Simplified Settings for Discrete Logarithms in Small Characteristic Finite Fields

Antoine Joux [1] Cécile Pierrot [2]

[1]CryptoExperts, Paris and Chaire de la Fondation Partenariale de l'UPMC, France

[2]DGA, CNRS, INRIA and UPMC, Sorbonne-Universités, Paris, France

December 9th

Asiacrypt 2014, Kaohsiung, Taiwan

The Discrete Logarithm Problem (DLP)

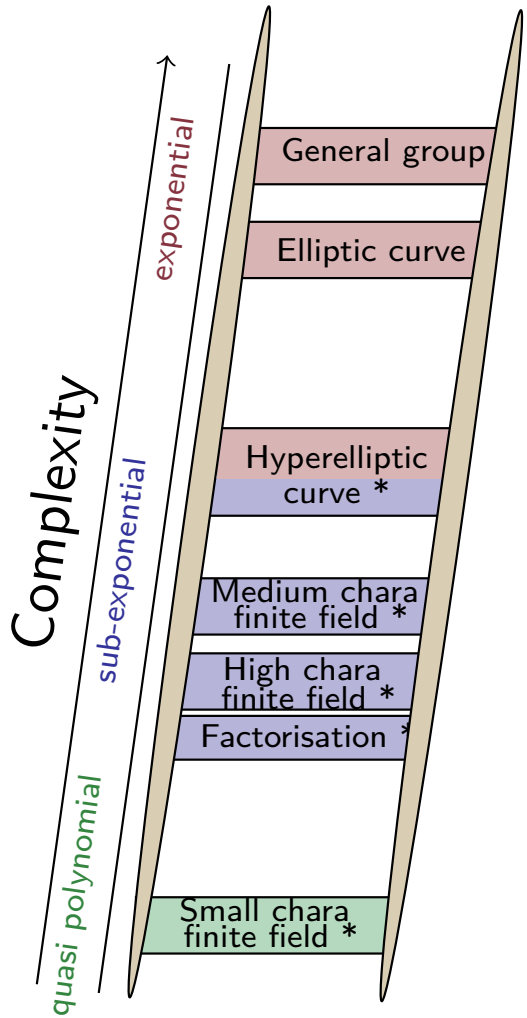
- Multiplicative group G generated by g: solving the discrete logarithm problem in G, is inverting the map $x \mapsto g^x$

The Discrete Logarithm Problem (DLP)

- Multiplicative group G generated by g: solving the discrete logarithm problem in G, is inverting the map $x \mapsto g^x$
- A hard problem in general, and used as such in cryptography.

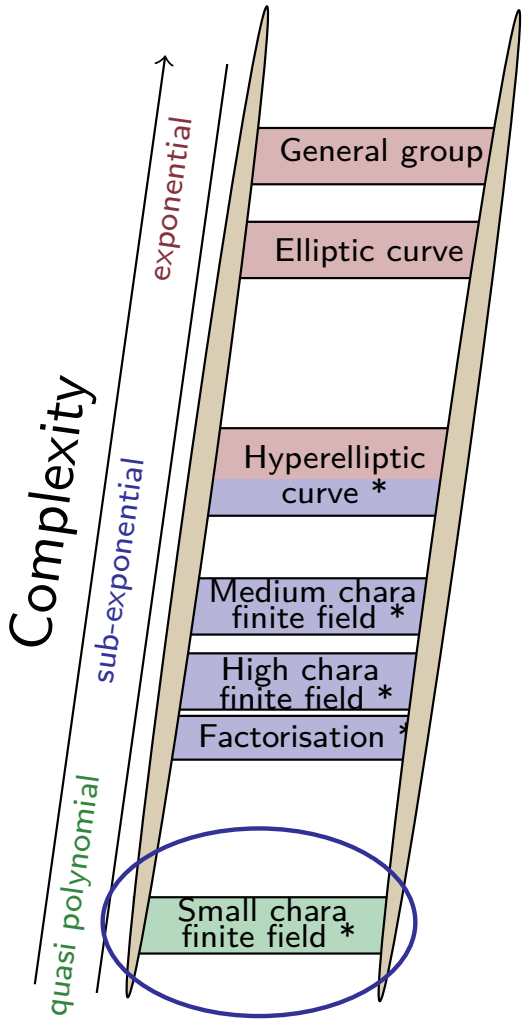
The Discrete Logarithm Problem (DLP)

- Multiplicative group G generated by g: solving the discrete logarithm problem in G, is inverting the map $x \mapsto g^x$
- A hard problem in general, and used as such in cryptography.
- Several groups in practice:



