# Algorithms for lattices of compatibly embedded finite fields

Luca De Feo[2] Jean-Pierre Flori[4]

joint work with
Ludovic Brieulle[1] Javad Doliskani[3]
Édouard Rousseau[2 5] Éric Schost[3]

[1]Université d'Aix-Marseille [2]Université de Versailles – Saint-Quentin-en-Yvelines
[3]University of Waterloo [4]Agence nationale de sécurité des systèmes d'information
[5]Télécom Paristech

# The embedding problem

**Let**

- $\mathbb{F}_q$ be a field with $q$ elements,
- $f$ and
  $g$ be irreducible polynomials
  in $\mathbb{F}_q[X]$ and $\mathbb{F}_q[Y]$,
- $r = \deg f$, $s = \deg g$ and $r | s$.

$$\begin{tikzcd}
& K = \mathbb{F}_q[Y]/g(Y) \arrow[dd, no head] \\
k = \mathbb{F}_q[X]/f(X) \arrow[ur, no head, "\varphi"] \arrow[dr, no head] & \\
& \mathbb{F}_q
\end{tikzcd}$$

There exists a field embedding

$$\varphi : k \hookrightarrow K,$$

unique
up to $\mathbb{F}_q$-automorphisms of $k$.

# Embedding description

**Determine**
elements $\alpha$ and $\beta$ such that

- $\alpha$ generates $k = \mathbb{F}_q[\alpha]$,
- there exists $\varphi : \alpha \mapsto \beta$.

$$
\begin{tikzcd}
 & K \arrow[dd, dash, "s"] \\
k = \mathbb{F}_q[\alpha] \arrow[ur, dash, "\alpha \mapsto \beta"] \arrow[dr, dash, "r"'] & \\
 & \mathbb{F}_q
\end{tikzcd}
$$

**Naive solution:** take

- $\alpha = X \mod f(X)$, and
- $\beta$ a root of $f$ in $K$.

Cost of factorization: $\tilde{O}(rs^{(\omega+1)/2})$

# Some history

'91 Lenstra [8] proves that the isomorphism problem is in P.

- Based on Kummer theory, pervasive use of linear algebra.
- Does not prove precise complexity. Rough estimate: $\Omega(r^3)$.

'92 Pinch's algorithm [11]:

- Based on mapping algebraic groups over $k, K$.
- Incomplete algorithm, no complexity analysis.

'96 Rains [12] generalizes Pinch's algorithm.

- Complete algorithm, rigorous complexity analysis.
- Unpublished. Leaves open question of using elliptic curves.

'97 Magma [3] implements lattices of finite fields using on polynomial factorization and linear algebra [4].

# Some history (cont.)

'02 Allombert's variant of Lenstra's algorithm [1, 2]:

- Trades determinism for efficiency.
- Implementation integrated into Pari/GP [14].

'07 Magma implements Rains' algorithm.

'16 Narayanan proves the first $\tilde{O}(r^2)$ upper bound [10].

- Variant of Allombert's algorithm.
- Using asymptotically fast modular composition.

Now Knowledge systematization. Notable results:

- Better variants of Allombert's algorithm.
- $\tilde{O}(r^2)$ upper bound *without fast modular composition*.
- Generalized Rains' algorithm to elliptic curves.
- C/Flint [6] and Sage [5] implementations, experiments, comparisons.

# Allombert's algorithm

Assuming $\gcd(r, q) = 1$:

- Let $h$ be an irreducible factor of the $r$-th cyclotomic polynomial over $\mathbb{F}_q$;
- Extend the action of $\mathrm{Gal}(k/\mathbb{F}_q)$ to the **ring** $k[\zeta] = k[Z]/h(Z)$:

$$\begin{array}{cccc} \sigma : & k[\zeta] & \to & k[\zeta], \\ & x \otimes \zeta & \mapsto & \sigma(x) \otimes \zeta; \end{array}$$

$$\begin{tikzcd}
k \otimes \mathbb{F}_q(\zeta) \arrow[rr, dash, "\sim"] \arrow[dr, dash, "\sigma"] \arrow[dd, dash] & & K \otimes \mathbb{F}_q(\zeta) \arrow[dl, dash, "\sigma"'] \arrow[dd, dash] \\
& \mathbb{F}_q(\zeta) \arrow[dd, dash] & \\
k \arrow[dr, dash, "\sigma"] & & K \arrow[dl, dash, "\sigma"'] \\
& \mathbb{F}_q &
\end{tikzcd}$$

- **Solve Hilbert 90:** find $\theta_1 \in k[\zeta]$ such that $\sigma(\theta_1) = \zeta\theta_1$ using linear algebra;
- Compute $\theta_2 \in K[\zeta]$ similarly;
- Compute $c = \sqrt[r]{\theta_1^r/\theta_2^r} \in \mathbb{F}_q(\zeta)$;
- Project $\theta_1 \mapsto \alpha \in k$ and $c\theta_2 \mapsto \beta \in K$.

