Divisors, Bilinear Pairings and Pairing Enabled Cryptographic Applications

Wenbo Mao, Principal Engineer
HP Labs., Bristol
(Joint work with K. Harrison, HP Labs., Bristol)

Coverage

— Pairings in an abstract level of description

— Cryptanalysis and cryptographic applications of pairings

— Divisors: building blocks of pairings

— Pairings: we will construct them and prove important properties (e.g., bilinearity)

— Efficient computation of pairings (Miller's algorithm)

Pairings in an abstract level of description

Let $\mathbb{F}_q$ be a finite field, for ease of description, we confine to the case $\mathrm{Char}(\mathbb{F}_q) > 3$

Let $E : y^2 - (x^3 + Ax + B) = 0$ be an elliptic curve over $\mathbb{F}_q$

$$E(\mathbb{F}_q) = \{\, (X, Y) \,|\, X, Y \in \mathbb{F}_q \text{ solved from } E \,\} \cup \{\mathcal{O}\}$$

here $\mathcal{O}$ is "the point at infinity"; these points form an additive group with $\mathcal{O}$ being the group identity

Let n be a prime satisfying

$n | \# E(\mathbb{F}_q)$
$n \not| \; q - 1$
n and q co-prime

Then for some integer k, $E(\mathbb{F}_{q^k})$ (note the field extension) contains n^2 points of order n if and only if $n \,|\, q^k - 1$

Let $E[n]$ denote the set of these n^2 order-n points:

$$\forall P \in E[n] : nP = \mathcal{O}$$

Bilinear and Non-degenerate Pairings

The Weil (pronounce vay) pairing:

$$e_n : E[n] \times E[n] \mapsto \mu_n$$

where μ_n is the (multiplicative) group of n-th roots of unity in $\mathbb{F}_{q^k}$, i.e., $\forall a \in \mu_n$: $a^n = 1$ (i.e., points on the unit circle)

Clearly, μ_n with n elements is the unique subgroup of $\mathbb{F}_{q^k}$ (since $\mathbb{F}_{q^k}$ is cyclic); however $E[n]$ has n^2 points; therefore the mapping is many-to-one

Important Properties

For $P, Q, R \in E[n]$

$e_n(P, P) = 1$ — identity

However, it's easy to make a "distortion" modification to obtain $\widehat{e}_n(\cdot, \cdot)$ such that $\widehat{e}_n(P, P) \neq 1$

$e_n(P + Q, R) = e_n(P, R)e_n(Q, R)$
$e_n(R, P + Q) = e_n(R, P)e_n(R, Q)$ — bilinearity

$e_n(P, Q) \neq 1$ for some $P, Q \in E[n]$ — non-degeneracy

A Cryptanalysis Application (Menezes-Okamoto-Vanstone 1993)

For E being a supersingular curve, the necessary field extension can be a small one: $k \leq 6$, and it's easily to make $k = 2$, i.e., $\mu \subseteq \mathbb{F}_{q^2}$

Let P, aP be a discrete logarithm problem on this curve group, which was believed to be a problem of cost $O(\sqrt{q})$

Let $Q \in E[n]$ such that $e_n(P, Q) \neq 1$ (non-degeneracy)

Then (by bilinearity)

$$e_n(P, Q), \;\; e_n(aP, Q) = e_n(P, Q)^a$$

is a discrete logarithm problem in $\mathbb{F}_{q^2}$, it has a subexponential solver with cost sub_exp(2$|q|$) ($\ll \sqrt{q} = \exp(|q|/2)$)

This is the "MOV Attack"; it suggests not to use supersingular curves for cryptographic applications

but from year 2000 on ...

From Year 2000 On ...

Sakai, Ohgishi and Kasahara (2000) pioneered "non-interactive key-sharing: user X has a public key P_X and a private key $S_X = \ell P_X$, then Alice and Bob share an exclusive secret key even they have never talked to each other:

$$K_{AB} = \hat{e}_n(S_A, P_B) = \hat{e}_n(\ell P_A, P_B) = \hat{e}_n(P_A, P_B)^\ell$$
$$= \hat{e}_n(P_A, \ell P_B) = \hat{e}_n(P_A, S_B) = K_{BA}$$

Independently, Joux (2000) pioneered "tripartite Diffie-Hellman key agreement": user X broadcasts P_X, then

$$\hat{e}_n(P_B, P_C)^a = \hat{e}_n(P_A, P_C)^b = \hat{e}_n(P_A, P_B)^c = \hat{e}_n(P, P)^{abc} = K_{ABC}$$

Boneh and Franklin (2001): an identity-based cryptosystem (public key P_X can be an identity or anything with a distribution pleasant to human, in contrast, a conventional public key is at least pseudo-random)

... ...

