Discrete logarithm algorithms in pairing-relevant finite fields

Gabrielle De Micheli

Joint work with Pierrick Gaudry and Cécile Pierrot

Université de Lorraine, Inria Nancy, France

Journées C2, Fall 2020
Virtual Conference

The discrete logarithm problem (DLP)

Asymmetric cryptography relies on the hardness of either factorization (RSA) or the **discrete logarithm problem.**

→ Used in Diffie-Hellman, El-Gamal, (EC)DSA, etc

Definition

Given a finite cyclic group G, a generator $g \in G$ and a target $h \in G$, find x such that $h = g^x$.

Commonly used groups: prime finite fields $\mathbb{F}_p^* = (\mathbb{Z}/p\mathbb{Z})^*$, finite fields $\mathbb{F}_{p^n}^*$, elliptic curves over finite fields $\mathcal{E}(\mathbb{F}_p)$ ...

Groups G for which DLP is hard

Examples in the wild

Widely deployed protocols base their security on the hardness of DLP on a group G.

Ephemeral Diffie Hellman

Technical Details

Connection Encrypted (TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, 128 bit keys, TLS 1.2)

An interesting example: pairing-based protocols!

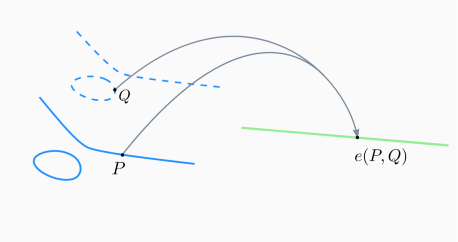


Fig from Diego Aranha

Pairing-based cryptography

What is a cryptographic pairing ?

- $\mathbb{G}_1, \mathbb{G}_2$: additive groups of prime order ℓ.
- $\mathbb{G}_T$: multiplicative group of prime order ℓ.

A pairing is a map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$

- with bilinearity: $\forall a, b \in \mathbb{Z}, e(aP, bQ) = e(P, Q)^{ab}$,
- non-degeneracy: $\exists P, Q$ such that $e(P, Q) \neq 1$,
- and such that e is efficiently computable (for practicality reasons).

Called symmetric if $\mathbb{G}_1 = \mathbb{G}_2$.

Security of pairing-based protocols

Most of the time, in cryptography:

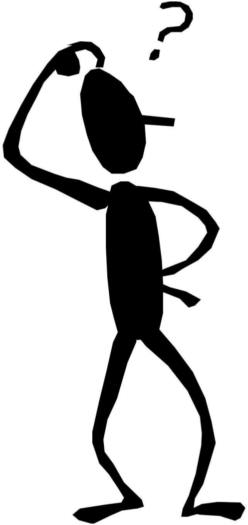
- $\mathbb{G}_1$ = subgroup of $\mathcal{E}(\mathbb{F}_p)$,
- $\mathbb{G}_2$ = subgroup of $\mathcal{E}(\mathbb{F}_{p^n})$,
- $\mathbb{G}_T$ = subgroup of finite field $\mathbb{F}_{p^n}^*$.

Why do we care ? hundreds of old and many recent protocols built with pairings.
Example: zk-SNARKS (blockchain, Zcash ...)

⟶ Example that uses DLP on both elliptic curves and finite fields.

Question: **How to construct a secure pairing-based protocol ?**
⟶ Look at DLP algorithms on both sides!

The discrete logarithm problem in finite fields



- Many different algorithms for DLP in $\mathbb{F}_{p^n}$
- Their complexity depends on the relation between characteristic p and extension degree n.

Useful notation

→ Complexity depends on the relation between characteristics p and extension degree n.

L-notation:

$$L_{p^n}(l_p, c) = \exp((c + o(1))(\log(p^n))^{l_p}(\log\log p^n)^{1-l_p}),$$

for $0 \leqslant l_p \leqslant 1$ and some constant $c > 0$.

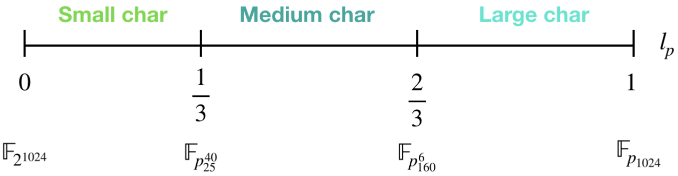
For complexities:

- When $l_p \to 0$: $\exp\left(\log\log p^n\right) \approx \log p^n$ Polynomial-time
- When $l_p \to 1$: p^n Exponential-time

In the middle, we talk about subexponential time.