# Implementation (take 1)

## Factorization

- The factor $h$ of $\Phi_r$ is of degree $\mathrm{ord}_r(q) = O(r)$;
- Computing it is $\tilde{O}(r)$ using Shoup [13];
- Computing $r$-th roots in $\mathbb{F}_q(\zeta)$ is $\tilde{O}(r^2)$ using Kaltofen–Shoup [7].

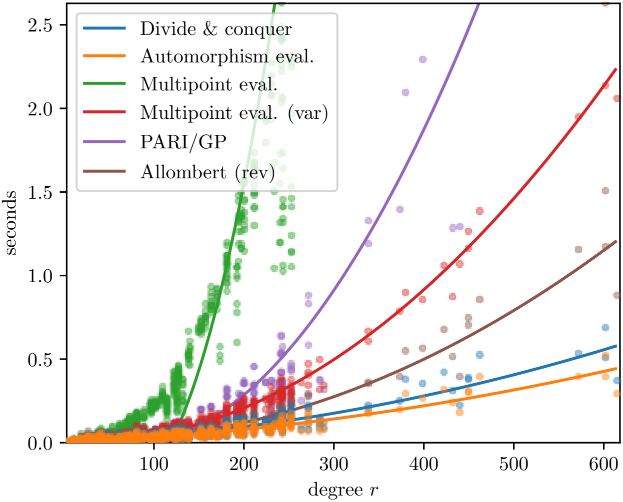
## Linear algebra

- Computing a matrix for $\sigma$ over $\mathbb{F}_q$ is $\tilde{O}(r^2)$;
- Computing its kernel over $\mathbb{F}_q(\zeta)$ is $\tilde{O}((sr)^\omega)$.

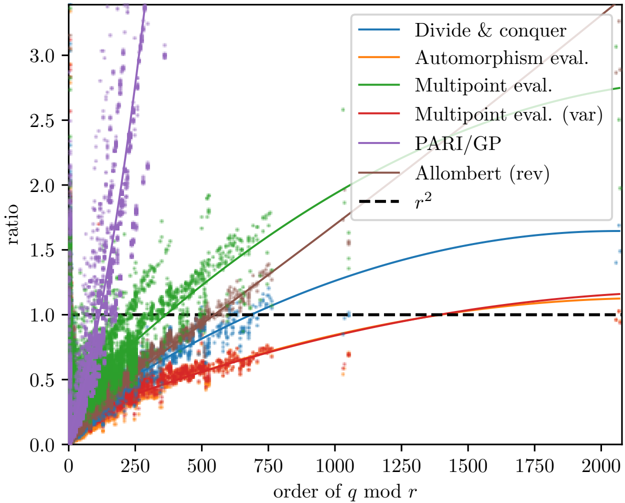
# Implementation (take 2, 3, 4, 5, . . . )

## Reaching subquadratic complexity

1. Use the factorization $h(S) = (S - \zeta)b(S)$ to perform linear algebra over $\mathbb{F}_q$.
2. Use the factorization $S^r - 1 = (S - \zeta)b(S)g(S)$ with $h$ and $g$ in $\mathbb{F}_q[S]$ to replace linear algebra by modular composition.




**Allombert's algorithm** where the auxiliary degree $s = \mathrm{ord}_r(q) \leq 10$. Dots represent individual runs, lines represent degree 2 linear regressions.




**Allombert's algorithm,** as a function of the auxiliary degree $s = \mathrm{ord}_r(q)$ scaled down by $r^2$.

# Pinch's algorithm

**Pinch's idea**

- Find *small* $\ell$ such that $k \simeq \mathbb{F}_q[\mu_\ell]$,
- Pick $\ell$-th roots of unity $\alpha \in k$, $\beta \in K$,
- Find $e$ s.t. $\alpha \mapsto \beta^e$ using brute force.
- **Problem 1:** worst case $\ell \in O(q^r)$.
- **Problem 2:** potentially $O(\ell)$ exponents $e$ to test depending on the splitting of $\Phi_l$ over $\mathbb{F}_q$.

$$\begin{tikzcd}
k \arrow[r, dash, "\sim"] \arrow[dr, dash] & \mathbb{F}_q[\mu_\ell] \arrow[r, dash, "\sim"] \arrow[d, dash] & K \arrow[dl, dash] \\
& \mathbb{F}_q &
\end{tikzcd}$$