These are very interesting cryptographic applications enabled by pairings, of course, using supersingular curves in order to achieve practical efficiency (recall a small field extension)

Boneh-Franklin ID-Based Encryption

Let P be a public point
Let Ppub = sP be also public
The pair (P, Ppub) is the public key of TA

Let A be Alice's ID
Let sA be Alice's private key

Let H( ) be a hash function

Encryption:
Bob picks random r, computes U = rP, V = H(e(rA, Ppub)) xor M
(U, V) is the ciphertext

Decryption:
Alice computes: H(e(xA, U)) xor V

Notice: e(rA, Ppub) = e(rA, xP) = e(A, P) = e(xA, rP) = e(xA, U)
Therefore Decryption indeed returns M

Decisional Diffie-Hellman Problem — Gone!

Decisional Diffie-Hellman Problem

Input: (P, aP, bP, cP)

Output: Yes if $c = ab$

This is a hard problem in general groups

Not hard anymore in groups of (supersingular) elliptic curves

Let $\widehat{e}(\cdot, \cdot)$ be a pairing satisfying $\widehat{e}(X, X) \neq 1$, then

$$\widehat{e}_n(P, cP) = \widehat{e}_n(P, P)^c = \widehat{e}_n(P, P)^{ab} = \widehat{e}_n(aP, bP)$$

if and only if $ab = c$

Now let's investigate how these magics happen

Divisors: Building Blocks of Pairings

In order to know the interesting properties of pairings (e.g., bilinearity) and how to compute pairings, let's study divisors

A divisor is a "formal" sum:

$$D = \sum_{P \in E} a_P [P]$$

here a_P is an integer, $[P]$ is a "formal" symbol

Please do not be scared by the word "formal!" Think the quote $[\cdot]$ being an artificial way to prevent D from becoming a point (so $a[P] + b[Q]$ is a divisor while $aP + bQ$ is a point)

How Large is the Sum?

Divisors of our interest will always be a small sum. This is because for most P on E we will have $a_P = 0$ even if the number of points on E can be intractably large or even infinite

Degree of a Divisor

$$\deg(D) = \sum_{P \in E} a_P$$

For divisors of our interest, $\deg(D) = 0$; so a_P must be positive for some P's and negative for some other P's. In fact, we shall see (in Miller's algorithm) that even if a D has a non-zero degree, we will modify it into D' so that $\deg(D') = 0$

A divisor D with $\deg(D) = 0$ is called a *principal divisor*, it is related to a function

Functions on an Elliptic Curve

Function $f(x,y)$ on $E(x,y)$ means all points (x,y) solved from the equation $f = E$, i.e., (x,y) on $f \cap E$

Examples

1) For $f : x = 0$ (the y axis), if B is a square in the field, then $(0, \sqrt{B})$, $(0, -\sqrt{B})$ and $\mathcal{O}$ are the three points on $f \cap E$

2) $E : y^2 = x^3 - x \quad f : x/y$; We know $(0,0)$ is a finite point on E; how about this point on f? (i.e., does $0/0$ make sense?)

   Because $f = E$ gives $x/y = y/(x^2 - 1)$, so at point $(0,0)$, $x/y = 0/(0^2 - 1) = 0$ makes a perfect sense

3) Now consider x/y meeting E at $\mathcal{O}$. Does $\frac{\infty}{\infty}$ make any sense?

   Because $f = E$ means $\dfrac{x}{y} = \dfrac{1}{\sqrt{x(1 + A/x^2 + B/x^3)}}$

   so x/y takes 0 when $x = \infty$ (regardless of y); hence, at the point of infinity $\mathcal{O}$, we have $x/y = \infty/\infty = 0$

Zeros and Poles

For P on $F \cap E$, P is called a $\begin{cases} \text{zero} & \text{if } f(P) = 0 \\ \text{pole} & \text{if } f(P) = \infty \end{cases}$

Factorisations of Zeros and Poles

In a finite field, 0 can be factored into $0^i \cdot g(P)$ s.t. i is a positive integer and $g(P) \neq 0$, $g(P) \neq \infty$

