E(x)}.$$

Addition of divisors

Cantor's algorithm gives addition and reduction of divisors in Mumford representation.

See Algorithm 1 in the paper.

It is straightforward to obtain the functions needed for Miller's algorithm.

For fast implementations use optimised explicit formulae (Harley, Lange,...).

Group sizes

Theorem: If C is a curve of genus g then the divisor class group has order

$$(\sqrt{q} - 1)^{2g} \leq \#\mathrm{Pic}_{\mathbb{F}_q}^0(C) \leq (\sqrt{q} + 1)^{2g}.$$

If q is large compared with g then this is $\approx q^g$.

Advantages of hyperelliptic over elliptic

- ▶ Security of the DLP depends on the size of the largest prime divisor of the order of the group.
So for elliptic curves over \mathbb{F}_q need $q > 2^{160}$.
For genus 2 can have $q \approx 2^{80}$ and for genus 3 can have $q \approx 2^{54}$.
In other words, *have a more complicated group law but over a smaller field.*
- ▶ Use special curve equations to simplify explicit formulae for the group law (see Lange and Lange-Stevens).
- ▶ Use **degenerate divisors** $(P) - (\infty)$ rather than $(P_1) + (P_2) + \cdots + (P_g) - g(\infty)$ (Katagi-Akishita-Kitamura-Takagi).

Pairings on hyperelliptic curves

- ▶ T. Okamoto and K. Sakurai, CRYPTO 1991.
- ▶ G. Frey and H.-G. Rück, Math. Comp. 1994.

Pairings

I assume that everyone here knows something about pairings.

Let C be a curve over \mathbb{F}_q .

Let $r \mid \#\text{Pic}_{\mathbb{F}_q}^0(C)$.

Let k be smallest positive integer such that $r \mid (q^k - 1)$.
We call k the embedding degree.

Let $P, Q \in \text{Pic}_{\mathbb{F}_{q^k}}^0(C)$ have order r .

Let $G_1 = \langle P \rangle$ and $G_2 = \langle Q \rangle$.

Let $G_T = \mu_r = \{z \in \mathbb{F}_{q^k} : z^r = 1\}$.

For security need, say, $r > 2^{160}$ and $q^k > 2^{1024}$ (or 2^{2048} ?)

Tate-Lichtenbaum pairing

There is a function $f_{r,P}$ which has a zero of multiplicity r at P and pole of multiplicity r (or rd where d is the degree of the effective part of the divisor P) at ∞ and which is normalised appropriately at ∞ .

The **reduced Tate-Lichtenbaum pairing** is

$$e(P, Q) = f_{r,P}(Q)^{(q^k-1)/r}.$$

Lemma: If $r \mid N \mid (q^k - 1)$ then

$$e(P, Q) = f_{N,P}(Q)^{(q^k-1)/N}.$$

Developments in pairing implementation

- ▶ Duursma and Lee were the first to exploit values N which are not multiples of r .
These ideas were extended by Barreto, Galbraith, O hÉigeartaigh and Scott (eta pairing) and Hess, Smart, Vercauteren (ate pairing).
- ▶ One of the key ideas is to use values N which are smaller than the order of P .
This is called **loop shortening**.
- ▶ One gets fast pairing computation on certain curves.

Elliptic ate pairing

Allows loop shortening depending on the size of the trace of Frobenius t .

Suitable curves maybe generated using the Brezing-Weng method.

Other talks, such as Mike Scott's, will present examples of this.

Hyperelliptic ate pairing

$G_1 = \text{Pic}_{\mathbb{F}_q}^0(C)[r]$ (1-eigenspace of $\pi := q$ -power Frobenius)

$G_2 = \text{Pic}_{\mathbb{F}_{q^k}}^0(C)[r] \cap \ker(\pi - q)$ (q -eigenspace of π)

Theorem: (Granger, Hess, Oyono, Thériault, Vercauteren)

Let $D_1 = E_1 - d_1(\infty) \in G_1$ and $D_2 = E_2 - d_2(\infty) \in G_2$ be reduced divisors. (Assume supports of E_1 and E_2 are disjoint).

Let $[q]D_2$ be the reduced divisor equivalent to qD_2 . Denote by f_{q,D_2} the function with divisor $qD_2 - [q]D_2$ with leading coefficient 1 with respect to an \mathbb{F}_q -rational uniformizer at ∞ .

