

# Point Counting Algorithms - A Survey

CSCI 599 Prof. Ming Deh-Huang

Heekwan Lee, Iftikhar A Burhanuddin

# Outline

# Outline

Hasse's Theorem

# Outline

Hasse's Theorem  
Sato's Algorithm

# Outline

Hasse's Theorem  
Sato's Algorithm  
Future Directions

# Hasse's Theorem

*Let  $E/K$  be an elliptic curve defined over a finite field  $\mathbb{F}_q$  with  $q$  elements. Then*

$$|\#E(\mathbb{F}_q) - q - 1| \leq 2\sqrt{q}$$

# Hasse's Theorem

Let  $E/K$  be an elliptic curve defined over a finite field  $\mathbb{F}_q$  with  $q$  elements. Then

$$|\#E(\mathbb{F}_q) - q - 1| \leq 2\sqrt{q}$$

deg of map = det of map for  $T_l[E] = \mathbb{Z}_l \oplus \mathbb{Z}_l$ ,

# Hasse's Theorem

Let  $E/K$  be an elliptic curve defined over a finite field  $\mathbb{F}_q$  with  $q$  elements. Then

$$|\#E(\mathbb{F}_q) - q - 1| \leq 2\sqrt{q}$$

deg of map = det of map for  $T_l[E] = \mathbb{Z}_l \oplus \mathbb{Z}_l$ ,

$$\text{Tr}(\phi) = 1 + \det(\phi) - \det(\phi - id)$$

# Hasse's Theorem

Let  $E/K$  be an elliptic curve defined over a finite field  $\mathbb{F}_q$  with  $q$  elements. Then

$$|\#E(\mathbb{F}_q) - q - 1| \leq 2\sqrt{q}$$

deg of map = det of map for  $T_l[E] = \mathbb{Z}_l \oplus \mathbb{Z}_l$ ,

$$\text{Tr}(\phi) = 1 + \det(\phi) - \det(\phi - id)$$

We know that  $\#E(\mathbb{F}_p) = \det(\phi - id)$

# $p$ -adic Numbers

**$p$ -adic numbers [FGH]:** Let  $\pi_n$  be the projection from  $\mathbb{Z}/p^{n+1}\mathbb{Z}$  onto  $\mathbb{Z}/p^n\mathbb{Z}$ . This projection is a ring homomorphism. One can give a formal definition of  $p$ -adic integers as follows.

**Definition:** A  $p$ -adic integer is a sequence  $x = (x_1, x_2, \dots, x_n, \dots)$  with  $x_n \in \mathbb{Z}/p^n\mathbb{Z}$  and such that  $\pi_n(x_{n+1}) = x_n$  for  $n \geq 1$ . The ring of  $p$ -adic integers is denoted by  $\mathbb{Z}_p$ .

**$q$ -adic numbers [FGH]:** Let  $q = p^n$  with  $p$  prime. Let  $f(t)$  be a monic polynomial in  $\mathbb{Z}_p[t]$  of degree  $n$  such that the polynomial  $\pi(f)$  obtained by projecting the coefficients is irreducible in  $\mathbb{F}_p[t]$ .

**Definition:** The ring  $\mathbb{Z}_q$  is  $\mathbb{Z}_p[t]$  modulo (the ideal generated by)  $f(t)$ .

# Satoh's Algorithm

**Goal:** Compute Trace of Frobenius in  $E$

# Satoh's Algorithm

**Goal:** Compute Trace of Frobenius in  $E$

Trace upstairs

# Satoh's Algorithm

**Goal:** Compute Trace of Frobenius in  $E$

$$\text{Trace upstairs} = \text{Tr}(\mathcal{F}) = \text{Tr}(\hat{\mathcal{F}})$$

# Satoh's Algorithm

**Goal:** Compute Trace of Frobenius in  $E$

$$\begin{aligned}\text{Trace upstairs} &= \text{Tr}(\mathcal{F}) = \text{Tr}(\hat{\mathcal{F}}) \\ &= \text{Tr}(\hat{F})\end{aligned}$$

