Discrete logarithm algorithms in pairing-relevant finite fields

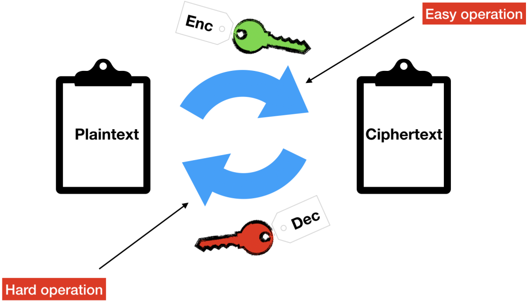
Gabrielle De Micheli

Joint work with Pierrick Gaudry and Cécile Pierrot

Université de Lorraine, Inria Nancy, France

February 26th, 2020
Northeastern University, Boston

Asymmetric cryptography



Relies on the hardness of two main mathematical problems:

- Factorization (RSA cryptosystem)
- Discrete logarithm problem

The discrete logarithm problem (DLP)

→ 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 $h = g^x$.

Which group G should we consider ?

Groups for DLP

In cryptography, choose G such as DLP is difficult:

- prime finite fields $\mathbb{F}_p^* = (\mathbb{Z}/p\mathbb{Z})^*$,
- class groups of number fields,
- finite fields $\mathbb{F}_{p^n}^*$,
- elliptic curves over finite fields $\mathcal{E}(\mathbb{F}_p)$,
- genus 2 hyperelliptic curves.

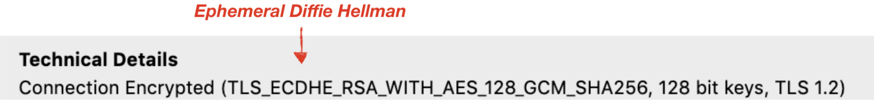
One bad idea: $(\mathbb{Z}/N\mathbb{Z}, +)$ where DLP is simply a division.

Classical assumptions:

- The order of the group is known.
- There exists an efficient algorithm for the group law.

Examples in the wild

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



An interesting example: pairing-based protocols!

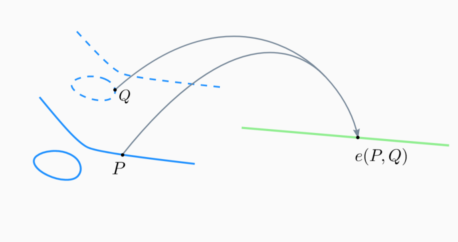


Fig from Diego Aranha

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:

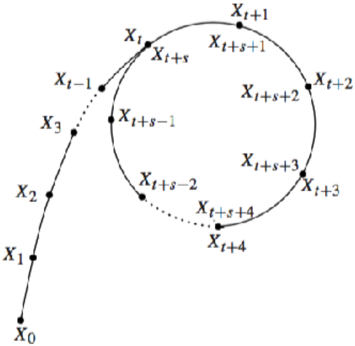
- $\mathbb{G}_1 =$ subgroup of $\mathcal{E}(\mathbb{F}_p)$,
- $\mathbb{G}_2 =$ subgroup of $\mathcal{E}(\mathbb{F}_{p^n})$,
- $\mathbb{G}_T =$ subgroup of finite field $\mathbb{F}_{p^n}^*$.

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

$\longrightarrow$ Example that uses DLP on both elliptic curves and finite fields.

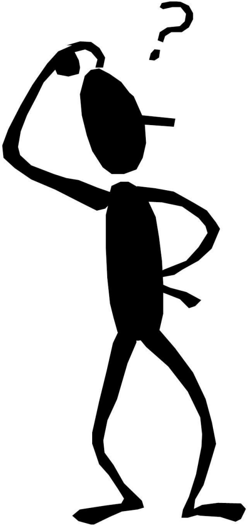
Question: **How to construct a secure pairing-based protocol ?**
$\longrightarrow$ Look at DLP algorithms on both sides!

The discrete logarithm problem on elliptic curves



- Best algorithm: Pollard Rho
- Complexity: square root of the size of the subgroup considered.
- No gain except for constant factor since the 70s.