Then

$$a(D_2, D_1) = f_{q,D_2}(E_1)$$

defines a non-degenerate bilinear pairing

$$a : G_2 \times G_1 \rightarrow \mu_r.$$

Hyperelliptic ate pairing

- ▶ **Note:** No final exponentiation!
- ▶ But only if all denominators computed.
In practice, it is better to avoid computing denominators and use a final exponentiation.
- ▶ Hence, there seems to be no practical advantage to not needing the final exponentiation!
- ▶ Worse: lack of final exponentiation may make pairing inversion easier.

Implementation of the ate pairing

- ▶ As usual, one uses Miller's algorithm with various standard implementation tricks.
- ▶ Have to evaluate functions at divisor $E = (u_E(x), y - v_E(x))$. This can be done using resultants.
- ▶ A good trick is to replace y by $v_E(x)$ (so all polynomials depend on x only) and reduce modulo $u_E(x)$ (so all polynomials are of degree $\leq g$). Then do a single resultant at the end.
- ▶ There is a connection with norms in function fields.
- ▶ See the paper for more details.

Example (Barreto, G., Ó hÉigeartaigh, Scott)

(Using eta pairing)

Consider

$$C : y^2 + y = x^5 + x^3 + d \quad d \in \{0, 1\}$$

over \mathbb{F}_{2^m} with $\gcd(m, 6) = 1$.

Genus 2, embedding degree $k = 12$.

Pairing degenerate divisors using the eta pairing over $\mathbb{F}_{2^{103}}$ gives the fastest pairing computation time in software.

Example (Barreto, G., Ó hÉigearthaigh, Scott)

But:

- ▶ The implementation exploits 128-bit architecture.
- ▶ Degenerate divisors not necessarily secure for these parameters.
(e.g., if using $H(ID) = (P) - (\infty)$ then probability of hash collision only about $1/2^{52}$.)
Using non-degenerate divisors slower than elliptic case.

Comparing elliptic and hyperelliptic pairings

Consider

$$e : G_1 \times G_2 \rightarrow G_T.$$

Many criteria for comparison:

- ▶ Computation time for G_1 , G_2 , G_T and pairing computation.
- ▶ Size of representation for G_1 , G_2 , G_T .
- ▶ Flexibility and efficiency of parameter generation.
- ▶ Any other special properties.

See paper for full details.

Comparing elliptic and hyperelliptic pairings

Recall that one of the main advantages of hyperelliptic curves is that q can be smaller (i.e., *have a more complicated group law but over a smaller field*).

However, for pairings applications the field \mathbb{F}_{q^k} has to be large independent of the genus.

Hence, at least one of G_1, G_2 , *have a more complicated group law but over the same sized field*.

Similarly, in Miller's algorithm, evaluating more complicated functions at more complicated divisors over the same sized field. Hence, intuitively, pairings on curves of genus $g \geq 2$ cannot be faster than elliptic curves.

Comparison of loop shortening methods

In genus g the loop shortening is by a factor $1/g$.

For elliptic curves the loop shortening can be by a factor $\varphi(k)$.
(Note: also need k quite large, maximising $\varphi(k)$ alone is not desirable.)

Hence, one can usually match the loop shortening in the hyperelliptic case by using elliptic curves.

Embedding degrees

- ▶ The original motivation for hyperelliptic curves was larger k .
- ▶ Since group size is q^g and embedding is into (a subfield of) $\mathbb{F}_{q^k}^*$ the right measure of security expansion is k/g .
- ▶ There are some nice supersingular examples.
But supersingular curves have $k/g \leq 7.5$ for small genus g .
- ▶ Future security needs larger k/g , so use ordinary curves.
- ▶ In the elliptic case there has been great success with ordinary pairing-friendly curves (e.g., Cocks-Pinch, Barreto-Lynn-Scott, Brezing-Weng, Barreto-Naehrig, Freeman-Scott-Teske).
- ▶ Much less success for hyperelliptic curves (Freeman).
Indeed, no good non-supersingular example known, which is a pity.
Improving these methods is an interesting research problem.