# Satoh's Algorithm

**Goal:** Compute Trace of Frobenius in  $E$

$$\begin{aligned}\text{Trace upstairs} &= \text{Tr}(\mathcal{F}) = \text{Tr}(\hat{\mathcal{F}}) \\ &= \text{Tr}(\hat{F}) = \text{Tr}(F) =\end{aligned}$$

# Satoh's Algorithm

**Goal:** Compute Trace of Frobenius in  $E$

$$\begin{aligned}\text{Trace upstairs} &= \text{Tr}(\mathcal{F}) = \text{Tr}(\hat{\mathcal{F}}) \\ &= \text{Tr}(\hat{F}) = \text{Tr}(F) = \text{Trace downstairs}\end{aligned}$$

# Satoh's Algorithm

**Goal:** Compute Trace of Frobenius in  $E$

$$\begin{aligned}\text{Trace upstairs} &= \text{Tr}(\mathcal{F}) = \text{Tr}(\hat{\mathcal{F}}) \\ &= \text{Tr}(\hat{F}) = \text{Tr}(F) = \text{Trace downstairs}\end{aligned}$$

So we compute Trace upstairs

$$\text{Tr}(\hat{\mathcal{F}}) = c + \frac{q}{c}$$

# Satoh's Algorithm

**Goal:** Compute Trace of Frobenius in  $E$

$$\begin{aligned}\text{Trace upstairs} &= \text{Tr}(\mathcal{F}) = \text{Tr}(\hat{\mathcal{F}}) \\ &= \text{Tr}(\hat{F}) = \text{Tr}(F) = \text{Trace downstairs}\end{aligned}$$

So we compute Trace upstairs

$$\text{Tr}(\hat{\mathcal{F}}) = c + \frac{q}{c}$$

The formula for  $c$  was discovered by **Skjerna**

# Satoh's Algorithm

**Goal:** Compute Trace of Frobenius in  $E$

$$\begin{aligned}\text{Trace upstairs} &= \text{Tr}(\mathcal{F}) = \text{Tr}(\hat{\mathcal{F}}) \\ &= \text{Tr}(\hat{F}) = \text{Tr}(F) = \text{Trace downstairs}\end{aligned}$$

So we compute Trace upstairs

$$\text{Tr}(\hat{\mathcal{F}}) = c + \frac{q}{c}$$

The formula for  $c$  was discovered by **Skjernaa**

**J Vélu: Isogénies entre courbes elliptiques (1971)**

# Satoh's Algorithm

**Goal:** Compute Trace of Frobenius in  $E$

$$\begin{aligned}\text{Trace upstairs} &= \text{Tr}(\mathcal{F}) = \text{Tr}(\hat{\mathcal{F}}) \\ &= \text{Tr}(\hat{F}) = \text{Tr}(F) = \text{Trace downstairs}\end{aligned}$$

So we compute Trace upstairs

$$\text{Tr}(\hat{\mathcal{F}}) = c + \frac{q}{c}$$

The formula for  $c$  was discovered by **Skjernaa**

**J Vélu: Isogénies entre courbes elliptiques (1971)**

$$\mathcal{E} \xrightarrow{\text{Isogeny } F} \mathcal{E}' \text{ over } \mathbb{F}_p$$

# Satoh's Algorithm

**Goal:** Compute Trace of Frobenius in  $E$

$$\begin{aligned}\text{Trace upstairs} &= \text{Tr}(\mathcal{F}) = \text{Tr}(\hat{\mathcal{F}}) \\ &= \text{Tr}(\hat{F}) = \text{Tr}(F) = \text{Trace downstairs}\end{aligned}$$

So we compute Trace upstairs

$$\text{Tr}(\hat{\mathcal{F}}) = c + \frac{q}{c}$$

The formula for  $c$  was discovered by **Skjernaa**

**J Vélu: Isogénies entre courbes elliptiques (1971)**

$$\mathcal{E} \xrightarrow{\text{Isogeny } F} \mathcal{E}' \text{ over } \mathbb{F}_p$$

Vélu discovered the formula for  $\text{Trace}(F)$  in terms of the  $J$ -invariant

