An introduction to Hyperelliptic Cryptography

Stefania Vanzetti
BUNNYTN 2011

Università degli Studi di Trento

12 settembre 2011

1.HYPERELLIPTIC CURVES

Why are hyperelliptic curves important?

Hyperelliptic curves are suitable for cryptography:

- They can be seen as an alternative to elliptic curves
- We can associate a group structure to every hyperelliptic curve
- We can use cryptographic protocols that rely on the difficulty of solving discrete logarithm problem

however

- the group operation is more difficult
- We can't use every hyperelliptic curve for our purpose, we have to choose them paying attention to their genus, the order of the group and their field of definition.

Basic definitions

Definition (Hyperelliptic curve)

A hyperelliptic curve $\mathbf{C}$ over a finite field $\mathbb{F}_q$ of genus g is the set of points in $\mathbb{F}_q \times \mathbb{F}_q$ such that:

$$\mathbf{C}: \; y^2 + h(x)y = f(x)$$

where

1. $h(x) \in \mathbb{F}_q[x]$ is a polynomial of degree less or equal than g
2. $f(x) \in \mathbb{F}_q[x]$ is a polynomial of degree $2g + 1$
3. $\mathbf{C}$ doesn't have any singular point over $\overline{\mathbb{F}}_q \times \overline{\mathbb{F}}_q$.

Nota

Elliptic curves are hyperelliptic curves of genus $g = 1$

2.JACOBIAN GROUP

Divisors

Given a curve $\mathbf{C}$, we can define formal sums of points of C called *divisors*:

$$D = \sum m_P P \quad con\ m_P \in \mathbb{Z},\ P \in \mathbf{C}$$

Divisors form an additive abelian group $Div(\mathbf{C})$:

$$\sum m_P P + \sum n_P P = \sum (m_P + n_P) P$$

The sum $\sum m_P$ is the *degree* of D. The set of degree 0 divisors is a subgroup $Div(\mathbf{C})^0$ of $Div(\mathbf{C})$.

Function field

The *coordinate ring* of $\mathbf{C}$ over $\mathbb{F}_q$, denoted with $\mathbb{F}_q[\mathbf{C}]$ is the quotient ring

$$\mathbb{F}_q[\mathbf{C}] = \mathbb{F}_q[x, y] \Big/ (y^2 + h(x)y - f(x)),$$

where $(y^2 + h(x)y - f(x))$ denote the ideal generated by the irreducible polynomial $y^2 + h(x)y - f(x)$.

The *function field* is field of fractions of $\mathbb{F}_q[\mathbf{C}]$:

$$\mathbb{F}_q(\mathbf{C}) = Frac\left(\mathbb{F}_q[x, y] \Big/ (y^2 + h(x)y - f(x))\right)$$

Function field

Theorem

We can associate a divisor to a rational function ϕ over the projective closure of $\mathbf{C}$ *(formal sum of affine points of* $\mathbf{C}$ *together with* $P_\infty = [0 : 1 : 0]$*)*

A divisor of rational functions is always a degree 0 divisor, called *principal*.
The set of principal divisors $\mathbf{P}$ is a subgroup of $Div(\mathbf{C})^0$.

Jacobian

Definition (Jacobian of $\mathbf{C}$)

The Jacobian of $\mathbf{C}$ is the quotient group

$$J(\mathbf{C}) = Div(\mathbf{C})^0/\mathbf{P}$$

Hence $D_1, D_2 \in Div(\mathbf{C})^0$ are equivalent if $D_1 - D_2 \in \mathbf{P}$.

In every equivalence class there's only one divisor D, called *reduced divisor*:

$$D = \sum m_P P - (\sum m_P) P_\infty$$

such that $\sum m_P \leq g$.

HCC

The Jacobian $J(\mathbf{C})$ is an abelian group with the operation of sum between divisors. So it can be used to implement Hyperelliptic Curves Cryptosystems (HCC).

We can use traditional discrete logarithm cryptosystems, such as:

1. Diffie Hellman (key exchange)
2. ElGamal (encryption)
3. ElGamal (digital signature)

3.SUM BETWEEN DIVISORS

Mumford representation

We have to find another way to express divisors in order to be able to implement the sum operation.