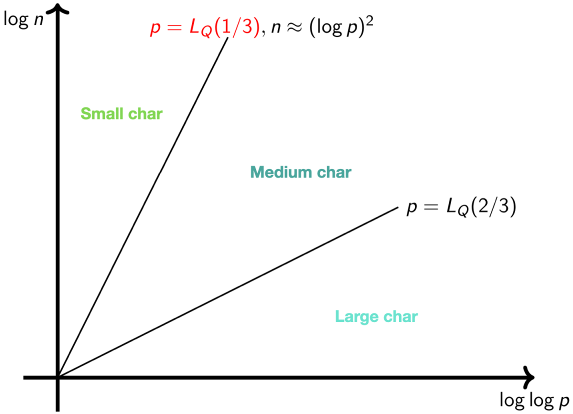
Three families of finite fields

Finite field: $\mathbb{F}_{p^n}$, with $p = L_{p^n}(l_p, c_p)$



- Different algorithms are used in the different zones.
- Algorithms don't have the same complexity in each zone.

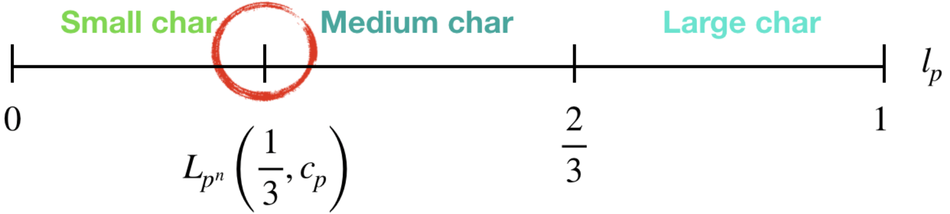
The L-notation for $\mathbb{F}_Q$ with $Q = p^n$



Question: Which area do we focus on ?

The first boundary case

In this work, we focus on the boundary case $p = L_{p^n}(1/3)$, the area between the small and the medium characteristics.



Why?

1. Area where pairings take their values.
2. Many algorithms overlap: $\longrightarrow$ which algorithm has the lowest complexity ?

Balancing complexities for the security of pairings

Idea: For pairings, we want DLP to be as hard on the elliptic curve side than on the finite field side.

- choose the area where DLP in finite fields is the most difficult;

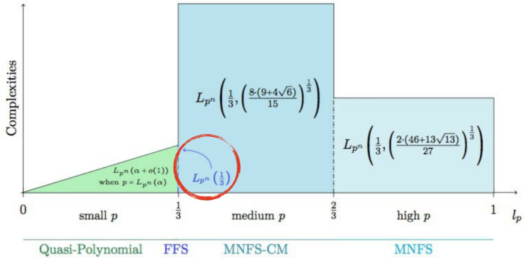
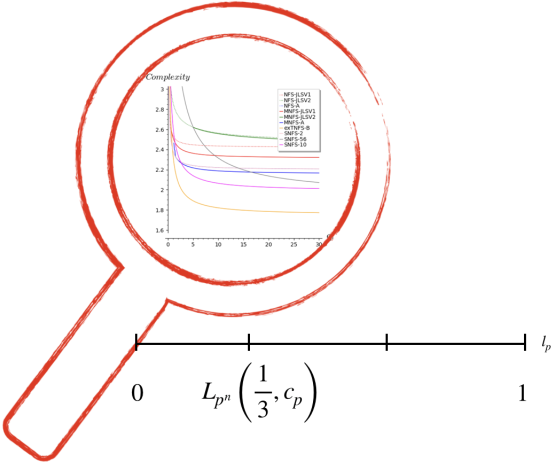


Fig. Cécile Pierrot

- "balance" complexity on elliptic curves and finite fields:

$$\sqrt{p} = L_{p^n}(1/3) \Rightarrow p = L_{p^n}(1/3)$$

Main results of the paper



- Analyse the behaviour of many algorithms in this area.
- Estimate the security of pairing-based protocols.

Consider a finite field $\mathbb{F}_{p^n}$.
Factor basis: $\mathcal{F}$ = small set of " small " elements.
Three main steps:

1. Relation collection: find relations between the elements of $\mathcal{F}$.
2. Linear algebra: solve a system of linear equations where the unknowns are the discrete logarithms of the elements of $\mathcal{F}$.
3. Individual logarithm: for a target element $h \in \mathbb{F}_{p^n}$, compute the discrete logarithm of h.

The Number Field Sieve and its variants are examples of *index calculus algorithms.*

The complexity of NFS and its variants

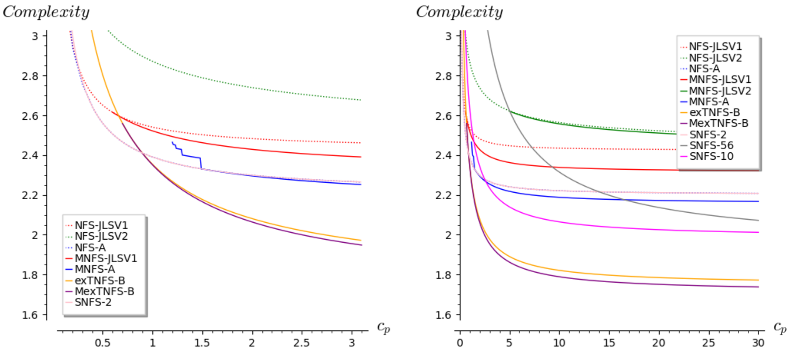
- 3 phases = 3 costs $\longrightarrow$ overall complexity is sum of 3 costs.