# Satoh's Algorithm

**Goal:** Compute Trace of Frobenius in  $E$

$$\begin{aligned}\text{Trace upstairs} &= \text{Tr}(\mathcal{F}) = \text{Tr}(\hat{\mathcal{F}}) \\ &= \text{Tr}(\hat{F}) = \text{Tr}(F) = \text{Trace downstairs}\end{aligned}$$

So we compute Trace upstairs

$$\text{Tr}(\hat{\mathcal{F}}) = c + \frac{q}{c}$$

The formula for  $c$  was discovered by **Skjernaa**

**J Vélu: Isogénies entre courbes elliptiques (1971)**

$$\mathcal{E} \xrightarrow{\text{Isogeny } F} \mathcal{E}' \text{ over } \mathbb{F}_p$$

Vélu discovered the formula for  $\text{Trace}(F)$  in terms of the  $J$ -invariant

**Satoh: The canonical Lift of an ordinary Elliptic curve over a Finite Field and Its Point Counting (1999)**

Satoh was the first to relate  $J$ -invariant,  $J'$ -invariant (Frobenius automorphism) to counting points in Elliptic Curves

Satoh was the first to relate  $J$ -invariant,  $J'$ -invariant (Fronbenius automorphism) to counting points in Elliptic Curves

$$Tr(\mathbb{F}_q) = c_1 + \frac{q}{c_1}$$

Satoh was the first to relate  $J$ -invariant,  $J'$ -invariant (Fronbenius automorphism) to counting points in Elliptic Curves

$$Tr(\mathbb{F}_q) = c_1 + \frac{q}{c_1}$$

## B. Skjerna: Satoh's algorithm in characteristic 2 (2000)

Satoh was the first to relate  $J$ -invariant,  $J'$ -invariant (Frobenius automorphism) to counting points in Elliptic Curves

$$\text{Tr}(\mathbb{F}_q) = c_1 + \frac{q}{c_1}$$

## B. Skjerna: Satoh's algorithm in characteristic 2 (2000)

He developed an explicit formula of the Trace of the Frobenius map in terms of the  $J$ -invariant

**Proposition:** Let  $\tau_i = -\frac{x}{y}$  be the local parameter of

$\mathcal{E}_i$  at  $O$  and let  $c_i$  be defined as  $\tau_{i+1}$ .

$$\hat{\Sigma}_i = c_i \tau_i + O(\tau_i^2)$$

Satoh was the first to relate  $J$ -invariant,  $J'$ -invariant (Frobenius automorphism) to counting points in Elliptic Curves

$$Tr(\mathbb{F}_q) = c_1 + \frac{q}{c_1}$$

## B. Skjerna: Satoh's algorithm in characteristic 2 (2000)

He developed an explicit formula of the Trace of the Frobenius map in terms of the  $J$ -invariant

**Proposition:** Let  $\tau_i = -\frac{x}{y}$  be the local parameter of

$\mathcal{E}_i$  at  $O$  and let  $c_i$  be defined as  $\tau_{i+1}$ .

$$\hat{\Sigma}_i = c_i \tau_i + O(\tau_i^2)$$

Denote the non-trivial point in  $\ker(\hat{\Sigma}_i)$  by  $Q_i = (x_i, y_i)$  and  $z_i = \frac{x_i}{2}$  and

Satoh was the first to relate  $J$ -invariant,  $J'$ -invariant (Frobenius automorphism) to counting points in Elliptic Curves

$$Tr(\mathbb{F}_q) = c_1 + \frac{q}{c_1}$$

## B. Skjerna: Satoh's algorithm in characteristic 2 (2000)

He developed an explicit formula of the Trace of the Frobenius map in terms of the  $J$ -invariant

**Proposition:** Let  $\tau_i = -\frac{x}{y}$  be the local parameter of

$\mathcal{E}_i$  at  $O$  and let  $c_i$  be defined as  $\tau_{i+1}$ .

$$\hat{\Sigma}_i = c_i \tau_i + O(\tau_i^2)$$

Denote the non-trivial point in  $\ker(\hat{\Sigma}_i)$  by  $Q_i = (x_i, y_i)$  and  $z_i = \frac{x_i}{2}$  and