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

$\longrightarrow$ Complexity depends on the relation between characteristics p and extension degree n.

L-notation:

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

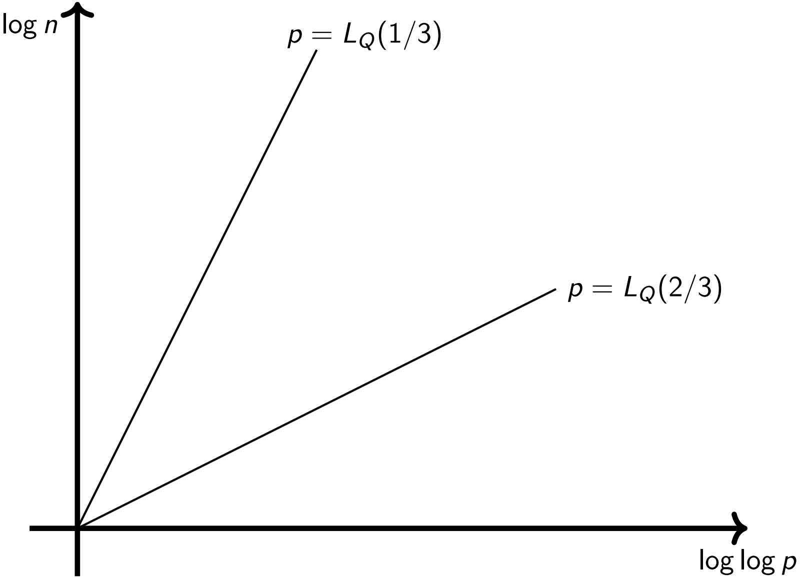
for $0 \leqslant l_p \leqslant 1$ and some constant $c > 0$.

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

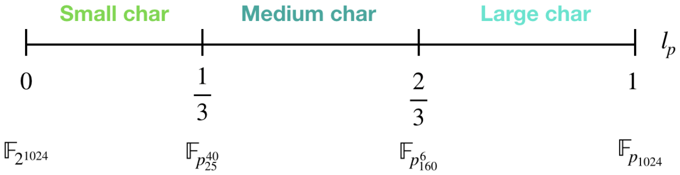
In the middle, we talk about subexponential time.

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



Three families of finite fields

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

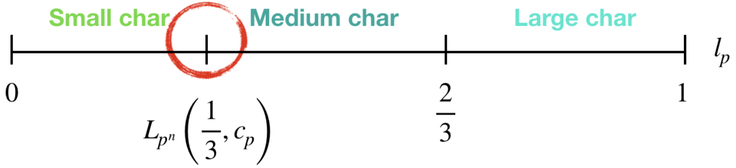


- Different algorithms are used in the different zones.
- Algorithms don't have the same complexity in each zone.

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;

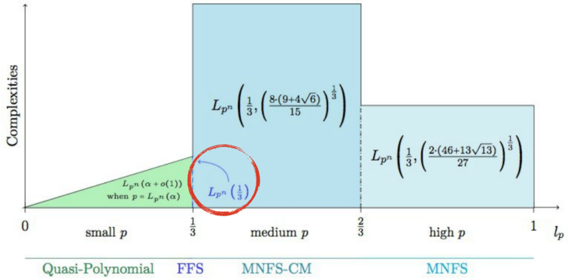
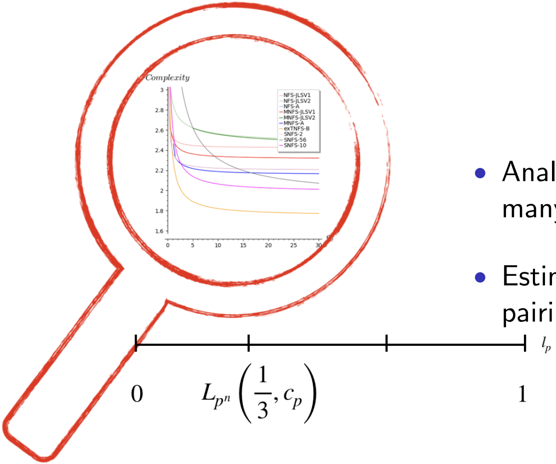