Goal: Optimize the maximum of these three costs.

Why complicated?

1. Many parameters $\longrightarrow$ discrete or continuous, boundary issues.
2. Optimization problem $\longrightarrow$ Lagrange multipliers.
3. Solving a polynomial system $\longrightarrow$ Gröbner basis algorithm.
4. Uses many analytic number theory results.

A summary of these complexities

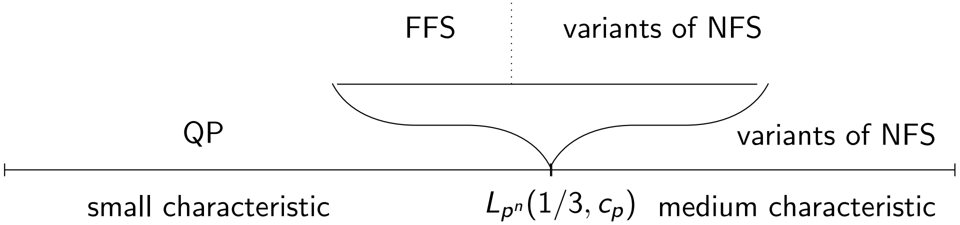


Recall $p = L_{p^n}(1/3, c_p)$, and complexities $= L_{p^n}(1/3, f(c_p))$

Surprising fact:

- Not all the variants are applicable at the boundary case: STNFS has a much higher complexity!

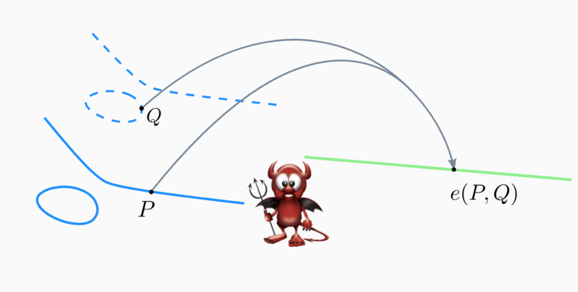
And the winners are ... !



For the variants of NFS, the best algorithm depends on considerations on n and p.

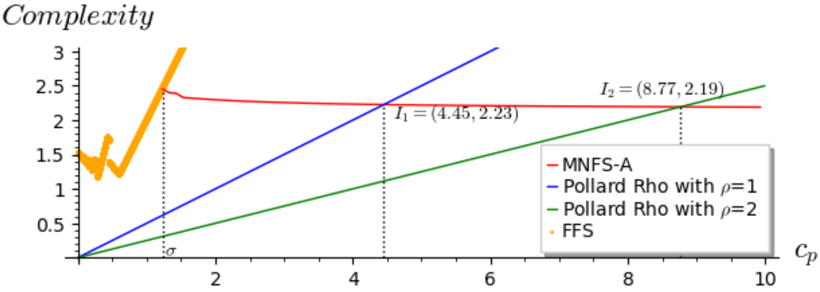
Constructing secure pairings

Asymptotically what finite field $\mathbb{F}_{p^n}$ should be considered in order to achieve the highest level of security when constructing a pairing?



Goal: find the optimal p and n that answers this question.

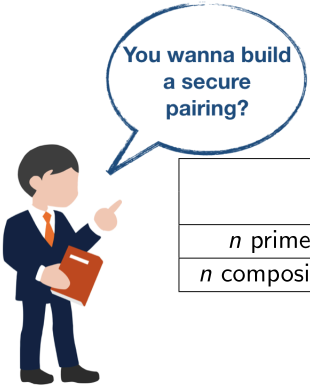
Goal: Look for value of c_p that maximizes $\min(\text{comp}_{\mathcal{E}}, \text{comp}_{\mathbb{F}_{p^n}})$.



- Complexities for finite field DLP are decreasing functions.
- Pollard rho is an increasing function ($\text{complexity}_{\mathcal{E}} = p^{1/2\rho}$)

→ optimal c_p given by the intersection point!

Conclusion for pairings



	normal p	special p $\lambda = 20$	special p $\lambda = 3$
n prime	$c_p = 4.45, \; c_{\textbf{MNFS-}\mathcal{A}} = 2.23$		$c_p = 4.36, \; c_{\textbf{SNFS-3}} = 2.18$
n composite	$c_p = 3.91, \; c_{\textbf{MexTNFS-B}} = 1.91$		

Suprising fact: Using a special form for p does not always make the pairing less secure ! It depends on the value of λ.

Thank you for your attention!

Questions?