An Improvement of the Elliptic Net Algorithm

Binglong Chen and **Chang-An Zhao**

Department of Mathematics
Sun Yat-sen University

Outline

1. Background
   - Usage of Elliptic Nets
   - Previous Work
2. Our Results
   - Main Results
   - Efficiency analysis and implementations

Outline

1. Background
   - Usage of Elliptic Nets
   - Previous Work
2. Our Results
   - Main Results
   - Efficiency analysis and implementations

Usage of Elliptic Nets

- Point counting or scalar multiplication (as elliptic nets of rank one are elliptic divisibility sequences or division polynomials)
- Solve ECDLP in special cases
- Computation of bilinear pairings (using elliptic net of rank two)

Usage of Elliptic Nets

- Point counting or scalar multiplication (as elliptic nets of rank one are elliptic divisibility sequences or division polynomials)
- Solve ECDLP in special cases
- Computation of bilinear pairings (using elliptic net of rank two)

Usage of Elliptic Nets

- Point counting or scalar multiplication (as elliptic nets of rank one are elliptic divisibility sequences or division polynomials)
- Solve ECDLP in special cases
- Computation of bilinear pairings (using elliptic net of rank two)

Usage of Elliptic Nets

- Point counting or scalar multiplication (as elliptic nets of rank one are elliptic divisibility sequences or division polynomials)
- Solve ECDLP in special cases
- Computation of bilinear pairings (using elliptic net of rank two)

Outline

1. Background
   - Usage of Elliptic Nets
   - Previous Work
2. Our Results
   - Main Results
   - Efficiency analysis and implementations

Previous Work

- Stange proposed the elliptic net algorithm to compute the Tate-Lichtenbaum Pairing (2007)
- Naoki et al.(2011) and Tang et al.(2014) compute the Ate-like pairings via the elliptic net algorithm
- Uchida et al. (2013) and Tran(2014) generalize the concept of elliptic nets to hyperelliptic case

Previous Work

- Stange proposed the elliptic net algorithm to compute the Tate-Lichtenbaum Pairing (2007)
- Naoki et al.(2011) and Tang et al.(2014) compute the Ate-like pairings via the elliptic net algorithm
- Uchida et al. (2013) and Tran(2014) generalize the concept of elliptic nets to hyperelliptic case

Previous Work

- Stange proposed the elliptic net algorithm to compute the Tate-Lichtenbaum Pairing (2007)
- Naoki et al.(2011) and Tang et al.(2014) compute the Ate-like pairings via the elliptic net algorithm
- Uchida et al. (2013) and Tran(2014) generalize the concept of elliptic nets to hyperelliptic case

Definition of an elliptic net

Definition

R - integral domain

G - finite-rank free abelian group

An elliptic net W : G $\to$ R satisfies the recurrence relation

$$W(p+q+s)W(p-q)W(r+s)W(r) + W(q+r+s)W(q-r)W(p+s)W(p) + W(r+p+s)W(r-p)W(q+s)W(q) = 0$$

for all $p, q, r, s \in G$.

Recurrence relation from matrices

Term	m_1	m_2	m_3	m_4
	$r+\frac{s}{2}$	$q+\frac{s}{2}$	$p+\frac{s}{2}$	$\frac{s}{2}$

Let A be a 4×4 anti-symmetric matrix defined by

$$A = (W(m_\rho + m_\lambda)(W(m_\rho - m_\lambda))_{1\leq\rho,\lambda\leq 4}$$

$$A = \begin{pmatrix} 0 & W(r+q+s)W(r-q) & W(r+p+s)W(r-p) & W(r+s)W(s) \\ & 0 & W(q+p+s)W(q-p) & W(q+s)W(q) \\ & & 0 & W(p+s)W(p) \\ & & & 0 \end{pmatrix}$$

Recurrence relation from matrices

Recurrence relation derived from

$$Pf(A) = 0$$

That is,

$$\begin{aligned} W(r+q+s)W(r-q)W(p+s)W(p) \\ - W(r+p+s)W(r-p)W(q+s)W(q) \\ + W(q+p+s)W(q-p)W(r+s)W(s) = 0 \end{aligned}$$

Construction of an elliptic net from elliptic curves

Theorem

(Stange 2007) E - elliptic curve over a field K
For all $v \in \mathbb{Z}^n$, there exist functions ψ_v

$$\psi_v : E^n \to K$$

such that
1. Each ψ_v is doubly periodic(or elliptic) in each variable
2.For any fixed $P \in E^n$, the function $W : \mathbb{Z}^n \to K$ defined by $W(v) = \psi_v(P)$ is an elliptic net.