$t_i = (12z_i^2 + z_i)(j(\mathcal{E}_i) - 1728) - 36$ , then

Satoh was the first to relate  $J$ -invariant,  $J'$ -invariant (Frobenius automorphism) to counting points in Elliptic Curves

$$\text{Tr}(\mathbb{F}_q) = c_1 + \frac{q}{c_1}$$

## B. Skjerna: Satoh's algorithm in characteristic 2 (2000)

He developed an explicit formula of the Trace of the Frobenius map in terms of the  $J$ -invariant

**Proposition:** Let  $\tau_i = -\frac{x}{y}$  be the local parameter of

$\mathcal{E}_i$  at  $O$  and let  $c_i$  be defined as  $\tau_{i+1}$ .

$$\hat{\Sigma}_i = c_i \tau_i + O(\tau_i^2)$$

Denote the non-trivial point in  $\ker(\hat{\Sigma}_i)$  by  $Q_i = (x_i, y_i)$  and  $z_i = \frac{x_i}{2}$  and

$t_i = (12z_i^2 + z_i)(j(\mathcal{E}_i) - 1728) - 36$ , then

$$c_i^2 = \frac{j(\mathcal{E}_i) - (504 + 12096z_i)t_i}{j(\mathcal{E}_i) + 240t_i}$$

## **M. Fouquet, P. Gaudry et R. Harley: An extension of Satoh's algorithm and its implementation (2000)**

**M. Fouquet, P. Gaudry et R. Harley: An extension of Satoh's algorithm and its implementation (2000)**

$$\begin{array}{cccccccc}
 \mathcal{E}_0 & \xrightarrow{\widehat{\Sigma}_0} & \mathcal{E}_1 & \xrightarrow{\widehat{\Sigma}_1} & \dots & \xrightarrow{\widehat{\Sigma}_{n-2}} & \mathcal{E}_{n-1} & \xrightarrow{\widehat{\Sigma}_{n-1}} & \mathcal{E}_0 \\
 \uparrow & & \uparrow & & & & \uparrow & & \uparrow \\
 E_0 & \xrightarrow{\widehat{\sigma}_0} & E_1 & \xrightarrow{\widehat{\sigma}_1} & \dots & \xrightarrow{\widehat{\sigma}_{n-2}} & E_{n-1} & \xrightarrow{\widehat{\sigma}_{n-1}} & E_0
 \end{array}$$

Start with  $E_0$  elliptic curve over  $\mathbb{F}_q$ ,  $q = p^n$

**M. Fouquet, P. Gaudry et R. Harley: An extension of Satoh's algorithm and its implementation (2000)**

$$\begin{array}{cccccccc}
 \mathcal{E}_0 & \xrightarrow{\widehat{\Sigma}_0} & \mathcal{E}_1 & \xrightarrow{\widehat{\Sigma}_1} & \dots & \xrightarrow{\widehat{\Sigma}_{n-2}} & \mathcal{E}_{n-1} & \xrightarrow{\widehat{\Sigma}_{n-1}} & \mathcal{E}_0 \\
 \uparrow & & \uparrow & & & & \uparrow & & \uparrow \\
 E_0 & \xrightarrow{\widehat{\sigma}_0} & E_1 & \xrightarrow{\widehat{\sigma}_1} & \dots & \xrightarrow{\widehat{\sigma}_{n-2}} & E_{n-1} & \xrightarrow{\widehat{\sigma}_{n-1}} & E_0
 \end{array}$$

Start with  $E_0$  elliptic curve over  $\mathbb{F}_q$ ,  $q = p^n$

Apply little Frobenius, have cycle

**M. Fouquet, P. Gaudry et R. Harley: An extension of Satoh's algorithm and its implementation (2000)**

$$\begin{array}{cccccccc}
 \mathcal{E}_0 & \xrightarrow{\widehat{\Sigma}_0} & \mathcal{E}_1 & \xrightarrow{\widehat{\Sigma}_1} & \dots & \xrightarrow{\widehat{\Sigma}_{n-2}} & \mathcal{E}_{n-1} & \xrightarrow{\widehat{\Sigma}_{n-1}} & \mathcal{E}_0 \\
 \uparrow & & \uparrow & & & & \uparrow & & \uparrow \\
 E_0 & \xrightarrow{\widehat{\sigma}_0} & E_1 & \xrightarrow{\widehat{\sigma}_1} & \dots & \xrightarrow{\widehat{\sigma}_{n-2}} & E_{n-1} & \xrightarrow{\widehat{\sigma}_{n-1}} & E_0
 \end{array}$$