# Rains' algorithm and variants

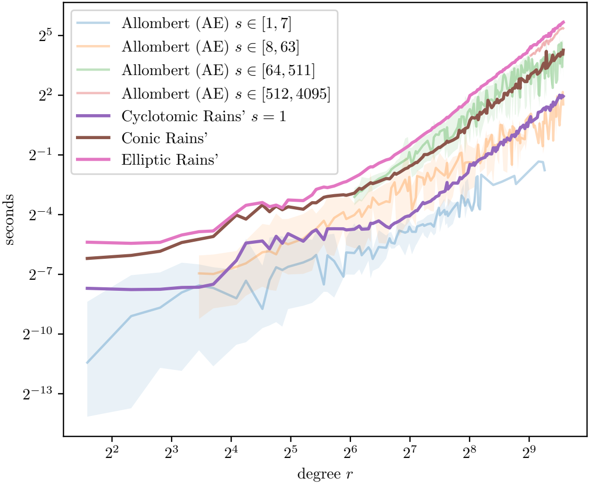
- Replace $\alpha, \beta$ with *Gaussian periods*:

$$\eta(\alpha) = \sum_{\sigma \in S} \alpha^\sigma$$

where $(\mathbb{Z}/\ell\mathbb{Z})^\times = \langle q \rangle \times S$.

  - Periods are normal elements, hence yield bases of $k$, $K$.
  - Periods are unique up to Galois action, hence $\eta(\alpha) \mapsto \eta(\beta)$ always defines an isomorphism.
  - The size of $\ell$ can be controlled by allowing auxiliary extensions.

- Use higher dimensional algebraic groups:
  - Replace $\mathbb{F}_q[\mu_\ell]$ with the $\ell$-torsion of *random* elliptic curves $E/\mathbb{F}_q$;
  - Replace Gaussian periods with *elliptic periods* [9];
  - This removes the need for auxiliary extensions.




**Allombert's vs Rains'** at some fixed auxiliary extension degrees $s$. Lines represent median times, shaded areas minimum and maximum times.

# Part II: Compatible embeddings

# The compatibility problem

**Context:**

- $E$, $F$, $G$ fields
- $E$ subfield of $F$ and $F$ subfield of $G$
- $\phi_{E\hookrightarrow F}$, $\phi_{F\hookrightarrow G}$, $\phi_{E\hookrightarrow G}$ embeddings

$$\begin{tikzcd}
G & \\
 & F \arrow[ul, "\phi_{F\hookrightarrow G}"'] \\
E \arrow[uu, "\phi_{E\hookrightarrow G}"] \arrow[ur, "\phi_{E\hookrightarrow F}"'] &
\end{tikzcd}$$

$$\phi_{F\hookrightarrow G} \circ \phi_{E\hookrightarrow F} \stackrel{?}{=} \phi_{E\hookrightarrow G}$$

# The compatibility problem

$$
\begin{tikzcd}
& H & \\
G \arrow[ur, "\phi_{G\hookrightarrow H}"] & & F \arrow[ul, "\phi_{F\hookrightarrow G}"'] \\
& E \arrow[ul, "\phi_{E\hookrightarrow G}"] \arrow[ur, "\phi_{E\hookrightarrow F}"'] &
\end{tikzcd}
$$

$$\phi_{G\hookrightarrow H} \circ \phi_{E\hookrightarrow G} \stackrel{?}{=} \phi_{F\hookrightarrow H} \circ \phi_{E\hookrightarrow F}$$

# Bosma, Cannon and Steel '97 [4]

- Based upon *naive* embedding algorithms.
- Supports arbitrary, user-defined finite fields.
- Allows to compute the embeddings in arbitrary order.
- Implemented by MAGMA.

# The Bosma, Cannon and Steel framework

$$\begin{tikzcd}
 & G & \\
 & & F \arrow[ul, dashed] \\
E \arrow[uur, "\phi_{E \hookrightarrow G}"] \arrow[urr, "\phi_{E \hookrightarrow F}"'] & &
\end{tikzcd}$$

- Take $\phi'_{F \hookrightarrow G}$ an arbitrary embedding between $F$ and $G$
- Find $\sigma \in \mathrm{Gal}(G/\mathbb{F}_p)$ such that $\sigma \circ \phi'_{F \hookrightarrow G} \circ \phi_{E \hookrightarrow F} = \phi_{E \hookrightarrow G}$
- Set $\phi_{F \hookrightarrow G} := \sigma \circ \phi'_{F \hookrightarrow G}$
- There are $|\mathrm{Gal}(F/E)|$ compatible morphisms

# Bosma, Cannon and Steel framework

What about several subfields $E_1, E_2, \dots, E_r$ ?