Pairing computation via elliptic nets

Theorem

E - an elliptic net over a finite field K
m - a positive integer
P ∈ E(K)[m] Q ∈ E(K)
Tate-Lichtenbaum pairing defined by elliptic nets of rank 2

$$e(P,Q) = \frac{W(m+1,1)W(1,0)}{W(m+1,0)W(1,1)}$$

where $W(m,n) = \psi_{m,n}(P,Q)$.
key step in pair computation: compute W(n, i), i = 1 or 0 recursively.

Iteration step of the elliptic net algorithm

Double step:

		$(i-1,1)$	$(i,1)$	$(i+1,1)$			
$(i-3,0)$	$(i-2,0)$	$(i-1,0)$	$(i,0)$	$(i+1,0)$	$(i+2,0)$	$(i+3,0)$	$(i+4,0)$
			$\Downarrow$				
		$(2i-1,1)$	$(2i,1)$	$(2i+1,1)$			
$(2i-3,0)$	$(2i-2,0)$	$(2i-1,0)$	$(2i,0)$	$(2i+1,0)$	$(2i+2,0)$	$(2i+3,0)$	$(2i+4,0)$

DoubleAdd step:

		$(i-1,1)$	$(i,1)$	$(i+1,1)$			
$(i-3,0)$	$(i-2,0)$	$(i-1,0)$	$(i,0)$	$(i+1,0)$	$(i+2,0)$	$(i+3,0)$	$(i+4,0)$
			$\Downarrow$				
		$(2i,1)$	$(2i+1,1)$	$(2i+2,1)$			
$(2i-2,0)$	$(2i-1,0)$	$(2i,0)$	$(2i+1,0)$	$(2i+2,0)$	$(2i+3,0)$	$(2i+4,0)$	$(2i+5,0)$

In each loop, 11 variables should be updated always.

Iteration formula for W(n,0) and W(n,1)

Term	m_1	m_2	m_3	m_4	n_1	n_2	n_3	n_4
W(2i,0)	i+1	i-1	1	0	0	0	0	0
W(2i-1,0)	i	i-1	1	0	0	0	0	0
W(2i+j,1)	i	i+j	1	0	1	0	0	0

where $j = -1, 0, 1, 2$.

Let A be a 4×4 anti-symmetric matrix defined by

$$A = (W(m_\rho + m_\lambda, n_\rho + n_\lambda)(W(m_\rho - m_\lambda, n_\rho - n_\lambda))_{1 \le \rho, \lambda \le 4}$$

Iteration formula derived from

$$Pf(A) = 0$$

Iteration formula for W(2i,0)

$$\begin{pmatrix} 0 & (2i,0)(2,0) & (i+2,0)(i,0) & (i+1,0)^2 \\ & 0 & (i,0)(i-2,0) & (i-1,0)^2 \\ & & 0 & (1,0)^2 \\ & & & 0 \end{pmatrix} = 0$$

$$\begin{aligned} W(2i,0)W(2,0)W(1,0)^2 & \\ & - W(i+2,0)W(i,0)W(i-1,0)^2 \\ & \quad + W(i+1,0)^2 W(i,0)W(i-2,0) = 0 \end{aligned}$$

Iteration formula for W(2i-1,0)

$$\begin{pmatrix} 0 & (2i-1,0)(1,0) & (i+1,0)(i-1,0) & (i,0)^2 \\ & 0 & (i,0)(i-2,0) & (i-1,0)^2 \\ & & 0 & (1,0)^2 \\ & & & 0 \end{pmatrix} = 0$$

$$\begin{aligned} &W(2i-1,0)W(1,0)^3 \\ &\quad - W(i+1,0)W(i-1,0)^3 \\ &\quad\quad + W(i,0)^3 W(i-2,0) = 0 \end{aligned}$$

Iteration formula for W(2i-1,1)

$$\begin{pmatrix} 0 & (2i-1,1)(1,1) & (i+1,1)(i-1,1) & (i,1)^2 \\ & 0 & (i,0)(i-2,0) & (i-1,0)^2 \\ & & 0 & (1,0)^2 \\ & & & 0 \end{pmatrix} = 0$$

$$\begin{aligned} W(2i-1,1)W(1,1)W(1,0)^2 & \\ -W(i+1,1)W(i-1,1)W(i-1,0)^2 & \\ +W(i,1)^2W(i,0)W(i-2,0) &= 0 \end{aligned}$$