Fig. Cécile Pierrot

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

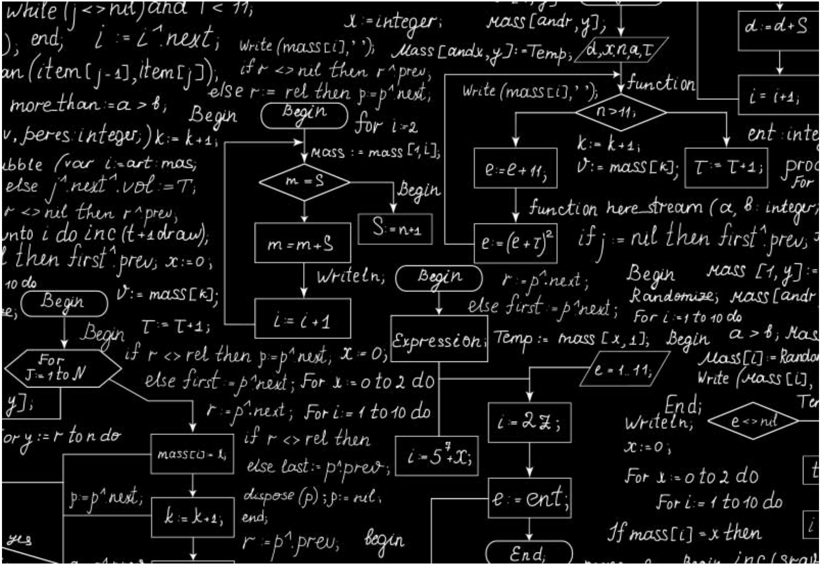
$$\boxed{\sqrt{p} = L_{p^n}(1/3) \Rightarrow p = L_{p^n}(1/3)}$$

The road ahead



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

Index Calculus Algorithms



The index calculus algorithms

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

$$\begin{tikzcd}[column sep=small]
 & & a - bx \quad \mathbb{Z}[X] \arrow[dl, "X \mapsto \theta_1"'] \arrow[dr, "X \mapsto \theta_2"] & & \\
\begin{array}{c} a - b\theta_1 \\ N_1(a - b\theta_1) \\ \textbf{B-smooth?} \end{array} & \mathbb{Q}[X]/(f_1(X)) \arrow[dr, "\theta_1 \mapsto m"'] & & \mathbb{Q}[X]/(f_2(X)) \arrow[dl, "\theta_2 \mapsto m"] & \begin{array}{c} a - b\theta_2 \\ N_2(a - b\theta_2) \\ \textbf{B-smooth?} \end{array} \\
 & & \mathbb{F}_{p^n} & &
\end{tikzcd}$$

1. f_1, f_2 irreducible in $\mathbb{Z}[X]$ s.t. the diagram commutes.
2. Compute the algebraic norms in $\mathbb{Z}$: $N(a - b\theta_i)$
3. Factor $N_i(a - b\theta_i)$ in $\mathbb{Z}$ into prime numbers
4. If prime factors $\leqslant B$ on both sides $\longrightarrow$ relation

Collecting relations, solving a system...

A relation in $\mathbb{F}_{p^n}$ implies the equality:

$$a - b\theta_1 \text{``} = \text{''} \prod_{f \in \mathcal{F}} f^{\alpha_i} \equiv \prod_{f \in \mathcal{F}} f^{\beta_i} \text{``} = \text{''} a - b\theta_2 .$$

Take the discrete logarithm on both sides:

$$\sum_{f \in \mathcal{F}} \alpha_i \log f = \sum_{f \in \mathcal{F}} \beta_i \log f \pmod{p^n - 1}$$

$=$ **linear relation between log elements of the factor basis $\mathcal{F}$.**

Goal: Get as many equations/relations of log of elements of the factor basis.

Why? we want to solve a linear system!

Solving the linear system and a descent phase

Linear algebra:

- unknowns are the $\log f$ for $f \in \mathcal{F}$.
- solve the system to recover the values $\log f$.

How do we solve the system? Sparse linear algebra algorithms : block Wiedemann algorithm in $O(k^2)$, where k is the size of the system.