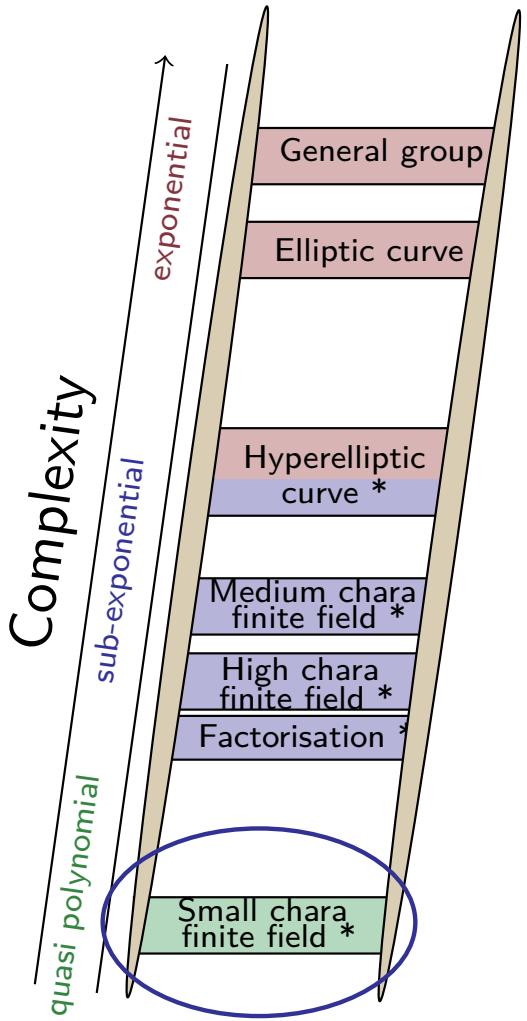
The Discrete Logarithm Problem (DLP)

- Multiplicative group G generated by g: solving the discrete logarithm problem in G, is inverting the map $x \mapsto g^x$
- A hard problem in general, and used as such in cryptography.
- Several groups in practice:



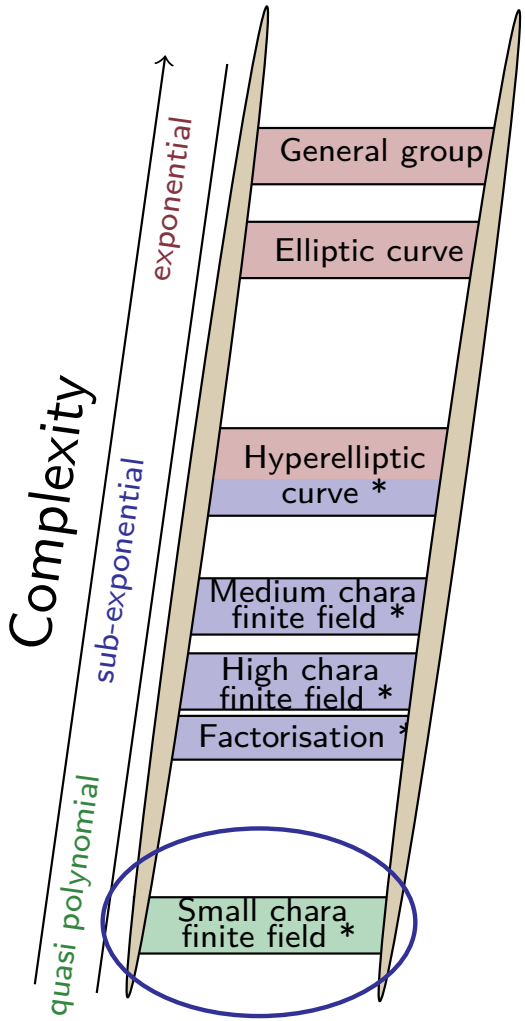
The Discrete Logarithm Problem (DLP)

- Multiplicative group G generated by g: solving the discrete logarithm problem in G, is inverting the map $x \mapsto g^x$
- A hard problem in general, and used as such in cryptography.
- Several groups in practice:
- Two families of algorithms :
  - Generic algorithms (Pollard's Rho, Pohlig-Hellman...)
  - Specific algorithms (Index Calculus)



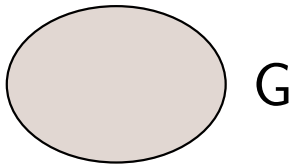
The Discrete Logarithm Problem (DLP)

- Multiplicative group G generated by g: solving the discrete logarithm problem in G, is inverting the map $x \mapsto g^x$
- A hard problem in general, and used as such in cryptography.
- Several groups in practice:
- Two families of algorithms :
  - Generic algorithms (Pollard's Rho, Pohlig-Hellman...)
  - Specific algorithms (Index Calculus *)



Index Calculus Algorithms

If you want to compute Discrete Logs in G:



1. Collection of Relations

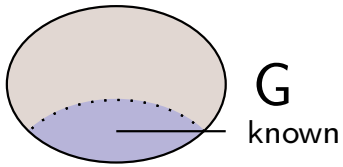
   $\rightarrow$ Create a lot of sparse multiplicative relations between some (small) specific elements = the factor base

$$\prod g_i^{e_i} = \prod g_i^{e'_i} \quad \Rightarrow \quad \sum (e_i - e'_i) \log(g_i) = 0$$

   $\rightarrow$ So a lot of sparse linear equations

Index Calculus Algorithms

If you want to compute Discrete Logs in G:



1. Collection of Relations

   $\rightarrow$ Create a lot of sparse multiplicative relations between some (small) specific elements = the factor base

$$\prod g_i^{e_i} = \prod g_i^{e'_i} \quad \Rightarrow \quad \sum (e_i - e'_i) \log(g_i) = 0$$