> if $g(P) = 0$ then increase i until $g(P) \neq 0$; if $g(P) = \infty$ then decrease i until $g(P) \neq \infty$

Viewing ∞ as "0^{-1}", we can analogously factor a pole into $0^{-i} \cdot g(P)$ for $g(P) \neq 0$, $g(P) \neq \infty$

Let ord_P denote i or $-i$ when $f(P)$, as a zero or pole, is factored in the above manners; ord_P shows how "strong" a zero (pole) is

Remember

> $\text{ord}_P > 0$ if P a zero
>
> $\text{ord}_P < 0$ if P is a pole
>
> $\text{ord}_P = 0$ if P is a not a zero or pole

Facts of Orders of Important Zeros

Zeros of linear functions are important

Let $\ell : y = ux + v$ be a line ($u \neq 0$). A zero $P = (x_0, y_0)$ of ℓ is a finite solution solved from

$$\ell \cap E: \ (ux+v)^2 = y^2 = x^3 + Ax + B$$

i.e., x_0 is a root of

$$(ux+v)^2 - (x^3 + Ax + B) = 0$$

This 0 can be factored into

$$(x - x_0)^d \cdot g \text{ with } g(x_0) \neq 0 \text{ and } d = \begin{cases} 2 & \text{if } P \text{ is a tangent point} \\ 1 & \text{otherwise} \end{cases}$$

Therefore

$$\mathrm{ord}_P(\ell) = \begin{cases} 1 & \text{if } \ell \text{ cuts } E \text{ at } P \\ 2 & \text{if } \ell \text{ is tangent to } E \text{ at } P \end{cases}$$

(Question: how many point satisfying $d = 3$?)

The same result holds for the special case $\ell : x = c$ (a vertical line)

Facts of Orders of Important Poles

Poles of linear functions are also important; let's first investigate $\mathrm{ord}_{\mathcal{O}}(x)$ and $\mathrm{ord}_{\mathcal{O}}(y)$

Writing $\left(\frac{x}{y}\right)^2 = \frac{x^2}{x^3\cdot(1+\cdots)}$, we have $x = \left(\frac{x}{y}\right)^{-2} \cdot \frac{1}{(1+\cdots)}$

Recall Example (3) in Slide 9, we know $\frac{x}{y} = 0$ at $\mathcal{O}$; also notice $\frac{1}{(1+\cdots)} = 1$ at $\mathcal{O}$; therefore

$$\mathrm{ord}_{\mathcal{O}}(x) = -2$$

Now because $y = \left(\frac{x}{y}\right)^{-1} \cdot x = \left(\frac{x}{y}\right)^{-3} \cdot \frac{1}{(1+\cdots)}$, we have

$$\mathrm{ord}_{\mathcal{O}}(y) = -3$$

Summary Let a linear function ℓ be $ux + vy + w = 0$, then

$$\mathrm{ord}_{\mathcal{O}}(\ell) = \begin{cases} -3 & \text{if } v \neq 0 \\ -2 & \text{otherwise} \end{cases}$$

Divisor of a Function

Let $f \neq 0$ be a function on E, the divisor of f is

$$\mathrm{div}(f) = \sum_{P \in E} \mathrm{ord}_P(f)[P]$$

Divisors of linear functions are important ones

For a linear function $\ell : ux + vy + w = 0$ $(u, v \neq 0)$, we know ℓ joins E at exactly three finite points P_1, P_2, P_3 (two of them may coincide, i.e., ℓ is tangent to E at the point), each of these points is a single zero of ℓ (the tangent point is a double zero)

In addition, we have also seen that ℓ has a pole at $\mathcal{O}$, and since $v \neq 0$, the pole is a triple one; thus, we have

$$\mathrm{div}(\ell) = [P_1] + [P_2] + [P_3] - 3[\mathcal{O}]$$

Divisor of a Function (II)

Let ℓ' be a vertical line ($v = 0$) cut E at P_3 and $-P_3$; they are single zeros (unless $y = 0$, a tangent case of a double zero); ℓ' also has a double pole at $\mathcal{O}$, hence

$$\mathrm{div}(\ell') = [P_3] + [-P_3] - 2[\mathcal{O}]$$

With $P_3 = -(P_1 + P_2)$, $\mathrm{div}(\ell) - \mathrm{div}(\ell')$ becomes

$$\mathrm{div}\left(\frac{\ell}{\ell'}\right) = \mathrm{div}(\ell) - \mathrm{div}(\ell') = [P_1] + [P_2] - [P_1 + P_2] - [\mathcal{O}] \qquad (1)$$

Why does the first equation of (1) hold?