Iteration formula for W(2i,1)

$$\begin{pmatrix} 0 & (2i,1)(0,1) & (i+1,1)(i-1,1) & (i,1)^2 \\ & 0 & (i+1,0)(i-1,0) & (i,0)^2 \\ & & 0 & (1,0)^2 \\ & & & 0 \end{pmatrix} = 0$$

$$\begin{aligned} W(2i,1)W(0,1)W(1,0)^2 \\ - W(i+1,1)W(i-1,1)W(i,0)^2 \\ + W(i,1)^2 W(i+1,0)W(i-1,0) = 0 \end{aligned}$$

Iteration formula for W(2i+1,1)

$$\begin{pmatrix} 0 & (2i+1,1)(-1,1) & (i+1,1)(i-1,1) & (i,1)^2 \\ & 0 & (i+2,0)(i,0) & (i+1,0)^2 \\ & & 0 & (1,0)^2 \\ & & & 0 \end{pmatrix} = 0$$

$$\begin{aligned} W(2i+1,1)W(-1,1)W(1,0)^2 \\ - W(i+1,1)W(i-1,1)W(i+1,0)^2 \\ + W(i,1)^2 W(i+2,0)W(i,0) = 0 \end{aligned}$$

Iteration formula for W(2i+2,1)

$$\begin{pmatrix} 0 & (2i+2,1)(-2,1) & (i+1,1)(i-1,1) & (i,1)^2 \\ & 0 & (i+3,0)(i+1,0) & (i+2,0)^2 \\ & & 0 & (1,0)^2 \\ & & & 0 \end{pmatrix} = 0$$

$$W(2i+2,1)W(-2,1)W(1,0)^2 \\ - W(i+2,0)^2 W(i+1,1)W(i-1,1) \\ + W(i+3,0)W(i+1,0)W(i,1)^2 = 0$$

Outline

1. Background
   - Usage of Elliptic Nets
   - Previous Work
2. Our Results
   - Main Results
   - Efficiency analysis and implementations

Improved elliptic net algorithms

1. Update iteration loops using *10* intermediate variables
2. Convert elliptic net algorithms in a non-adjacent form
3. Make W(2,0)=1 by using the equivalence of elliptic nets and choosing special base fields.

New Double steps

$$
\begin{array}{ccccccc}
 & & (i-1,1) & (i,1) & (i+1,1) & & \\
(i-3,0) & (i-2,0) & (i-1,0) & (i,0) & (i+1,0) & (i+2,0) & (i+3,0) \\
 & & & \Downarrow & & & \\
 & & (2i-1,1) & (2i,1) & (2i+1,1) & & \\
(2i-3,0) & (2i-2,0) & (2i-1,0) & (2i,0) & (2i+1,0) & (2i+2,0) & (2i+3,0)
\end{array}
$$

Fact

W(i+4,0) is not necessary for updating process of the double steps. This will save some costs.

New DoubleAdd steps

$$\begin{array}{ccccccc}
 & & (i-1,1) & (i,1) & (i+1,1) & & \\
(i-3,0) & (i-2,0) & (i-1,0) & (i,0) & (i+1,0) & (i+2,0) & (i+3,0) \\
 & & & \Downarrow & & & \\
 & & (2i,1) & (2i+1,1) & (2i+2,1) & & \\
(2i-2,0) & (2i-1,0) & (2i,0) & (2i+1,0) & (2i+2,0) & (2i+3,0) & (2i+4,0)
\end{array}$$

How to obtain W(2i+4,0)

Term	m_1	m_2	m_3	m_4	n_1	n_2	n_3	n_4
2i+4	2i+2	2	1	0	0	0	0	0

$$\begin{pmatrix} 0 & (2i+4,0)(2i,0) & (2i+3,0)(2i+1,0) & (2i+2,0)^2 \\ & 0 & (3,0)(1,0) & (2,0)^2 \\ & & 0 & (1,0)^2 \\ & & & 0 \end{pmatrix} = 0$$

$$W(2i+4,0)W(2i,0)W(1,0)^2 - W(2i+3,0)W(2i+1,0)W(2,0)^2 + W(2i+2,0)^2W(2i+3,0)W(2i+1,0) = 0$$

- All terms appeared in the formula of W(2i+4,0) have been computed.
- The cost for W(2i+4,0) will be 1I + 3M.