We can associate a couple of polynomials $(a(x), b(x))$ belonging to $\mathbb{F}_q[x]$ to every divisor, where:

- $deg(b) < deg(a) \leq g$
- a monic
- $a | b^2 + bh - f$

The last condition implies that there's only one element of $J(\mathbf{C})$ represented by this couple of polynomials:

$$D = MCD(div(a(x)), div(b(x) - y))$$

We will use the notation $D = div(a, b)$.

Cantor's Algorithm

This algorithm consists of two parts:

Composition: Receives two divisors representation $D_1 = div(a_1, b_1)$ and $D_2 = div(a_2, b_2)$ and computes the divisor $D \sim D_1 + D_2$. (If D_1, D_2 are reduced, then $deg(D) \leq 2g$). The two most expensive steps use the Extended Euclidean Algorithm to find the GCD of two polynomials and the coefficients of Bezout's Identity.
Reduction: Receives a divisor $D = div(a, b)$ and computes a reduced divisor $D' \sim D$: the canonical representative belonging to the same equivalence class of D.
Every time the reduction procedure is repeated, the degree of the divisor D decreases by 2. So we will need at most $\lceil \frac{g}{2} \rceil$ steps.

In this case $deg(D) = \sum m_P$.

NUCOMP

Idea: We want to merge the steps of compositions and reduction: this let us work with polynomials of smaller degree than before, so the final reduction is less expensive (at the end of the main body of the algorithm, if the divisor isn't reduced, we will need at most two more steps of reduction).

Result: the complexity of the two algorithms is asymptotically the same, but heuristically NUCOMP in most cases needs less operations.

Scalar multiplication

Most of cryptographic protocols need to compute a multiple of a divisor. The most natural thing to do is use the sum algorithm repeatedly.

There are special versions of previous algorithms that improve the speed of scalar multiplication:

1. with Cantor's algorithm we get high degree divisors, so it's necessary to improve the reduction step
2. NUDPL: it's a version of NUCOMP that works better for duplication.

What algorithm?

The choice depends on the genus of the curve:

- $g = 2$ and $g = 3$ there are explicit formulas. If $g = 2$ they need 25 multiplications and one inversion for the sum and 27 multiplications and one inversion for the duplication.
- $g < 10$ Cantor
- $g \geq 10$ NUCOMP/NUDPL

4.DISCRETE LOGARITHM

Classic attacks

HCDLP is a generalization of ECDLP, every attack that works for elliptic curves can be used in our case as well:

- Pohlig-Hellman: solving the HCDLP is as difficult as solving DLP in the biggest subgroup with prime order of the Jacobian.
- MOV/Frey-Rück: reduces the HCDLP to a DLP over the multiplicative group of a finite field.
- Pollard's ρ: $O(q^{g/2})$ group operations to solve HCDLP in a soubgroup of order q of the Jacobian.
- Baby-Step-Giant-Step: $O(q^{g/2})$ group operations, but needs more memory than Pollard's ρ

Index Calculus

Idea: If we are able to compute the discrete logarithm for some simple elements of the group, the we can compute the discrete logarithm of those elements which can be written as a combination of the first ones.

Suppose that we want to compute k s.t. $b = a^k$, $a, b \in G$ finite abelian group, the algorithm consists of the following:

1. Choice of a factor base $B = \{g_1, \ldots, g_n\}$: set of simple elements of the set.
2. Search for relations between the factor base elements and the index a.
3. If we have enough relations, we can compute the discrete logarithm of $g_1, \ldots, g_n$ by linear algebra
4. If we can write a power of b as a combination of powers of $g_1, \ldots, g_n$, we can compute k.

Index Calculus

Let $\mathbb{F}_p^*$ the multiplicative group of the finite field $\mathbb{F}_p$, generated by a primitive root a. Given $b \in \mathbb{F}_p^*$, we want to find $log_a b$.

