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
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 $\mathcal{E}(\mathbb{F}_p)$ ...
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!
What is a cryptographic pairing?

- $G_1, G_2$: additive groups of prime order $\ell$.
- $G_T$: multiplicative group of prime order $\ell$.

**A pairing is a map** $e : G_1 \times G_2 \rightarrow G_T$

- 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 $G_1 = G_2$. 

Pairing-based cryptography
Security of pairing-based protocols

Most of the time, in cryptography:

- $G_1 =$ subgroup of $E(\mathbb{F}_p)$,
- $G_2 =$ subgroup of $E(\mathbb{F}_{p^n})$,
- $G_T =$ 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 $0 \leq l_p \leq 1$ and some constant $c > 0$.

For complexities:

- When $l_p \to 0$: $\exp(\log \log p^n) \approx \log p^n$ 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 $p = L_{p^n} \left( \frac{1}{3} \right)$, the area between the small and the medium characteristics.

Why?

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

\[
\sqrt{p} = L_p^n \left( \frac{1}{3} \right) \Rightarrow p = L_p^n \left( \frac{1}{3} \right)
\]
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 $\rightarrow$ overall complexity is sum of 3 costs.

**Goal:** Optimize the maximum of these three costs.

**Why complicated?**
1. Many parameters $\rightarrow$ discrete or continuous, boundary issues.
2. Optimization problem $\rightarrow$ Lagrange multipliers.
3. Solving a polynomial system $\rightarrow$ 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 $p$. 
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 $\min(\text{comp}_E, \text{comp}_{F_{p^n}})$.

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

$\rightarrow$ optimal $c_p$ given by the intersection point!
### Conclusion for pairings

<table>
<thead>
<tr>
<th></th>
<th>normal $p$</th>
<th>special $p$ $\lambda = 20$</th>
<th>special $p$ $\lambda = 3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$ prime</td>
<td>$c_p = 4.45$, $c_{\text{MNFS-A}} = 2.23$</td>
<td>$c_p = 4.36$, $c_{\text{SNFS-3}} = 2.18$</td>
<td></td>
</tr>
<tr>
<td>$n$ composite</td>
<td>$c_p = 3.91$, $c_{\text{MexTNFS-B}} = 1.91$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

You wanna build a secure pairing?
Thank you for your attention!

Questions?