Descent phase: our target is $h \in \mathbb{F}_{p^n}$. Find $\log h$.

A few variants...

The Multiple NFS

Considering multiple number fields.

$$\begin{tikzcd}
& & & \mathbb{Z}[X] \arrow[dlll] \arrow[dll] \arrow[d, "X \mapsto \theta_i"] \arrow[drr] \arrow[drrr] & & & \\
\mathbb{Q}(\theta_1) \arrow[drrr] & \mathbb{Q}(\theta_2) \arrow[drr] & \cdots & \mathbb{Q}(\theta_i) \arrow[d, "\theta_i \mapsto m"] & \cdots & \mathbb{Q}(\theta_{V-1}) \arrow[dll] & \mathbb{Q}(\theta_V) \arrow[dlll] \\
& & & \mathbb{F}_{p^n} & & &
\end{tikzcd}$$

- f_1, f_2 as in NFS
- $V - 2$ other polynomials; linear combinations of f_1, f_2.

The Tower NFS

$R = \mathbb{Z}[\iota]/h(\iota)$, h monic irreducible of degree n (more algebraic structure).

$$\begin{tikzcd}
 & R\left[X\right] \arrow[dl] \arrow[dr] & \\
K_{f_1} \supset R\left[X\right]/(f_1(X)) \arrow[dr, "\alpha_{f_1} \mapsto m"] & & K_{f_2} \supset R\left[X\right]/(f_2(X)) \arrow[dl, "\alpha_{f_2} \mapsto m"] \\
 & R/\mathfrak{p} = \mathbb{F}_{p^n} &
\end{tikzcd}$$

The Special NFS

The characteristic p is the evaluation of a polynomial P of degree λ with small coefficients: $p = P(u)$ for $u \ll p$.

Example: BN family

- $P(z) = 36z^4 + 36z^3 + 24z^2 + 6z + 1$
- $u = -(2^{62} + 2^{55} + 1)$
- $p = P(u)$ (254 bits)

$$\boxed{p = \mathbf{16798108731015832284940804142231733909889187121439069848933715426072753864723}\ .}$$

The complexity of NFS and its variants

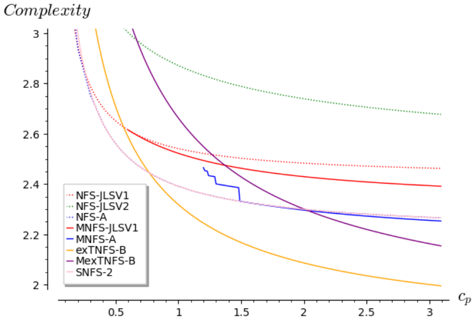
- 3 phases $=$ 3 costs $\longrightarrow$ overall complexity is sum of 3 costs.

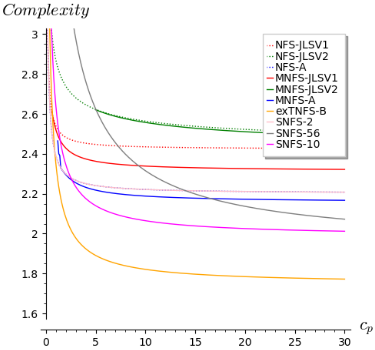
Goal: Optimize the maximum of these three costs.

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





Surprising facts:

- Not all the variants are applicable at the boundary case: STNFS has a much higher complexity!
- For small values of c_p, exTNFS better than MexTNFS.

What happens in small characteristics ?

The Function Field Sieve

$R = \mathbb{F}_p[\iota]$.

$$\begin{tikzcd}
& \mathbb{F}_p[X, Y] \arrow[dl, "Y \leftarrow g_2(X)"'] \arrow[dr, "X \leftarrow g_1(Y)"] & \\
\mathbb{F}_p[X] \arrow[dr, "X \leftarrow x"] & & \mathbb{F}_p[Y] \arrow[dl, "Y \leftarrow y"] \\
& \mathbb{F}_{p^n} &
\end{tikzcd}$$

- Function fields instead of number fields.
- Similar to the special variant.