Rubin-Silverberg compression

- ▶ Rubin and Silverberg proposed an alternative way to view pairings on abelian varieties.
- ▶ They observe that many supersingular abelian varieties can be identified with subvarieties of Weil restrictions of supersingular elliptic curves.
- ▶ An alternative way to view their method is as a form of point compression for elliptic curves.
- ▶ Their method has the effect of making an elliptic curve look like it has larger embedding degree.
- ▶ Their method works for all elliptic curves, not only supersingular curves.
- ▶ For details see the paper.

Rubin-Silverberg compression

For example, the supersingular genus 2 curve over \mathbb{F}_{2^m} with $k = 12$ has group order $2^{2m} \pm 2^{(3m+1)/2} + 2^m \pm 2^{(m+1)/2} + 1$.

A subgroup of the same order can be obtained using a supersingular elliptic curve over $\mathbb{F}_{2^{3m}}$ with $k = 4$.

The Rubin-Silverberg compression means that group elements require the same storage.

Hence one gets more-or-less the same functionality using elliptic curves with $k = 4$ as with the genus 2 curve with $k = 12$.

Torsion structures

- ▶ Some protocols in pairing-based cryptography (or even non-pairing cryptography, e.g., vector decomposition problem) use fact that elliptic curves give non-cyclic groups.
- ▶ Subgroup membership can be tested using the Weil pairing as, for $P, Q \neq \infty$ of prime order r we have

$$e_r(P, Q) = 1 \text{ iff } Q \in \langle P \rangle.$$

- ▶ In genus $g \geq 2$ then the torsion structures are even more rich. Are there new applications for this?

Torsion structures

Lemma: Let A be a supersingular abelian surface over \mathbb{F}_q with characteristic polynomial of Frobenius $T^4 + aT^2 + q^2$.

Let $r \parallel \#A(\mathbb{F}_q)$

Then the eigenvalues of Frobenius on $A[r]$ are $1, -1, q, -q$.

Let D_1, D_2, D_3, D_4 be an ordered eigenbasis for $A[r]$.

Let e be a Galois invariant non-degenerate pairing.

Then

$$e(D_i, D_j) = 1$$

unless $(i, j) = (1, 3), (3, 1), (2, 4), (4, 2)$.

Conclusions

- ▶ For non-pairing cryptography there are potential advantages of using hyperelliptic curves of genus $g \geq 2$.
It is thus natural to consider hyperelliptic curves for pairing-based cryptography.
- ▶ Our analysis indicates that, in practice, hyperelliptic curves are not more efficient than elliptic curves for general pairing applications.
- ▶ The only potentially significant advantage seems to be the speed of operations in G_1 .
Hence, hyperelliptic curves may be preferable for protocols with few pairing computations but many operations in G_1 .
- ▶ But, hyperelliptic pairings still a good research area!

Open problems

- ▶ Can further loop shortening be performed for the hyperelliptic Tate pairing?
- ▶ Give methods to construct non-supersingular pairing friendly curves of genus $g \geq 2$ and k in the range, say, $6g \leq k \leq 30g$. Ideally, these curves would have a single point at infinity and would have useful twists.
- ▶ Work in progress of my student Dave Mireles gives fast methods to compute pairings on hyperelliptic curves with two points at infinity.
- ▶ Can also consider implementation of pairings for non-hyperelliptic curves, although security is less good due to Diem's index calculus method.
- ▶ Consider whether efficient and secure pairing-based cryptosystems can be developed for curves of genus $g > 3$, in spite of the index calculus attacks on curves in this case.

Open problems

- ▶ Exploit the richer torsion structure available for abelian varieties.

In particular, find cryptographic applications of pairings on groups which require 3 or more generators.

A related problem is to give efficient methods to choose divisors in the particular subgroups.

- ▶ Improve the efficiency of the Rubin-Silverberg elliptic curve point decompression method.
- ▶ Generalise the Rubin-Silverberg method to divisor class groups of curves of genus $g \geq 2$.
- ▶ Recall in the Rubin-Silverberg construction one can identify certain abelian varieties with subvarieties of the Weil restriction of supersingular curves.

In the case where the abelian variety is a Jacobian, is there a way to compute explicit homomorphisms between the elliptic curve representation and the Jacobian representation?

Pairing inversion

Pairing-based cryptography relies on new computational problems.

It is important that these problems are studied, to give assurance that pairing-based cryptography is secure.

See “Aspects of pairing inversion” cryptography eprint 2007/256.

Also see talk by Satoh.