Consider how the following "poem" contributes positive/negative signs to the co-efficients in the right-hand side of (1):

A zero of ℓ is a zero of ℓ/ℓ'
a zero of ℓ' is a pole of ℓ/ℓ'
a pole of ℓ' is a zero of ℓ/ℓ'
a pole of ℓ is a pole of ℓ/ℓ'

A mathematician is akin to a poet, both are serious artists :-)

Pairing Construction by Repeated Addition of Divisors

Let $P \in E[n]$, i.e., $nP = \mathcal{O}$; let $P = P_1 = P_2$, then (1) becomes

$$\text{div}(f_1) = 2[P] - [2P] - [\mathcal{O}]$$

for some function f_1

Keeping on adding P and applying (1), we can derive

$$\begin{aligned}
&\text{div}(f_2) = 3[P] - [3P] - 2[\mathcal{O}] \\
&\text{div}(f_3) = 4[P] - [4P] - 3[\mathcal{O}] \\
&\ldots \\
&\text{div}(f_{n-1}) = n[P] - [nP] - (n-1)[\mathcal{O}]
\end{aligned}$$

Since $nP = \mathcal{O}$, the final equation, $\text{div}(f_{n-1})$, becomes

$$\text{div}(f_P) = n[P] - n[\mathcal{O}] \tag{2}$$

for some function f_P on E (we have renamed f_{n-1} into f_P)

What have we done?
We have used $P \in E[n]$ to construct a function f_P satisfying (2)

The Tate Pairing Constructed Using f_P

Let Q, S be any curve points, define un-named "pairing":

$$\text{un}_n(P,Q)_S = \frac{f_P(Q+S)}{f_P(S)} \tag{3}$$

this value depends not only on P, Q, but also on random S, and so should be more precisely called *triplet* than "pairing"

However, for $\text{un}_n(P,Q)_S$ and $\text{un}_n(P,Q)_{S'}$ constructed using $S \neq S'$, applying "Weil reciprocity" it can be shown

$$\frac{\text{un}_n(P,Q)_S}{\text{un}_n(P,Q)_{S'}} = \xi^n \quad \text{for some } \xi \in \mathbb{F}_{q^k}$$

Then by Fermat's theorem we have

$$\left(\frac{\text{un}_n(P,Q)_S}{\text{un}_n(P,Q)_{S'}}\right)^{(q^k-1)/n} = \xi^{q^k-1} = 1$$

i.e.,

$$(\text{un}_n(P,Q)_S)^{(q^k-1)/n} = (\text{un}_n(P,Q)_{S'})^{(q^k-1)/n}$$

is independent from any random points S, S'; so define

$$t_n(P,Q) = (\text{un}_n(P,Q)_S)^{(q^k-1)/n}$$

$t_n(P,Q)$ is indeed a pairing of P, Q and is in fact the Tate pairing

The Weil Pairing

In the case of the Weil pairing, $P, Q \in E[n]$; the Weil pairing can be defined from the Tate pairing:

$$e_n(P,Q) = \frac{t_n(P,Q)}{t_n(Q,P)}$$

The Type of these Pairings

We have seen:

$$f_P = \frac{\ell_1 \cdot \ell_2 \cdots \ell_s}{\ell'_1 \cdot \ell'_2 \cdots \ell'_t}$$

where ℓ_i, ℓ'_j are all linear functions of $(x, y) \in \mathbb{F}_{q^k}$, therefore

$$\mathsf{un}_n(P,Q)_S,\ t_n(P,Q),\ e_n(P,Q) \in \mathbb{F}_{q^k}$$

Moreover, since $t_n(P,Q)^n = 1$, we further have

$$t_n(P,Q) \in \mu_n \qquad e_n(P,Q) \in \mu_n$$

i.e., they are on the unit circle

Why field extension?
Recall $n \not| \, q-1$ (Slide 2); there is no order-n elements in $\mathbb{F}_q$, so P or Q must have coordinates in the extended field

Bilinearity — Case 1

Applying (3):