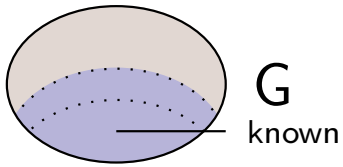
   $\rightarrow$ So a lot of sparse linear equations

2. Linear Algebra

   $\rightarrow$ Recover the Discrete Logs of the factor base

Index Calculus Algorithms

If you want to compute Discrete Logs in G:



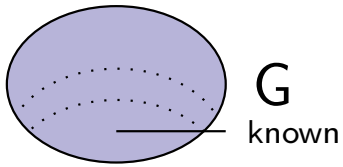
1. Collection of Relations
   $\rightarrow$ Create a lot of sparse multiplicative relations between some (small) specific elements = the factor base

$$\prod g_i^{e_i} = \prod g_i^{e_i'} \quad \Rightarrow \quad \sum (e_i - e_i') \log(g_i) = 0$$

   $\rightarrow$ So a lot of sparse linear equations
2. Linear Algebra
   $\rightarrow$ Recover the Discrete Logs of the factor base
3. Extension Phase (for small characteristic finite fields)
   $\rightarrow$ Recover the Discrete Logs of the extended factor base

Index Calculus Algorithms

If you want to compute Discrete Logs in G:



1. Collection of Relations
   $\rightarrow$ Create a lot of sparse multiplicative relations between some (small) specific elements = the factor base

$$\prod g_i^{e_i} = \prod g_i^{e_i'} \quad \Rightarrow \quad \sum (e_i - e_i') \log(g_i) = 0$$

   $\rightarrow$ So a lot of sparse linear equations
2. Linear Algebra
   $\rightarrow$ Recover the Discrete Logs of the factor base
3. Extension Phase (for small characteristic finite fields)
   $\rightarrow$ Recover the Discrete Logs of the extended factor base
4. Individual Logarithm Phase
   $\rightarrow$ Recover the Discrete Log of an arbitrary element

Complexities

- Asymptotic Complexities:

Collection of Relations Linear Algebra Extension Phase	Polynomial time
Individual Logarithm Phase	Quasipolynomial time

Complexities

- Asymptotic Complexities:

  Collection of Relations, Linear Algebra, Extension Phase } Polynomial time

  Individual Logarithm Phase } Quasipolynomial time

- But in practice:
  Linear algebra and extension phases dominate

Complexities

- Asymptotic Complexities:

  Collection of Relations, Linear Algebra, Extension Phase } Polynomial time

  Individual Logarithm Phase } Quasipolynomial time

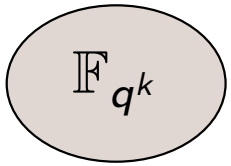
- But in practice:
  Linear algebra and extension phases dominate
- What do we do ?
  Simplified description of algorithms + additional ideas
  $\Rightarrow$ Improve the complexity of the polynomial phases.

Frobenius Representation Algorithms

- Our goal: solve the DLP in $\mathbb{F}_{q^k}$.

Frobenius Representation Algorithms

- Our goal: solve the DLP in $\mathbb{F}_{q^k}$.



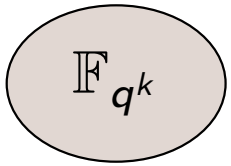
- How ? Represent $\mathbb{F}_{q^k} \simeq \mathbb{F}_q[X]/(I(X))$ where $I(X)$ is an irreducible polynomial of degree k such that:

$$I(X) | h_1(X)X^q - h_0(X)$$

where h_0 and h_1 are polynomials of low degrees.

Frobenius Representation Algorithms

- Our goal: solve the DLP in $\mathbb{F}_{q^k}$.



- How ? Represent $\mathbb{F}_{q^k} \simeq \mathbb{F}_q[X]/(I(X))$ where $I(X)$ is an irreducible polynomial of degree k such that:

$$I(X) | h_1(X)X^q - h_0(X)$$

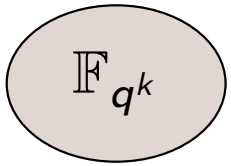
where h_0 and h_1 are polynomials of low degrees.

- Why ? To have two equations in the finite field:

$$\prod_{\alpha \in \mathbb{F}_q} (X - \alpha) = X^q - X \qquad \text{and} \qquad \underbrace{X^q = \frac{h_0(X)}{h_1(X)}}_{\text{Frobenius Representation}}$$

Frobenius Representation Algorithms

- Our goal: solve the DLP in $\mathbb{F}_{q^k}$.



- How ? Represent $\mathbb{F}_{q^k} \simeq \mathbb{F}_q[X]/(I(X))$ where $I(X)$ is an irreducible polynomial of degree k such that:

$$I(X) | h_1(X)X^q - h_0(X)$$

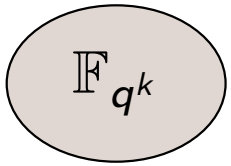
  where h_0 and h_1 are polynomials of low degrees.
- Why ? To have two equations in the finite field:

$$\prod_{\alpha \in \mathbb{F}_q} (X - \alpha) = X^q - X \qquad \text{and} \qquad \underbrace{X^q = \frac{h_0(X)}{h_1(X)}}_{\text{Frobenius Representation}}$$

- What choice do we have ? Degree of h_0 and h_1.

