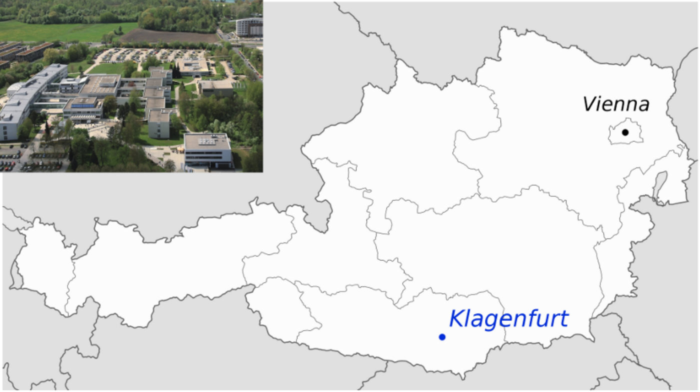
Computational Aspects of Hyperelliptic Curve Cryptography

Michela Mazzoli

Institut für Mathematik
Alpen-Adria-Universität Klagenfurt

Torino, 22 Dicembre 2014

Alpen-Adria-Universität Klagenfurt, Austria



Motivation 1: DLP-based crypto

Alice and Bob want to exchange private messages over a public channel. They agree on a secret key with the following scheme:

1. let $G = \langle g \rangle$ be a cyclic group (publicly known)
2. Alice chooses an integer a and sends g^a to Bob
3. Bob chooses an integer b and sends g^b to Alice
4. Alice computes $(g^b)^a$
5. Bob computes $(g^a)^b$
6. the common secret key is g^{ab}

Security relies on the fact that it is *hard* to find b from g^a and g^{ab}.

This is equivalent to solve the Discrete Logarithm Problem, and no polynomial-time algorithm for the DLP is known.

Motivation 2: pairing-based crypto

Let $(G_1, +)$ and $(G_2, \cdot)$ be cyclic groups of prime order q.
A pairing map is $\varepsilon : G_1 \times G_1 \to G_2$ such that

1. ε is bilinear: $\varepsilon(aP, bQ) = \varepsilon(P, Q)^{ab} \quad \forall a, b \in \mathbb{F}_q^* \;\; \forall P, Q \in G_1$
2. ε is non-degenerative: $P \neq 0 \;\Rightarrow\; e(P, P) \neq 1$
3. ε is efficiently computable

Motivation 2: pairing-based crypto

Let $(G_1, +)$ and $(G_2, \cdot)$ be cyclic groups of prime order q.
A pairing map is $\varepsilon : G_1 \times G_1 \to G_2$ such that

1. ε is bilinear: $\varepsilon(aP, bQ) = \varepsilon(P, Q)^{ab} \quad \forall a, b \in \mathbb{F}_q^* \;\; \forall P, Q \in G_1$
2. ε is non-degenerative: $P \neq 0 \;\Rightarrow\; e(P, P) \neq 1$
3. ε is efficiently computable

Weil pairing:

- G_1 is a subgroup of
  - the group of points of an elliptic curve over a finite field
  - the Jacobian of a hyperelliptic curve over a finite field
- G_2 is the group of the q-th roots of unity

ALPEN-ADRIA UNIVERSITÄT KLAGENFURT | WIEN GRAZ

One-round 3-party key exchange

Alice, Bob and Carl want to agree on a common secret key.

1. $G_1 = \langle P \rangle$ and G_2 cyclic groups; pairing $\varepsilon : G_1 \times G_1 \to G_2$ (publicly known)
2. personal secret keys: a, b, c
3. Alice sends aP to Bob and Carl
4. Bob sends bP to Alice and Carl
5. Carl sends cP to Alice and Bob
6. Alice computes $\varepsilon(bP, cP)^a$
7. Bob computes $\varepsilon(aP, cP)^b$
8. Carl computes $\varepsilon(aP, bP)^c$
9. the common secret key is $\varepsilon(P, P)^{abc}$

Security relies on the Bilinear Diffie-Hellman assumption:
it is *hard* to find $\varepsilon(P, P)^{abc}$ given P, aP, bP, cP.

State of the art

- Elliptic curve cryptography (ECC):
  - proposed independently by Koblitz and Miller in 1985
  - extensively studied
  - standardised cryptographic protocols
  - commercial applications

State of the art

- Elliptic curve cryptography (ECC):
  - proposed independently by Koblitz and Miller in 1985
  - extensively studied
  - standardised cryptographic protocols
  - commercial applications
- Hyperelliptic curve cryptography (HECC):
  - proposed by Koblitz in 1989
  - still under (theoretical) investigation
  - no real-world applications yet

