Sage days 10, Nancy, France

Implementing the Weil, Tate and Ate pairings using Sage software

Nadia EL MRABET
LIRMM, I3M,
Université Montpellier 2

Saturday 11th October 2008

Outline of the presentation

1. Definition of a pairing

2. Construction of a pairing

3. Implementation of a pairing

What is a pairing ?

Properties

Let G_1, G_2 and G_3 be three groups with the same order r. A pairing is a map :

$$e : G_1 \times G_2 \longrightarrow G_3$$

which verifies the following properties :

- *Non degenerate ;*
- ◇ $\forall P \in G_1 \ \{0\} \exists Q \in G_2 / e(P, Q) \neq 1$
- ◇ $\forall Q \in G_2 \ \{0\} \exists P \in G_1 / e(P, Q) \neq 1$
- *Bilinearity :* $\forall P, P' \in G_1, \forall Q, Q' \in G_2$
- ◇ $e(P + P', Q) = e(P, Q).e(P', Q)$
- ◇ $e(P, Q + Q') = e(P, Q).e(P, Q')$

What is a pairing ?

Properties

Let G_1, G_2 and G_3 be three groups with the same order r. A pairing is a map :

$$e : G_1 \times G_2 \longrightarrow G_3$$

which verifies the following properties :

- *Non degenerate ;*
- *Bilinearity ;*

Consequence

$$\forall j \in \mathbb{N}, e([j]P, Q) = e(P, Q)^j = e(P, [j]Q)$$

The MOV/Frey Rück attack against the DLP on elliptic curves in 1993, 1994 :
using pairings, the DLP on elliptic curves becomes a DLP on finite field.

- Given P and $Q = \alpha P \in E(\mathbb{F}_q)$,
  the DLP on $E(\mathbb{F}_q)$ consists in finding α.
- Let $S \in E(\mathbb{F}_q)$ be a point such that $e(P, S) \neq 1$,
  let $e(P, S) = g$ and $e(Q, S) = h \in E(\mathbb{F}_q)$, then
- the DLP becomes finding α such that $h = g^\alpha$ in a finite field.

Pairings allow the construction of novel protocols and simplification of existing protocols.

- The tri partite Diffie Hellman key exchange protocol (Joux 2001)
- The Identity Based Encryption (Boneh and Franklin 2001)
- Short signature scheme (Boneh, Lynn, Schackamm 2001)
- Group signatures schemes (Boneh, Schackamm, 2004)

Four pairings are principally used in cryptography :

- the Weil pairing,
- the Tate pairing,
- the η_T pairing,
- the Ate pairing.

I focused only on the pairings constructed by the same way. The Miller algorithm constructing the function $f_{r,P}$ is a central step for the Weil, Tate and Ate pairings.

Construction of the pairings

Data

To compute a pairing, we need the following elements :

- E an elliptic curve over $\mathbb{F}_q$:
  $E : y^2 = x^3 + ax + b$, where a, $b \in \mathbb{F}_q$.
- r a prime dividing card$(E(\mathbb{F}_q))$,
  consider $E[r]$: $E[r] = \{P \in E(\overline{\mathbb{F}_q}), [r]P = P_\infty\}$.
- The embedding degree k : minimal integer such that $r|(q^k - 1)$:
  If $gcd(r, q) = 1$, then $E[r] \cong \mathbb{Z}/r\mathbb{Z} \times \mathbb{Z}/r\mathbb{Z}$,
  If $k > 1$ then $E[r] = E(\mathbb{F}_{q^k})[r]$.
- A function $f_{r,P}$ described lately.

Construction of the pairings

The Weil pairing

Let $P \in E[r]$ and $Q \in E[r]$

The Weil pairing is the bilinear map :

$$e_W : E[r] \times E[r] \longrightarrow \mathbb{F}_{q^k}^*$$

$$(P, Q) \rightarrow \frac{f_{r,P}(Q)}{f_{r,Q}(P)}$$

Construction of the pairings

The Tate pairing

Let $P \in E(\mathbb{F}_q)[r]$, $Q \in E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k})$ and k be the embedding degree of the elliptic curve.

The Tate pairing is the bilinear map :

$$e_T : E(\mathbb{F}_q)[r] \times E(\mathbb{F}_{q^k})/rE(\mathbb{F}_{q^k}) \longrightarrow \mathbb{F}^*_{q^k}$$

$$(P, Q) \longrightarrow f_{r,P}(Q)^{\frac{q^k-1}{r}}$$

The Ate pairing is the latest optimisation of the Tate pairing. It is constructed by the same way.

The Ate pairing eats the T in Tate, and uses it in order to be computed with less iterations.
Let π_q be the Frobenius map over the elliptic curve :

$$\pi_q([x, y]) = [x^q, y^q]$$

t denotes the trace of the Frobenius over $E(\mathbb{F}_q)$ and $T = t - 1$.

Let $P \in E[r] \cap \mathrm{Ker}(\pi_q - [1])$ and $Q \in E[r] \cap \mathrm{Ker}(\pi_q - [q])$, i.e. Q verifying $\pi_q(Q) = [q]Q$.
The Ate pairing is the bilinear map :

$$e_A : E[r] \cap \mathrm{Ker}(\pi_q - [1]) \times E[r] \cap \mathrm{Ker}(\pi_q - [q]) \to \mathbb{F}^*_{q^k}$$

$$(P, Q) \to f_{T,P}(Q)^{\frac{q^k-1}{r}}$$

In order to compute the pairings, we need to compute the function $f_{r,P}$. The principal property of this function is that :

$$Div(f_{r,P}) = rDiv(P) - rDiv(P_\infty)$$

Victor Miller established the Miller equation :

$$f_{i+j,P} = f_{i,P} \times f_{j,P} \times \frac{l_{[i]P,[j]P}}{v_{[i+j]P}}$$

where $l_{[i]P+[j]P}$ is the line joining the points $[i]P$ and $[j]P$, and $v_{[i+j]P}$ is the vertical line passing through point $[i+j]P$.