Frobenius Representation Algorithms

- Our goal: solve the DLP in $\mathbb{F}_{q^k}$.



- How ? Represent $\mathbb{F}_{q^k} \simeq \mathbb{F}_q[X]/(I(X))$ where $I(X)$ is an irreducible polynomial of degree k such that:

$$I(X) | h_1(X)X^q - h_0(X)$$

where h_0 and h_1 are polynomials of low degrees.

- Why ? To have two equations in the finite field:

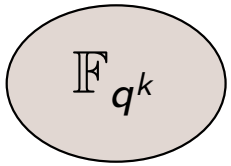
$$\prod_{\alpha \in \mathbb{F}_q} (X - \alpha) = X^q - X \qquad \text{and} \qquad \underbrace{X^q = \frac{h_0(X)}{h_1(X)}}_{\text{Frobenius Representation}}$$

- What choice do we have ? Degree of h_0 and h_1.
- What would be simple ? To take

h_0 : deg 1 polynomial
h_1 : deg 2 polynomial

Frobenius Representation Algorithms

- Our goal: solve the DLP in $\mathbb{F}_{q^k}$.



- How ? Represent $\mathbb{F}_{q^k} \simeq \mathbb{F}_q[X]/(I(X))$ where $I(X)$ is an irreducible polynomial of degree k such that:

$$I(X) | h_1(X)X^q - h_0(X) \quad \text{or} \quad I(X) | h_1(X^q)X - h_0(X^q)$$

  where h_0 and h_1 are polynomials of low degrees.
- Why ? To have two equations in the finite field:

$$\prod_{\alpha \in \mathbb{F}_q} (X - \alpha) = X^q - X \qquad \text{and} \qquad \underbrace{X^q = \frac{h_0(X)}{h_1(X)}}_{\text{Frobenius Representation}} \qquad \text{or} \qquad \underbrace{X = \frac{h_0(X^q)}{h_1(X^q)}}_{\text{Dual Frob. Rep.}}$$

- What choice do we have ? Degree of h_0 and h_1.
- What would be simple ? To take

$$\left.\begin{array}{ll} h_0 : & \text{deg 1 polynomial} \\ h_1 : & \text{deg 2 polynomial} \end{array}\right. \quad \text{or} \quad \left.\begin{array}{ll} h_0 : & \text{deg 2 polynomial} \\ h_1 : & \text{deg 1 polynomial} \end{array}\right\} \text{useful variant}$$

Creation of Relations

Our goal: multiplicative relation between small degree polynomials.

Creation of Relations

Our goal: multiplicative relation between small degree polynomials.

Main idea : start from $\prod_{\alpha \in \mathbb{F}_q} (X - \alpha) = X^q - X \quad (**)$.

Let A and B be 2 small polynomials in $\mathbb{F}_q[X]$ (i.e. of degree $\leqslant D$).

$$B(X) \prod_{\alpha \in \mathbb{F}_q} (A(X) - \alpha B(X)) \underset{\text{thanks to } (**)}{=} A(X)^q B(X) - A(X) B(X)^q$$

Creation of Relations

Our goal: multiplicative relation between small degree polynomials.

Main idea : start from $\prod_{\alpha \in \mathbb{F}_q} (X - \alpha) = X^q - X \quad (**)$.

Let A and B be 2 small polynomials in $\mathbb{F}_q[X]$ (i.e. of degree $\leqslant D$).

$$B(X) \prod_{\alpha \in \mathbb{F}_q} (A(X) - \alpha B(X)) \underset{\text{thanks to } (**)}{=} A(X)^q B(X) - A(X) B(X)^q \underset{\text{Frob. linearity}}{=} A(X^q) B(X) - A(X) B(X^q)$$

Creation of Relations

Our goal: multiplicative relation between small degree polynomials.

Main idea : start from $\prod_{\alpha \in \mathbb{F}_q} (X - \alpha) = X^q - X \quad (**)$.

Let A and B be 2 small polynomials in $\mathbb{F}_q[X]$ (i.e. of degree $\leqslant D$).

$$
\begin{aligned}
B(X) \textstyle\prod_{\alpha \in \mathbb{F}_q} (A(X) - \alpha B(X)) &\underset{\text{thanks to } (**)}{=} A(X)^q B(X) - A(X) B(X)^q \\
&\underset{\text{Frob. linearity}}{=} A(X^q) B(X) - A(X) B(X^q) \\
&\underset{\text{Frob. Rep.}}{=} \underbrace{A\left(\tfrac{h_0(X)}{h_1(X)}\right) B(X) - A(X) B\left(\tfrac{h_0(X)}{h_1(X)}\right)}_{[A,B]_D} \\
&\qquad\qquad \frac{[A,B]_D}{h_1(X)^D}
\end{aligned}
$$

Creation of Relations

Our goal: multiplicative relation between small degree polynomials.

Main idea : start from $\prod_{\alpha \in \mathbb{F}_q} (X - \alpha) = X^q - X \quad (**)$.