DoubleSubtraction steps

		$(i-1,1)$	$(i,1)$	$(i+1,1)$		
$(i-3,0)$	$(i-2,0)$	$(i-1,0)$	$(i,0)$	$(i+1,0)$	$(i+2,0)$	$(i+3,0)$
			$\Downarrow$			
		$(2i-2,1)$	$(2i-1,1)$	$(2i,1)$		
$(2i-4,0)$	$(2i-3,0)$	$(2i-2,0)$	$(2i-1,0)$	$(2i,0)$	$(2i+2,0)$	$(2i+3,0)$

How to obtain W(2i-4,0)

Term	m_1	m_2	m_3	m_4	n_1	n_2	n_3	n_4
(2i-4,0)	2i-2	2	1	0	0	0	0	0

$$\begin{pmatrix} 0 & (2i,0)(2i-4,0) & (2i-3,0)(2i-1,0) & (2i-2,0)^2 \\ & 0 & (3,0)(1,0) & (2,0)^2 \\ & & 0 & (1,0)^2 \\ & & & 0 \end{pmatrix} = 0$$

$$W(2i-4,0)W(2i,0)W(1,0)^2 - W(2i-3,0)W(2i-1,0)W(2,0)^2 + W(2i-2,0)^2W(3,0)W(1,0) = 0$$

How to obtain W(2i-2,1)

Term	m_1	m_2	m_3	m_4	n_1	n_2	n_3	n_4
(2i-2,1)	i	i-2	1	0	1	0	0	0

$$\begin{pmatrix} 0 & (2i-2,1)(2,1) & (i+1,1)(i-1,1) & (i,1)^2 \\ & 0 & (i-1,0)(i-3,0) & (i-2,0)^2 \\ & & 0 & (1,0)^2 \\ & & & 0 \end{pmatrix} = 0$$

$$W(2i-2,1)W(2,1)W(1,0)^2 - W(i+1,1)W(i-1,1)W(i-2,0)^2 + W(i-1,0)W(i-3,0)W(i,1)^2 = 0$$

Outline

1. Background
   - Usage of Elliptic Nets
   - Previous Work
2. Our Results
   - Main Results
   - Efficiency analysis and implementations

Efficiency analysis

Table: Cost of the *Double*(V) algorithm for the different methods

Method	**Operation Count**
Elliptic Net algorithm(Stange2007)	$6S+(26+6i)M+S_i+\frac{3}{2}M_i$
This work	$5S+(22+6i)M+S_i+\frac{3}{2}M_i$

Efficiency analysis

Table: Cost of the DoubleAdd/Sub(V) algorithm for the different cases

Method	Operation count
Elliptic Net algorithm(Stange2007)	$6S + (26 + 6i)M + S_i + 2M_i$
This work	$5S + (23 + 6i)M + I + S_i + 2M_i$

Efficiency analysis

Table: Maximal value of the density ρ for the proposed method

Density	l = 10M	l = 20M	l = 30M
ρ	0.44	0.23	0.15

ρ - density of non-zero digits of the integer m in NAF representation.

Implementation results

Curve parameters

- $r = 2^{255} + 2^{41} + 1$
- $p = 12 \cdot (2^{1280} + 2^{31} + 2^{15}) \cdot r - 1;$
- $F_{p^2} = F_p[i]/(i^2 + 1)$
- $E : y^2 = x^3 - 3x$ over F_p

Running environment specification: Ubuntu Kylin 14.04 64bits, Core i5-4670 CPU 3.40GHz$\times$ 4, and memory, 8GB, Magma language.

Implementation Timing

Table: Cost of computing $f_{r,P}(Q)$ by the different methods-128 security level

Method	Operation Count	Time(ms)
Stange's algorithm	$11554.5M$	37.8
This work	$10352.5M$	33.2
Miller's algorithm	$4164M$	14.9

Summary

- Elliptic net algorithms have been improved when the loop parameter r has low Hamming weight.
- Miller's algorithm is still a valid candidate for practical pairing-based implementations
- More developments of the Elliptic Net algorithm should be required in future.

Thank you for your attention!

More details can be found in http://eprint.iacr.org/2015/276