Miller algorithm

Example

We want to compute $f_{7,P}$:

- $7 = 6 + 1$
- $f_{7,P} = f_{6,P} \times f_{1,P} \times \frac{l_{[6]P,P}}{v_{[7]P}}$

  $f_{1,P} = 1$

  $f_{7,P} = f_{6,P} \times \frac{l_{[6]P,P}}{v_{[7]P}}$
- $f_{6,P} = f_{3,P} \times f_{3,P} \times \frac{l_{[3]P,[3]P}}{v_{[6]P}}$

  when $i = j$, the line l is the tangent at point $[i]P$
- $f_{6,P} = f_{3,P}^2 \times \frac{l_{[3]P,[3]P}}{v_{[6]P}}$

  $f_{7,P} = f_{3,P}^2 \times \frac{l_{[3]P,[3]P}}{v_{[6]P}} \times \frac{l_{[6]P,P}}{v_{[7]P}}$

Miller algorithm

Example

We want to compute $f_{7,P}$:

- $f_{7,P} = f_{3,P}^2 \times \frac{l_{[3]P,[3]P}}{v_{[6]P}} \times \frac{l_{[6]P,P}}{v_{[7]P}}$
- $f_{3,P} = f_{2,P} \times f_{1,P} \times \frac{l_{[2]P,P}}{v_{[3]P}}$

  $f_{3,P} = f_{2,P} \times \frac{l_{[2]P,P}}{v_{[3]P}}$
- $f_{2,P} = f_{1,P} \times f_{1,P} \times \frac{l_{P,P}}{v_{[2]P}}$
- $f_{7,P} = \left(\frac{l_{P,P}}{v_{[2]P}} \times \frac{l_{[2]P,P}}{v_{[3]P}} \right)^2 \times \frac{l_{[3]P,[3]P}}{v_{[6]P}} \times \frac{l_{[6]P,P}}{v_{[7]P}}$

Computing pairings

Miller algorithm : return $f_{r,P}(Q)$

Data: $r = (r_n \dots l_0)_2$, $P \in E(\mathbb{F}_q)$ and $Q \in E(\mathbb{F}_{q^k})$;

Result: $f_{r,P}(Q) \in \mathbb{F}^*_{q^k}$;

1 : $T \leftarrow P$, $f_1 \leftarrow 1$, $f_2 \leftarrow 1$;

for $i = n - 1$ **to** 0 **do**

  2 : $T \leftarrow [2]T$;

  3 : $f_1 \longleftarrow f_1{}^2 \times l_1(Q)$;

  4 : $f_2 \longleftarrow f_2{}^2 \times v_2(Q)$;

  if $r_i = 1$ **then**

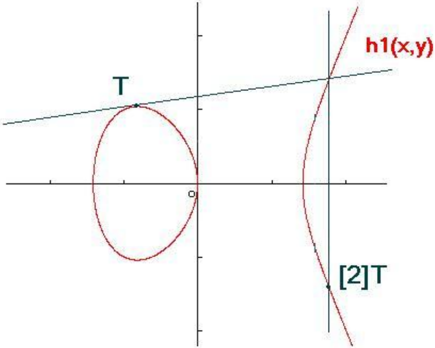
    5 : $T \leftarrow T + P$;

    ⋮

  end

end

return



Doubling on an elliptic curve

Computing pairings

Miller algorithm : return $f_{r,P}(Q)$

Data: $r = (r_n \ldots l_0)_2$, $P \in E(\mathbb{F}_q)$ and $Q \in E(\mathbb{F}_{q^k})$;

Result: $f_{r,P}(Q) \in \mathbb{F}^*_{q^k}$;

1 : $T \leftarrow P$, $f_1 \leftarrow 1$, $f_2 \leftarrow 1$;

for $i = n - 1$ **to** 0 **do**

  2 : $T \leftarrow [2]T$,;

  3 : $f_1 \longleftarrow {f_1}^2 \times l_d(Q)$;

  4 : $f_2 \longleftarrow {f_2}^2 \times v_d(Q)$;

  if $r_i = 1$ **then**

    5 : $T \leftarrow T + P$;

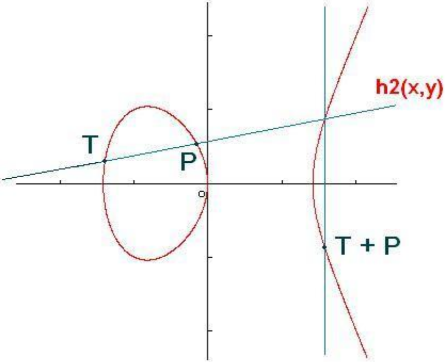
    6 : $f_1 \longleftarrow f_1 \times l_a(Q)$;

    7 : $f_2 \longleftarrow f_2 \times v_a(Q)$;

  end

end

return $\frac{f_1}{f_2}$



Addition on an elliptic curve

Implementation using Sage

Good points of Sage

- easy to write operation on the elliptic curve $P + Q$, and $2 * P$ for adding and multiplying point.
- the trace of the Frobenius is implemented
- random point on the elliptic curve
- the worksheet is very nice to use
- python quite easy to learn

Conclusion

To compute pairings, we have :

- arithmetic of finite field
- operation on elliptic curves

It is very easy to implement with Sage.
A "naive" implementation gives good result compare to Magma.
I have to improve my implementation, in order to have better performances.