Let A and B be 2 small polynomials in $\mathbb{F}_q[X]$ (i.e. of degree $\leqslant D$).

$$
\begin{aligned}
B(X)\prod_{\alpha\in\mathbb{F}_q}(A(X)-\alpha B(X)) &\underset{\text{thanks to } (**)}{=} A(X)^q B(X) - A(X)B(X)^q \\
&\underset{\text{Frob. linearity}}{=} A(X^q)B(X) - A(X)B(X^q) \\
&\underset{\text{Frob. Rep.}}{=} \underbrace{A\left(\frac{h_0(X)}{h_1(X)}\right)B(X) - A(X)B\left(\frac{h_0(X)}{h_1(X)}\right)}_{[A,B]_D} \\
&\qquad\qquad\qquad\qquad \frac{[A,B]_D}{h_1(X)^D}
\end{aligned}
$$

We finally get:

$$\underbrace{h_1(X)^D B(X) \prod_{\alpha \in \mathbb{F}_q} (A(X) - \alpha B(X))}_{\text{Product of small polynomials !!}} = [A,B]_D(X)$$

A Reduced Factor Base

We have: $$\underbrace{h_1(X)^D B(X) \prod_{\alpha \in \mathbb{F}_q} (A(X) - \alpha B(X))}_{\text{polynomials of degree} \leqslant D} = [A, B]_D(X)$$

- A natural Factor Base: Irreducible poly in $\mathbb{F}_q[X]$ of deg $\leqslant D$.

A Reduced Factor Base

We have: $$\underbrace{h_1(X)^D B(X) \prod_{\alpha \in \mathbb{F}_q} (A(X) - \alpha B(X))}_{\text{polynomials of degree} \leqslant D} = [A, B]_D(X)$$

- A natural Factor Base: Irreducible poly in $\mathbb{F}_q[X]$ of deg $\leqslant D$.
- $D \searrow \Rightarrow$ size of the factor base $\searrow \Rightarrow$ complexity of Linear Algebra $\searrow$. The smaller, the better. Until now, $D = 3$.

A Reduced Factor Base

We have: $\underbrace{h_1(X)^D B(X) \prod_{\alpha \in \mathbb{F}_q} (A(X) - \alpha B(X))}_{\text{polynomials of degree} \leqslant D} = [A, B]_D(X)$

- A natural Factor Base: Irreducible poly in $\mathbb{F}_q[X]$ of deg $\leqslant D$.
- $D \searrow \Rightarrow$ size of the factor base $\searrow \Rightarrow$ complexity of Linear Algebra $\searrow$. The smaller, the better. Until now, $D = 3$.
- What is simple ? Irreducible poly in $\mathbb{F}_q[X]$ of degree $\leqslant 2$.
- But... it's not possible to $\searrow D$, isn't it?

A Reduced Factor Base

We have: $\underbrace{h_1(X)^D B(X) \prod_{\alpha \in \mathbb{F}_q} (A(X) - \alpha B(X))}_{\text{polynomials of degree} \leqslant D} = \underbrace{[A, B]_D(X)}_{\text{Factorization ?}}$

- A natural Factor Base: Irreducible poly in $\mathbb{F}_q[X]$ of deg $\leqslant D$.
- $D \searrow \Rightarrow$ size of the factor base $\searrow \Rightarrow$ complexity of Linear Algebra $\searrow$. The smaller, the better. Until now, $D = 3$.
- What is simple ? Irreducible poly in $\mathbb{F}_q[X]$ of degree $\leqslant 2$.
- But... it's not possible to $\searrow D$, isn't it? Previous constraints:
  1. Need to generate enough good equations = equations where $[A, B]_2$ splits in terms of degree $\leqslant 2$. Pb: the probability $\mathcal{P}$ to have good equations is too small w.r.t the number of equations required (need $\mathcal{P} > 1/2$).
  2. Need to be able to descent large polynomials to degree 2 ones.

A Reduced Factor Base: Systematic factors of $[A, B]_D$

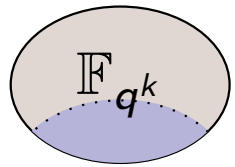
- Our goal, solving pb 1: i.e. improve the probability $\mathcal{P}$.
- How ? $[A, B]_2$ is a degree 6 polynomial. The prob that it factors into degree 2 polynomials is too low.

A Reduced Factor Base: Systematic factors of $[A, B]_D$

- Our goal, solving pb 1: i.e. improve the probability $\mathcal{P}$.
- How ? $[A, B]_2$ is a degree 6 polynomial. The prob that it factors into degree 2 polynomials is too low.
  Yet, $[A, B]_D$ has a systematic factor of degree 3 ! Namely $X\, h_1(X) - h_0(X)$.
- A degree 3 polynomial factors into terms of degree at most 2 with prob $\mathcal{P} > 2/3 > 1/2$.

A Reduced Factor Base: Systematic factors of $[A, B]_D$

- Our goal, solving pb 1: i.e. improve the probability $\mathcal{P}$.
- How ? $[A, B]_2$ is a degree 6 polynomial. The prob that it factors into degree 2 polynomials is too low.
  Yet, $[A, B]_D$ has a systematic factor of degree 3 ! Namely $X\, h_1(X) - h_0(X)$.
- A degree 3 polynomial factors into terms of degree at most 2 with prob $\mathcal{P} > 2/3 > 1/2$.