$\mathrm{un}_n(P, Q_1)_S \cdot \mathrm{un}_n(P, Q_2)_{Q_1+S}$

$$= \frac{f_P(Q_1 + S)}{f_P(S)} \cdot \frac{f_P(Q_2 + (Q_1 + S))}{f_P(Q_1 + S)}$$

$$= \frac{f_P((Q_1 + Q_2) + S)}{f_P(S)}$$

$$= \mathrm{un}_n(P, Q_1 + Q_2)_S$$

But the Tate and Weil pairings are independent from S, $S + Q_1$, therefore

$$t_n(P, Q_1) \cdot t_n(P, Q_2) = t_n(P, Q_1 + Q_2)$$

$$e_n(P, Q_1) \cdot e_n(P, Q_2) = e_n(P, Q_1 + Q_2)$$

Bilinearity — Case 2

Let $P_1, P_2, P_3 \in E[n]$ with $P_1 + P_2 = P_3$. Let g be a function satisfying (1), i.e.,

$$[P_3] - [\mathcal{O}] = ([P_1] - [\mathcal{O}]) + ([P_2] - [\mathcal{O}]) + \mathrm{div}(g) \tag{4}$$

Let further f_i $(i = 1, 2, 3)$ be functions constructed in (2), i.e.,

$$\mathrm{div}(f_i) = n[P_i] - n[\mathcal{O}]$$

Multiplying n to both sides of (4) and recall the "poem" in Slide 14, we derive

$$\mathrm{div}(f_3) = \mathrm{div}(f_1 f_2 g^n)$$

So for some constant c, we have $f_3(X) = c f_1(X) f_2(X) g^n(X)$

Applying (3), we have

$$\mathrm{un}_n(P_1 + P_2, Q)_S = \mathrm{un}_n(P_3, Q)_S = \frac{f_3(Q+S)}{f_3(S)}$$

$$= \frac{c}{c} \cdot \frac{f_1(Q+S)}{f_1(S)} \cdot \frac{f_2(Q+S)}{f_2(S)} \cdot g^n(X)$$

$$= \mathrm{un}_n(P_1, Q)_S \cdot \mathrm{un}_n(P_2, Q)_S \cdot g^n(X)$$

By disregarding the difference in n-th power, we have

$$t_n(P_1 + P_2, Q) = t_n(P_1, Q) \cdot t_n(P_2, Q)$$

Anologous for the Weil pairing

Pairing Computation

Now we know: to compute a pairing $\text{un}_n(P, Q)$ we need to find f_P satisfying

$$\text{div}(f_P) = n[P] - n[\mathcal{O}]$$

or more generally, satisfying

$$\text{div}(f_P) = n[P + R] - n[R]$$

($R = \mathcal{O}$ is a special case)

The constructive method we've just built ("keep adding") costs $O(n)$, infeasible for large n (large enough for cryptographic applications)

We have done a mathematician's construction of bilinear pairings (and proof of bilinearity); the cost is $O(n)$, too high, only suitable for people having an ivory-tower level of luxury!

We are not only mathematicians, but also computer scientists, we work for a computer company!

Miller's Magic (for People outside of the Ivory Tower!)

The following "Miller divisor" D_k has degree 0 for any integer k

$$MD_k = k[P+R] - k[R] - [kP] + [O]$$

So there exists f_k satisfying (for any k)

$$\mathrm{div}(f_k) = k[P+R] - k[R] - [kP] + [O]$$

When $k = n$, $\mathrm{div}(f_n) = MD_n = n[P+R] - n[R]$, so $f_n = f_P$ is what we want

Write

$$n = b_i 2^i + b_{i-1} 2^{i-1} + ... + b_0 \text{ with } b_j \in \{0, 1\}$$

Construct f_{2^j} for all $b_j \neq 0$ via "point doubling", then construct f_n using these f_{2^j} via "point adding", and we are done!

A "point doubling" algorithm will do the job quickly

Point Doubling

Given P, let ℓ be the tangent line through P; then the tangent form of (1) is

$$\text{div}(f_2) = 2[P] - [2P] - [\mathcal{O}]$$

Repeating the doubling, f_{2^j} can be constructed in j steps

Finally, since $n = \sum b_i 2^i$, the time for constructing $f_n = f_P$ using the "doubling-and-adding" algorithm is

$$O(\log n \cdot (\log q)^2)$$

where $(\log q)^2$ is the cost for point doubling or addition

By choosing the prime n with low Hamming weight, this cost is comparable to that of RSA using small public exponent e

Cool!

知己知彼
百战不殆