1. Fix a subset $B = \{p_1, ..., p_j\} \subseteq \mathbb{F}_p^*$ of all the primes smaller than a bound.
2. Find $j + t$ relations between the elements of B such as:

$$\prod_j p_j^{r_{i,j}} \equiv a^{r_i} \ mod p \quad \text{obtaining} \quad \prod_j a^{log_a(p_j) r_{i,j}} \equiv a^{r_i} \ mod p$$

which can be written as:

$$\sum_j log_a(p_j) r_{i,j} \equiv r_i \ mod(p-1)$$

Index Calculus

3. Compute the discrete logarithm $log_a(p_1), \ldots, log_a(p_j)$ of the primes in B solving the system:

$$\begin{cases} \sum_j log_a(p_j) r_{1,j} \equiv r_1 \ mod(p-1) \\ \vdots \\ \sum_j log_a(p_j) r_{j+t,j} \equiv r_{j+t} \ mod(p-1) \end{cases}$$

4. If $b \in G$ is $B-$smooth, there exist $k, k_i \in \mathbb{N}$ such that $b^k = p_1^{k_1} \ldots p_j^{k_j}$, we can find the logarithm of b from:

$$k \cdot log_a(b) = k_1 \cdot log_a(p_1) + \cdots + k_j \cdot log_a(p_j)$$

Index Calculus

Observations:

- The first 3 steps doesn't depend on b, so we can use this part of *precomputation* even to compute discrete logarithm of other elements of the group.
- The algorithm can be modified, including b in the search for relations, so we can compute k without finding $log_a g_1, \ldots, log_a g_n$.
- Index Calculus is a probabilistic method. If b isn't B-smooth we can't deduce $log_a b$ from $log_a g_1, \ldots, log_a g_n$.

Index Calculus v. Pollard's rho

Index-calulus has been adapted to the case of hyperelliptic curves, but we have an additive group and we have to find relations between divisors.

Efficiency will depend on the genus an the choice of the factor base. In the table there's a comparison between the Gaudry/Enge version of the Index Calculus and Pollard's rho. (q is the group order)

g	2	3	4	5	6
rho	q	$q^{3/2}$	q^2	$q^{5/2}$	q^3
Gaudry	$q^{4/3}$	$q^{3/2}$	$q^{8/5}$	$q^{5/3}$	$q^{12/7}$

If $g = 2$ Pollard's ρ is better. We have to use curves of genus $g \leq 2$ or curves with a Jacobian large enough to make Index Calculus inefficient (longer keys).

Jacobian order

We need to know the Jacobian order both to implement cryptographic systems and to solve the HCDLP with previous methods

Let us consider an hyperelliptic curve C of genus g defined over the finite field $\mathbb{F}_q$, with q power of a prime. The Jacobian order would lie in the Hasse-Weil's inteval:

$$(\sqrt{q}^k - 1)^{2g} \leq \#J(\mathbb{F}_{q^k}) \leq (\sqrt{q}^k + 1)^{2g}$$

Jacobian order

The Jacobian order can be computed using the Zeta function $Z_C(t)$.

Let C be defined over $\mathbb{F}_q$, let M_n be the number of points $\mathbb{F}_{q^n}$-rational over C.

The Zeta function C is the power series

$$Z_C(t) = exp\left(\sum_{r\geq 1} M_r \frac{t^r}{r}\right)$$

Jacobian order

The function $Z_C(t)$ can be written as:

$$Z_C(t) = \frac{P(t)}{(1-t)(1-qt)}$$

where $P(t)$ is a polynomial of degree $2g$ with integer coefficients that can be factored as:

$$P(t) = \prod_{i=1}^{g}(1-\alpha_i t)(1-\overline{\alpha_i} t).$$

The number of points in the Jacobian $J(\mathbb{F}_{q^n})$ will be:

$$\#J(\mathbb{F}_{q^n}) = \prod_{i=1}^{g} |1-\alpha_i^n|^2$$

5.SCELTA DELLA CURVA

Choice of the curve

Some selection criteria:

1. Curves over a field K of characteristic 2 (sum can be computed efficiently)
2. $\#J_C$ divisible by a big prime r (Pohlig-Hellman, Pollard's ρ).
3. If we have a curve over $\mathbb{F}_q$, Frey-Rück's method let us create an isomorphism between $J(C)$ and the multiplicative group of $\mathbb{F}_{q^k}$, so we have to check that r doesn't divide $q^k - 1$ for these k such that the DLP can be solved easily over $\mathbb{F}_{q^k}$.
4. If $g = 2$ the Index Calculus method is less useful than others and the Jacobian order can be computed easier.

Choice of the curve

Instead of considering a generic hyperelliptic curve and calculate the order of the Jacobian to verify whether it is suitable or not, we can do the opposite.

There are methods to choose a ″secure″ Jacobian and find later a curve with the desired Jacobian order.

Nota

If $g = 2$ these methods are quite efficient, they become more and more difficult with genus increasing.

Grazie per l'attenzione!