Start with  $E_0$  elliptic curve over  $\mathbb{F}_q$ ,  $q = p^n$

Apply little Frobenius, have cycle

Lift to  $\mathcal{E}_0$  elliptic curve over  $\mathbb{Z}_q$

## Time and space complexity analysis

**[FGH]** So for the field  $\mathbb{F}_{p^n}$  we work with  $\mathbb{Z}_{p^n}$  with  $O(p^{O(n)})$  precision. So each "coefficient" takes  $O(n \log p)$  and each element of  $\mathbb{Z}_{p^n}$  takes  $O(n^2 \log p)$ .

# Time and space complexity analysis

[FGH] So for the field  $\mathbb{F}_{p^n}$  we work with  $\mathbb{Z}_p^n$  with  $O(p^{O(n)})$  precision. So each "coefficient" takes  $O(n \log p)$  and each element of  $\mathbb{Z}_p^n$  takes  $O(n^2 \log p)$ .

Working with *bounded* number of elements for each of the  $n$  conjugate curve, the total memory is  $O(n^3 \log p)$

# Time and space complexity analysis

**[FGH]** So for the field  $\mathbb{F}_{p^n}$  we work with  $\mathbb{Z}_p^n$  with  $O(p^{O(n)})$  precision. So each "coefficient" takes  $O(n \log p)$  and each element of  $\mathbb{Z}_p^n$  takes  $O(n^2 \log p)$ .

Working with *bounded* number of elements for each of the  $n$  conjugate curve, the total memory is  $O(n^3 \log p)$

**Satoh's Theorem [FGH]:** Let  $E$  be an elliptic curve over the finite field with  $q = p^n$  elements. Assume that  $j(E) \notin \mathbb{F}_{p^2}$ . Then there exists a deterministic algorithm for computing the order of  $E$ , which for fixed  $p$ , requires  $O(n^3)$  memory and  $O(n^{3+\varepsilon})$  bit operations.

# Time and space complexity analysis

**[FGH]** So for the field  $\mathbb{F}_{p^n}$  we work with  $\mathbb{Z}_p^n$  with  $O(p^{O(n)})$  precision. So each "coefficient" takes  $O(n \log p)$  and each element of  $\mathbb{Z}_p^n$  takes  $O(n^2 \log p)$ .

Working with *bounded* number of elements for each of the  $n$  conjugate curve, the total memory is  $O(n^3 \log p)$

**Satoh's Theorem [FGH]:** Let  $E$  be an elliptic curve over the finite field with  $q = p^n$  elements. Assume that  $j(E) \notin \mathbb{F}_{p^2}$ . Then there exists a deterministic algorithm for computing the order of  $E$ , which for fixed  $p$ , requires  $O(n^3)$  memory and  $O(n^{3+\varepsilon})$  bit operations.

A enhanced version of Satoh's algorithm was designed by Vercauteren et al which also runs in the same amount of time asymptotically but consumes  $O(n^2)$  of memory compared to Satoh's  $O(n^3)$

# Time and space complexity analysis

**[FGH]** So for the field  $\mathbb{F}_{p^n}$  we work with  $\mathbb{Z}_p^n$  with  $O(p^{O(n)})$  precision. So each "coefficient" takes  $O(n \log p)$  and each element of  $\mathbb{Z}_p^n$  takes  $O(n^2 \log p)$ .

Working with *bounded* number of elements for each of the  $n$  conjugate curve, the total memory is  $O(n^3 \log p)$

**Satoh's Theorem [FGH]:** Let  $E$  be an elliptic curve over the finite field with  $q = p^n$  elements. Assume that  $j(E) \notin \mathbb{F}_{p^2}$ . Then there exists a deterministic algorithm for computing the order of  $E$ , which for fixed  $p$ , requires  $O(n^3)$  memory and  $O(n^{3+\varepsilon})$  bit operations.