$\Rightarrow$ Linear Algebra permits to recover the DLogs of the factor base in $O((\underbrace{\#\text{ factor base}}_{q^2})^2(\underbrace{\#\text{ of entries}}_{q})) \approx O(q^5)$ operations.

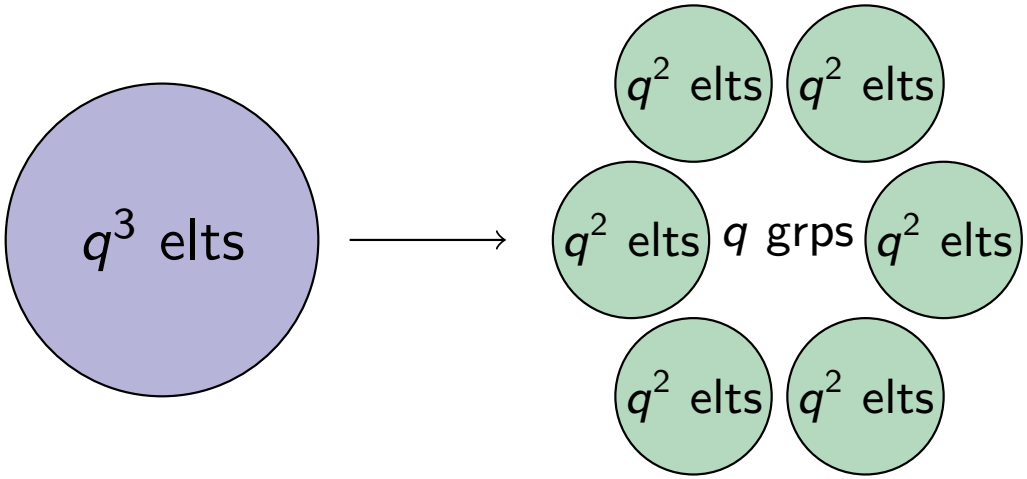
Extend the Factor Base to Degree 3

Our goal: Solving pb 2 i.e. extend the factor base to degree 3
BUT without performing linear algebra on a matrix of dim q^3.

Extend the Factor Base to Degree 3

Our goal: Solving pb 2 i.e. extend the factor base to degree 3
BUT without performing linear algebra on a matrix of dim q^3.

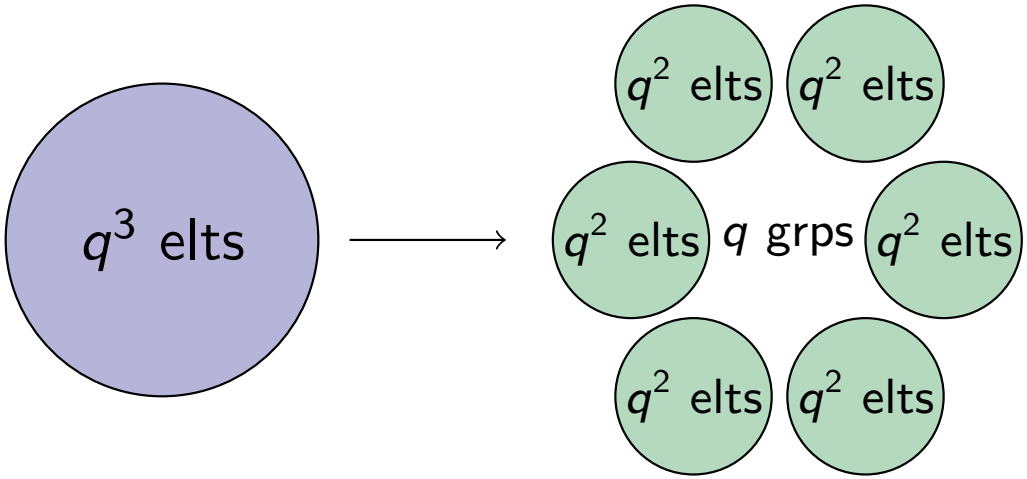
1. Divide the irreducible deg. 3 monic polynomials in groups.



Extend the Factor Base to Degree 3

Our goal: Solving pb 2 i.e. extend the factor base to degree 3
BUT without performing linear algebra on a matrix of dim q^3.

1. Divide the irreducible deg. 3 monic polynomials in groups.

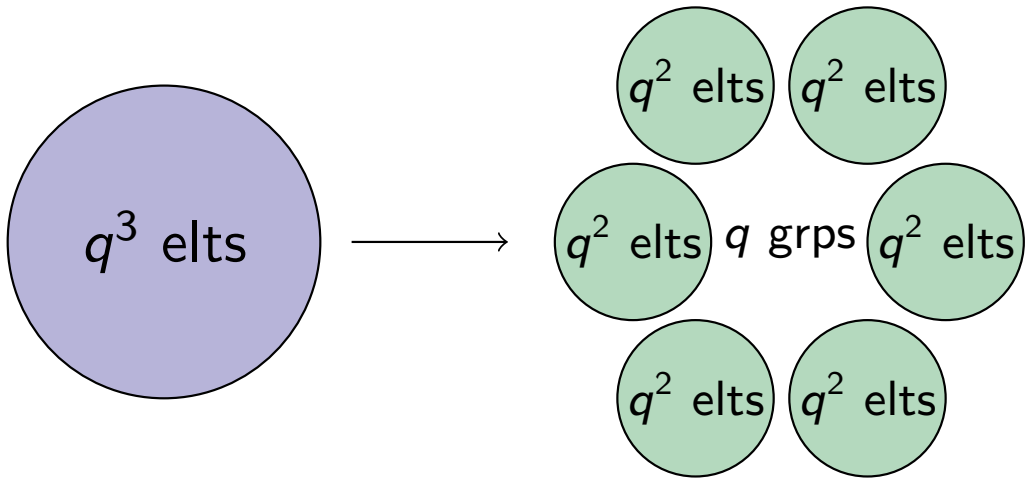


What is simple ? To consider that 2 polynomials belongs to the same group if they have the same constant coefficient.