- Enforce these axioms on the lattice:
  - CE1 (Unicity) At most one morphism $\phi_{E \hookrightarrow F}$
  - CE2 (Reflexivity) For each $E$, $\phi_{E \hookrightarrow E} = \mathrm{Id}_E$
  - CE3 (Invertibility) For each pair $(E, F)$ with $E \cong F$, $\phi_{E \hookrightarrow F} = \phi_{F \hookrightarrow E}^{-1}$
  - CE4 (Transitivity) For any triple $(E, F, G)$ with $E$ subfield of $F$ and $F$ subfield of $G$, if we have computed $\phi_{E \hookrightarrow F}$ and $\phi_{F \hookrightarrow G}$, then $\phi_{E \hookrightarrow G} = \phi_{F \hookrightarrow G} \circ \phi_{E \hookrightarrow F}$
  - CE5 (**Intersections**) For any triple $(E, F, G)$ with $E$ and $F$ subfields of $G$, we have that the field $S = E \cap F$ is embedded in $E$ and $F$, *i.e.* we have computed $\phi_{S \hookrightarrow E}$ and $\phi_{S \hookrightarrow F}$

# The Bosma, Cannon and Steel framework

$$\begin{tikzcd}
 & & & G & \\
 & & & & F \arrow[ul, dashed] \\
E_1 \arrow[uurrr] \arrow[urrrr] & E_2 \arrow[uurr] \arrow[urrr] & \cdots & E_r \arrow[uu] \arrow[ur] &
\end{tikzcd}$$

- Set $F'$ the field generated by the fields $E_i$ in $F$
- Set $G'$ the field generated by the fields $E_i$ in $G$

## Theorem

There exists a unique isomorphism $\chi : F' \to G'$ that is compatible with all embeddings, *i.e.* such that for all $i$, $\phi_{E_i \hookrightarrow G'} = \chi \circ \phi_{E_i \hookrightarrow F'}$.

# New problem: compute embeddings with common subfields

- We want to embed $E$ in $F$
  - additionnal information: $S$ is a field embedded in $E$ and $F$
- The naive algorithm can be sped up by replacing $\mathbb{F}_p$ with $S$ as base field
  (degree $[E:S]$ polynomial factorization **vs** degree $[E:\mathbb{F}_p]$)
- More generally: $S$ the compositum of all **known** fields embedded in $E$ and $F$.

# Some questions

- Bosma, Cannon and Steel framework + Allombert's algorithm: *any smart optimizations possible*?
- Allombert's algorithm *with common subfield knowledge*?

# Demo

- Our implementations of Allombert's algorithm + *embedding evaluation* are being pushed into Flint (https://github.com/wbhart/flint2/pull/351);
- A compatible embedding framework is being added to Nemo (https://github.com/Nemocas/Nemo.jl/issues/233).

**Go to the demo:**

https://github.com/defeo/Nemo-embeddings-demo

# Questions ?

# References I


Bill Allombert.
Explicit computation of isomorphisms between finite fields.
*Finite Fields Appl.*, 8(3):332 – 342, 2002.

Bill Allombert.
Explicit computation of isomorphisms between finite fields.
Revised version. https://www.math.u-bordeaux.fr/~ballombe/fpisom.ps, 2002.

Wieb Bosma, John Cannon, and Catherine Playoust.
The MAGMA algebra system I: the user language.
*J. Symbolic Comput.*, 24(3-4):235–265, 1997.

Wieb Bosma, John Cannon, and Allan Steel.
Lattices of compatibly embedded finite fields.
*Journal of Symbolic Computation*, 24(3-4):351–369, 1997.

The Sage Developers.
*SageMath, the Sage Mathematics Software System (Version 7.5.rc0)*, 2016.
http://www.sagemath.org.

William Hart.
Fast library for number theory: an introduction.
*Mathematical Software-ICMS 2010*, pages 88–91, 2010.

# References II


Erich Kaltofen and Victor Shoup.
Fast polynomial factorization over high algebraic extensions of finite fields.
In *ISSAC '97: Proceedings of the 1997 international symposium on Symbolic and algebraic computation*, pages 184–188, New York, NY, USA, 1997. ACM.

Hendrik W. Lenstra.
Finding isomorphisms between finite fields.
*Mathematics of Computation*, 56(193):329–347, 1991.

Preda Mihailescu, François Morain, and Éric Schost.
Computing the eigenvalue in the Schoof-Elkies-Atkin algorithm using abelian lifts.
In *ISSAC '07: Proceedings of the 2007 international symposium on Symbolic and algebraic computation*, pages 285–292, New York, NY, USA, 2007. ACM.

Anand Kumar Narayanan.
Fast computation of isomorphisms between finite fields using elliptic curves.
arXiv preprint arXiv:1604.03072, 2016.

Richard G. E. Pinch.
Recognising elements of finite fields.
In *Cryptography and Coding II*, pages 193–197. Oxford University Press, 1992.

Eric M. Rains.
Efficient computation of isomorphisms between finite fields.
personal communication, 1996.

# References III


Victor Shoup.
Fast construction of irreducible polynomials over finite fields.
*Journal of Symbolic Computation*, 17(5):371–391, 1994.

The PARI Group, Bordeaux.
*PARI/GP, version* `2.8.0`, 2016.