Quasi-polynomial algorithms

A lot of recent progress:

- 2013: complexity of $L_{p^n}(1/4 + o(1))$ (Joux)
- 2014: heuristic expected running time of $2^{O((\log \log p^n)^2)}$ (Barbulescu, Gaudry, Joux, Thomé)
- 2019: proven complexity! (Kleinjung and Wesolowski [KP19])

Theorem (Theorem 1.1 in [KP19])

Given any prime number p and any positive integer n, the discrete logarithm problem in the group $\mathbb{F}_{p^n}^{\times}$ can be solved in expected time $C_{QP} = (pn)^{2\log_2(n)+O(1)}$.

Lowering the complexity of FFS

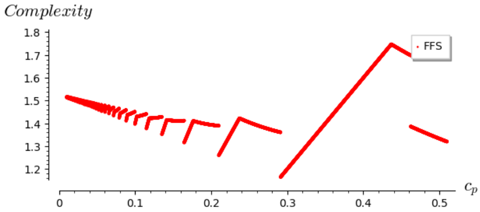
A shifted FFS

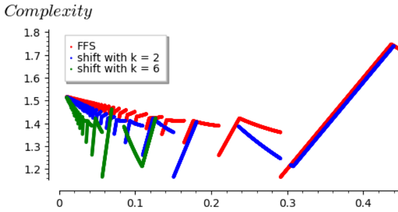
Our work: when $n = \kappa\eta$, we **lower** the complexity of FFS.

Main idea: work in a **shifted** finite field (similar to **Tower** setup)

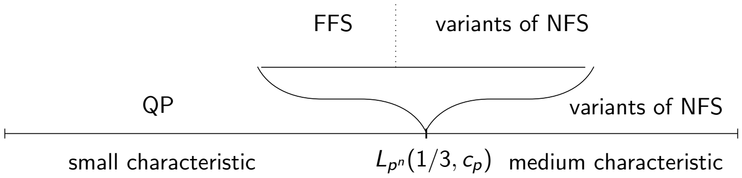
- Re-write: $\mathbb{F}_Q = \mathbb{F}_{p^n} = \mathbb{F}_{p^{\eta\kappa}} = \mathbb{F}_{p'^{\eta}}$, where $p' = p^{\kappa}$.
- From $p = L_Q(1/3, c_p)$, we get $p' = L_Q(1/3, \kappa c_p)$.

Complexity in $\mathbb{F}_{p^n}$ for $c_p = \alpha \Leftrightarrow$ complexity in $\mathbb{F}_{p'^{\eta}}$ at $c'_p = \kappa\alpha$.



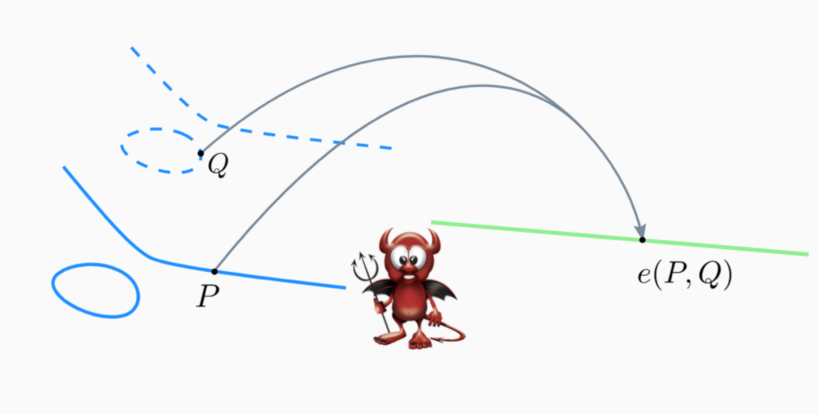


And the winners are ... !



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

On the security of pairings



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.

Did we study the correct area ?

Naive approach: $\sqrt{p} = L_Q\left(1/3, c_p\right)$.
More precise approach:

- Choose finite field where DLP is hard $\Rightarrow$ avoid QP area.

$$\boxed{p \geqslant \textbf{cross-over point between FFS and QP}}$$

- All the variants of FFS and NFS have a complexity in $L_Q(1/3, c)$: pick a finite field where the most efficient algorithm has the highest c.