Extend the Factor Base to Degree 3

Our goal: Solving pb 2 i.e. extend the factor base to degree 3
BUT without performing linear algebra on a matrix of dim q^3.

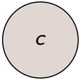
1. Divide the irreducible deg. 3 monic polynomials in groups.



What is simple ? To consider that 2 polynomials belongs to the same group if they have the same constant coefficient.

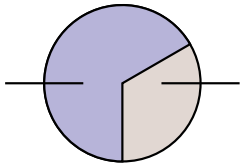
2. Given q^2, generate equations involving only poly in q^2 and degree 1 and 2 polynomials (whose logs are already known).

- An example: let $\textcircled{c} = \{(X^3 + c) + \alpha X^2 + \beta X | (\alpha, \beta) \in \mathbb{F}_q{}^2\}$.



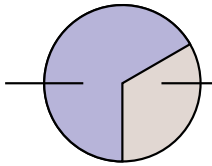
Extend the Factor Base to Degree 3

- An example: let c $= \{(X^3 + c) + \alpha X^2 + \beta X | (\alpha, \beta) \in \mathbb{F}_q{}^2\}$.



Extend the Factor Base to Degree 3

- An example: let $c = \{(X^3 + c) + \alpha X^2 + \beta X | (\alpha, \beta) \in \mathbb{F}_q{}^2\}$.

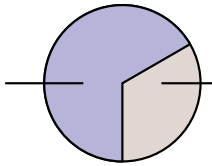


As for degree 2: set $A(X) = (X^3 + c) + \alpha X^2$ and $B(X) = (X^3 + c) + \beta X$ and create relations of the form:

$$\underbrace{h_1(X)^3 B(X) \prod_{\alpha \in \mathbb{F}_q} (A(X) - \alpha B(X))}_{\text{all belongs to } c \text{ !!}} = \underbrace{[A, B]_3(X)}_{\begin{array}{l}\text{deg 8 with these } A \text{ and } B \\ + \text{ deg 3 systematic factor} \\ + \text{ divisible by } X\end{array}}$$

Extend the Factor Base to Degree 3

- An example: let $\mathcal{C} = \{(X^3 + c) + \alpha X^2 + \beta X | (\alpha, \beta) \in \mathbb{F}_q{}^2\}$.



$\Rightarrow$ new unknowns

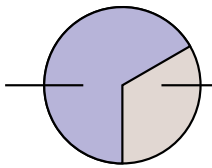
As for degree 2: set $A(X) = (X^3 + c) + \alpha X^2$ and $B(X) = (X^3 + c) + \beta X$ and create relations of the form:

$$\underbrace{h_1(X)^3 B(X) \prod_{\alpha \in \mathbb{F}_q} (A(X) - \alpha B(X))}_{\text{all belongs to } \mathcal{C} \text{ !!}} = \underbrace{[A, B]_3(X)}_{\substack{\text{deg 8 with these } A \text{ and } B \\ + \text{ deg 3 systematic factor} \\ + \text{ divisible by } X}}$$

Prob that $[A, B]_3$ factors into deg $\leqslant 3 \Rightarrow 41\%$. Enough !

Extend the Factor Base to Degree 3

- An example: let $\boxed{c} = \{(X^3 + c) + \alpha X^2 + \beta X | (\alpha, \beta) \in \mathbb{F}_q{}^2\}$.



As for degree 2: set $A(X) = (X^3 + c) + \alpha X^2$ and $B(X) = (X^3 + c) + \beta X$ and create relations of the form:

$$\underbrace{h_1(X)^3 B(X) \prod_{\alpha \in \mathbb{F}_q} (A(X) - \alpha B(X))}_{\text{all belongs to } \boxed{c} \text{ !!}} = \underbrace{[A, B]_3(X)}_{\substack{\text{deg 8 with these } A \text{ and } B \\ + \text{ deg 3 systematic factor} \\ + \text{ divisible by } X}}$$

Prob that $[A, B]_3$ factors into deg $\leqslant 3 \Rightarrow 41\%$. Enough !