A enhanced version of Satoh's algorithm was designed by Vercauteren et al which also runs in the same amount of time asymptotically but consumes  $O(n^2)$  of memory compared to Satoh's  $O(n^3)$

**Records [PG]:** Satoh's algorithm -  $\mathbb{F}_{5^{569}}$

FGH's extension to even char -  $\mathbb{F}_{2^{8009}}$

# Time and space complexity analysis

**[FGH]** So for the field  $\mathbb{F}_{p^n}$  we work with  $\mathbb{Z}_p^n$  with  $O(p^{O(n)})$  precision. So each "coefficient" takes  $O(n \log p)$  and each element of  $\mathbb{Z}_p^n$  takes  $O(n^2 \log p)$ .

Working with *bounded* number of elements for each of the  $n$  conjugate curve, the total memory is  $O(n^3 \log p)$

**Satoh's Theorem [FGH]:** Let  $E$  be an elliptic curve over the finite field with  $q = p^n$  elements. Assume that  $j(E) \notin \mathbb{F}_{p^2}$ . Then there exists a deterministic algorithm for computing the order of  $E$ , which for fixed  $p$ , requires  $O(n^3)$  memory and  $O(n^{3+\varepsilon})$  bit operations.

A enhanced version of Satoh's algorithm was designed by Vercauteren et al which also runs in the same amount of time asymptotically but consumes  $O(n^2)$  of memory compared to Satoh's  $O(n^3)$

**Records [PG]:** Satoh's algorithm -  $\mathbb{F}_{5^{569}}$

FGH's extension to even char -  $\mathbb{F}_{2^{8009}}$

# Future Directions

- As always...more efficient algorithms wrt computation and memory

# Future Directions

- As always...more efficient algorithms wrt computation and memory
- Lower bounds for the point counting problem

# Future Directions

- As always...more efficient algorithms wrt computation and memory
- Lower bounds for the point counting problem
- "It is possible to write endlessly on elliptic curves. (This is not a threat.)" - **Serge Lang**

# Future Directions

- As always...more efficient algorithms wrt computation and memory
- Lower bounds for the point counting problem
- "It is possible to write endlessly on elliptic curves. (This is not a threat.)" - **Serge Lang**

*Remark:* Point counting in char 2 is almost as fast as an RSA key generation **[PG]**

## References

- RS Schoof, Rene, Counting Points On Elliptic Curves Over Finite Fields, Journal de Theorie des Nombres de Bordeaux 7, 1995
- FGH M. Fouquet, P. Gaudry and R. Harley, An extension of Satoh's algorithm and its implementation, J. Ramanujan Math. Soc. 15, 2000
- VPV Frederik Vercauteren, Bart Preneel, Joos Vandewalle, A Memory Efficient Version of Satoh's Algorithm, Advances in Cryptology - Eurocrypt 2001, Lecture Notes in Computer Science 2045, Springer, 2001.
- KG Kristian Gjøsteen, Schoof's algorithm, preprint, 2000
- AB Antonia W. Bluher, A Leisurely Introduction to Formal Groups and Elliptic Curves, preprint
- PG Pierrick Gaudry, Algorithms for counting points on curves, presentation, ECC 2001
- BSS Ian F. Blake, Gadiel Seroussi, and Nigel P. Smart, Elliptic Curves in Cryptography, Cambridge University Press, Cambridge, 1999
- AE Andreas Enge, Elliptic Curves and Their Applications to Cryptography - An Introduction, Kluwer Academic Publishers, 1999
- JHS Silverman, J. H., The Arithmetic of Elliptic Curves, Graduate Texts in Mathematics, Springer-Verlag, 1986
- HC Cohen, Henri, A Course in Computational Algebraic Number Theory, Graduate Texts in Mathematics, Springer-Verlag, 1993
- PGP Cohen, Henri et al, User's Guide to PARI/GP, 2000