  $\longrightarrow$ after our analysis, we can confirm that the highest complexities are indeed at $p = L_Q\left(1/3\right)$.

The ρ value in pairings

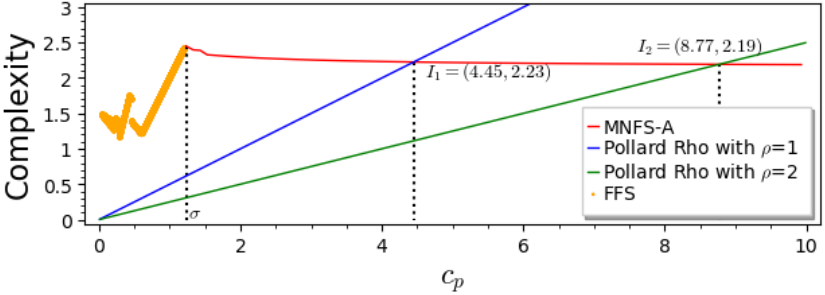
Consider a prime-order subgroup of $\mathcal{E}$ over $\mathbb{F}_p$ of size r.
Additional parameter: how large is this subgroup ?

$$\rho = \frac{\log p}{\log r}.$$

In all known construction: $\rho \in [1, 2]$.

(no efficient family of pairings asymptotically reaching $\rho = 1$.)

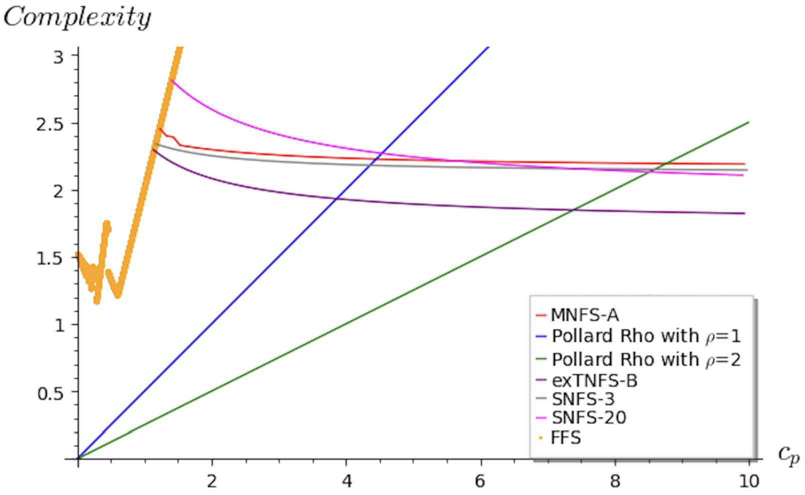
Goal: Look for value of c_p that maximizes $\min(\mathrm{comp}_{\mathcal{E}}, \mathrm{comp}_{\mathbb{F}_{p^n}})$.



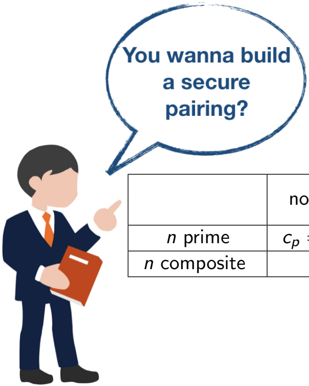
- Complexities for finite field DLP are decreasing functions.
- Pollard rho is an increasing function ($\text{complexity}_{\mathcal{E}} = p^{1/2\rho}$)

$\longrightarrow$ optimal c_p given by the intersection point!

When considering everyone!



Conclusion for pairings



	normal p	special p $\lambda = 20$	special p $\lambda = 3$
n prime	$c_p = 4.45,\ c_{\mathbf{MNFS}\text{-}\mathcal{A}} = 2.23$		$c_p = 4.36,\ c_{\mathbf{SNFS}\text{-}3} = 2.18$
n composite	$c_p = 3.93,\ c_{\mathbf{exTNFS}\text{-}\mathbf{B}} = 1.96$		

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?