State of the art

- Elliptic curve cryptography (ECC):
  - proposed independently by Koblitz and Miller in 1985
  - extensively studied
  - standardised cryptographic protocols
  - commercial applications
- Hyperelliptic curve cryptography (HECC):
  - proposed by Koblitz in 1989
  - still under (theoretical) investigation
  - no real-world applications yet
- Pairing-based cryptography:
  - initially used for cryptanalisys against supersingular elliptic curves (MOV attack, 1993; Frey-Rück attack, 1994)
  - rediscovered for "good" use by Joux in 2000, and Boneh-Franklin in 2001

Hyperelliptic curves

Let $\mathbb{F}_q$ be a finite field with $q = p^n$ elements.
A hyperelliptic curve $H/\mathbb{F}_q$ of genus $g \geq 1$ is a *non-singular* algebraic curve

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

where

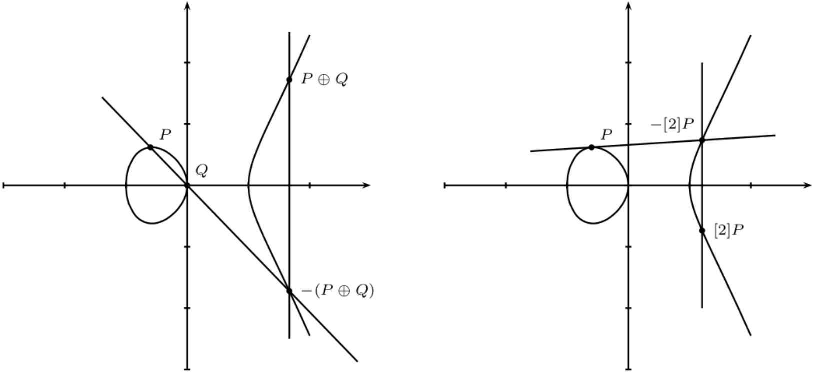
- $h(x),\ f(x) \in \mathbb{F}_q[x]$
- $f(x)$ is monic
- $\deg(f) = 2g + 1$
- $\deg(h) \leq g$

H has only one point at infinity $\infty = [0 : 1 : 0]$

For $g = 1$, H is an elliptic curve.

Arithmetic on elliptic curves

We can define the sum of points of H with the chord-tangent rule:



$H(\mathbb{F}_q)$ is a finite Abelian group, with neutral element ∞.

Divisors of a hyperelliptic curve

A divisor is a formal finite sum of points of H:

$$D = \sum_{i=1}^{d} m_i P_i \quad \text{with} \quad m_i \in \mathbb{Z}, \qquad \deg(D) = \sum_{i=1}^{d} m_i$$

The set of divisors of H is an additive group.

A principal divisor is

$$\operatorname{div}(F) = \sum_{P \in H} \operatorname{ord}_F(P) P - \left(\sum_{P \in H} \operatorname{ord}_F(P) \right) \infty$$

for any rational function $F(x, y)$ on H.

Let Div^0 be the subgroup of divisors of degree 0 and $\mathcal{P}$ the subgroup of principal divisors.
The Jacobian of H is $J = Div^0 / \mathcal{P}$.

Canonical representation of divisor classes

If we consider only divisors fixed by the Galois group of $\mathbb{F}_q$, then the Jacobian $J(\mathbb{F}_q)$ is a finite Abelian group.

Every divisor class of $J(\mathbb{F}_q)$ can be represented by a unique pair of polynomials $a(x)$, $b(x) \in \mathbb{F}_q[x]$ s.t.

- $a(x)$ is monic
- $\deg(b) < \deg(a) \leq g$
- $a(x) \mid b(x)^2 + h(x)b(x) - f(x)$

Addition in $J(\mathbb{F}_q)$ can be performed via polynomial arithmetic [Cantor's algorithm, 1987]:

- $D_1 + D_2 \approx 17g^2 + O(g)$ field operations
- $2D \approx 16g^2 + O(g)$ field operations

Security requirements

There are some security requirements for $J(\mathbb{F}_q)$ to be suitable for cryptographic applications:

- $g < 4$
- H must be *not* supersingular (except for pairing-based crypto)
- $|J(\mathbb{F}_q)|$ must have a large prime factor
- other conditions on $|J(\mathbb{F}_q)|$ to be resistant to all known attacks.

$H/\mathbb{F}_q$ is supersingular if there are no divisors of order p in $J(\mathbb{F}_{q^m})$ for any $m \geq 1$.

Computational problems

1. divisor class counting, i.e. find the order of $J(\mathbb{F}_q)$
2. supersingularity criteria
3. scalar multiplication, i.e. compute $nD = D + \cdots + D$ for $n \in \mathbb{Z}$, $D \in J(\mathbb{F}_q)$ in an efficient way
4. pairing computation

Frobenius endomorphism

The Frobenius endomorphism of $H/\mathbb{F}_q$ is

$$\tau(x, y) = (x^q, y^q)$$

and has characteristic polynomial

$$\chi(x) = x^{2g} + a_1 x^{2g-1} + \cdots + a_g x^g + a_{g-1} q x^{g-1} + \cdots + a_1 q^{g-1} x + q^g$$

Important: $|J(\mathbb{F}_q)| = \chi(1)$

$\chi(x)$ can be found by counting points on H:

$$\begin{aligned} M_k &= |H(\mathbb{F}_{q^k})| \\ a_k &= \frac{1}{k}\left(M_k - q^k - 1 + \sum_{i=1}^{k-1}(M_{k-i} - q^{k-i} - 1)a_i\right) \end{aligned}$$

Point counting on elliptic curves - I

$E/\mathbb{F}_q : y^2 = f(x)$. By Hasse theorem:

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

Frobenius characteristic polynomial: $\chi(x) = x^2 + a_1 x + q$

$$|E(\mathbb{F}_q)| = q + 1 - a_1$$
$$|a_1| \leq 2\sqrt{q}$$

Finding $|E(\mathbb{F}_q)|$ is equivalent to find a_1

Naive approach: compute the Legendre symbols

$$|a_1| = \sum_{x \in \mathbb{F}_q} \left(\frac{f(x)}{q} \right)$$

It takes $O(q \log q)$ $\rightsquigarrow$ exponential!

Point counting on elliptic curves - II

Schoof's algorithm [1985]:

1. compute a_1 modulo p for many small primes p such that $\Pi p \geq 4\sqrt{q}$
2. find a_1 with the Chinese Remainder Theorem

ALPEN-ADRIA UNIVERSITÄT KLAGENFURT I WIEN GRAZ

Point counting on elliptic curves - II

Schoof's algorithm [1985]:

1. compute a_1 modulo p for many small primes p such that $\prod p \geq 4\sqrt{q}$
2. find a_1 with the Chinese Remainder Theorem

- can compute $|E(\mathbb{F}_q)|$ in deterministic polynomial time $O(\log^8 q)$
- SEA algorithm: restrict the set of primes $\rightsquigarrow$ $O(\log^4 q)$ probabilistic
  (e.g. SEA is implemented in PARI/GP)
- there exist (in theory) polynomial-time SEA-like algorithms for hyperelliptic curves, but they are difficult to implement
- there is a practical algorithm only for $g = 2$ [Gaudry-Harley 2000]

ALPEN-ADRIA UNIVERSITÄT KLAGENFURT | WIEN GRAZ

Supersingularity

Point counting on hyperelliptic curves is important

- to find Frobenius characteristic polynomial $\chi(x)$
- to determine the order of the Jacobian $|J(\mathbb{F}_q)|$

Supersingularity

Point counting on hyperelliptic curves is important

- to find Frobenius characteristic polynomial $\chi(x)$
- to determine the order of the Jacobian $|J(\mathbb{F}_q)|$

...but also to tell whether a curve is supersingular or not.

Stichtenoth-Xing criterion [1995]:

$$H/\mathbb{F}_q \text{ supersingular } \Leftrightarrow \ a_k \equiv 0 \mod p^{\lceil \frac{kn}{2} \rceil} \ \forall k = 1 \dots g$$

($a_1, \dots, a_g$ are the coefficients of $\chi(x)$ and $q = p^n$)

Scalar multiplication - I

$H/\mathbb{F}_q$ and $D \in J(\mathbb{F}_{q^m})$, compute nD for $n \in \mathbb{Z}$, $n > 0$

Standard method: use binary expansion of n

$$n = \sum_{i=0}^{L} d_i 2^i, \quad d_i \in \{0,1\}$$

$$nD = d_0 D + 2(d_1 D + 2(d_2 D + \cdots + d_L D))$$

\# divisor doublings $\approx$ length of the expansion
\# divisor additions $\approx$ weight of the expansion

Scalar multiplication - II

$\tau(x,y) = (x^q, y^q)$ induces an endomorphism on $J(\mathbb{F}_{q^m})$:

$$\tau\left([a(x), b(x)]\right) = \left[a^{(q)}(x), b^{(q)}(x)\right]$$

which requires at most $2g$ q-th powers (i.e. cyclic shifts) in $\mathbb{F}_{q^m}$

Idea: represent integers to the basis τ

$$n = \sum_{i=0}^{L} d_i \tau^i$$

$$nD = d_0 D + \tau(d_1 D + \tau(d_2 D + \cdots + d_L D))$$

$\#$ evaluations of $\tau \approx$ length of the expansion
$\#$ divisor additions $\approx$ weight of the expansion
plus some precomputation $(d_i D)$

Scalar multiplication - III

Improvements:

- reduce the number of divisor additions by using a w-NAF expansion, i.e. in every block of w consecutive digits there is at most one non-zero digit
- reduce the precomputation effort by means of symmetric digit sets.

Questions:

- existence of a *finite* τ-adic expansion for every integer?
- average weight of the expansion?
- length of the expansion?
- practical recoding algorithm?

Grazie per l'attenzione!