- Complexity to recover the Dlogs of all degree 3 polynomials:

$$O(\underbrace{(\#\ \boxed{c})}_{q}\underbrace{(\#\ \text{factor base})^2}_{q^2}\underbrace{(\#\ \text{of entries})}_{q}) \approx O(q^6) \text{ ops.}$$

Extend the Factor Base to Degree 4

Our goal: extend the factor base to degree 4

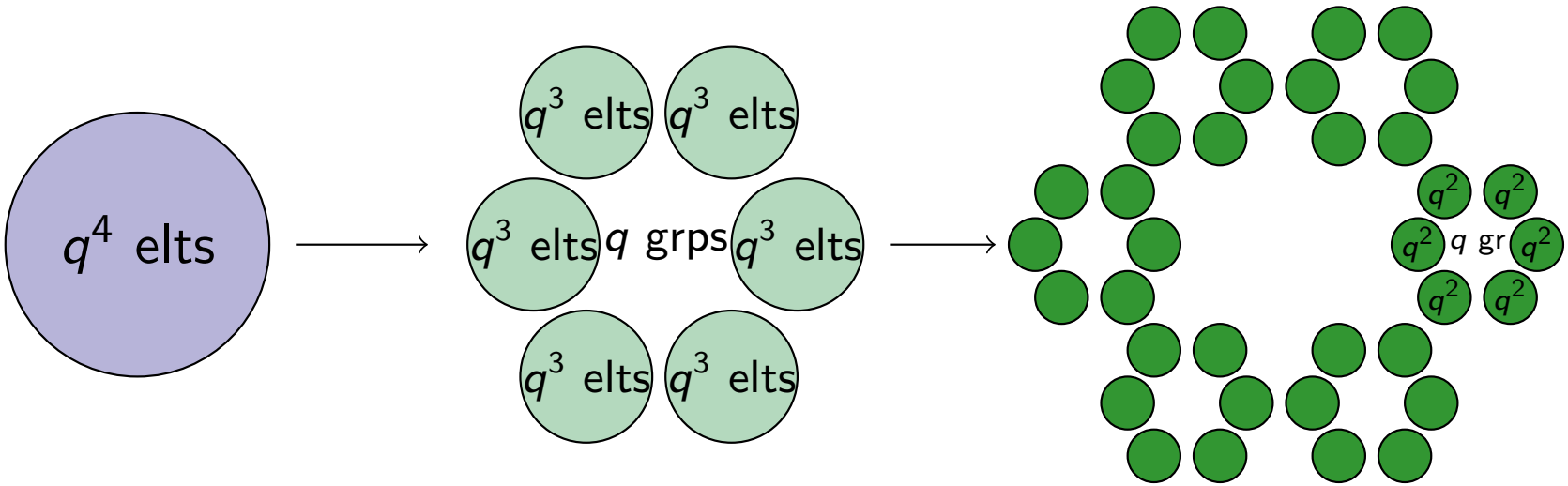
by performing smaller linear algebra steps.

Extend the Factor Base to Degree 4

Our goal: extend the factor base to degree 4

by performing smaller linear algebra steps.

1

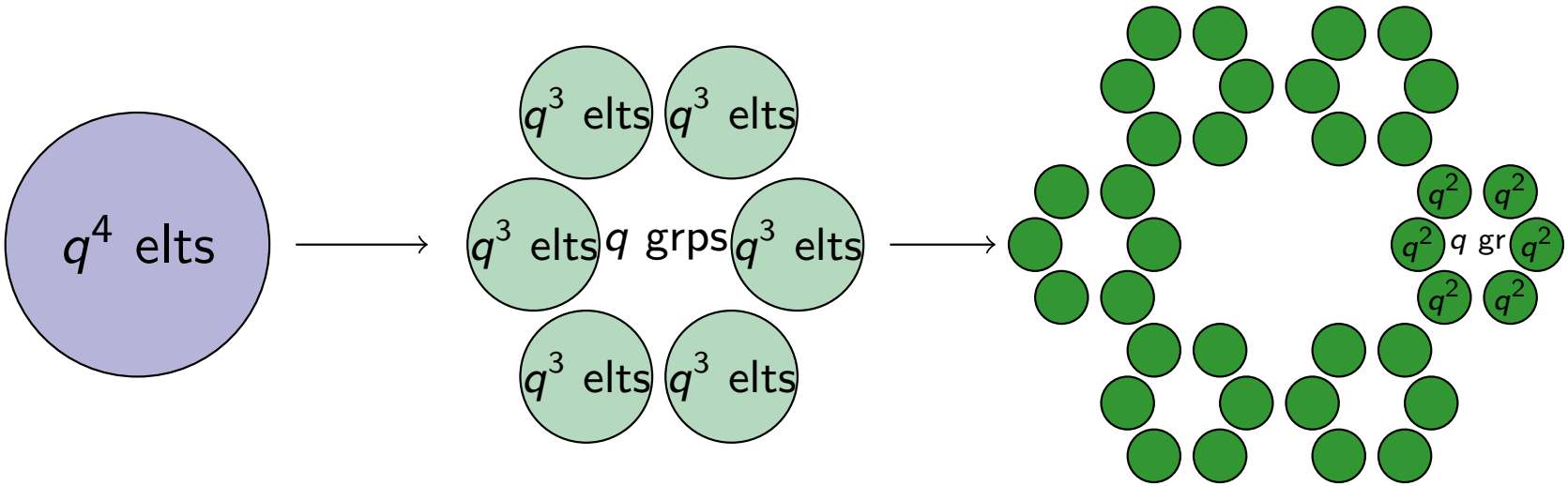


Extend the Factor Base to Degree 4

Our goal: extend the factor base to degree 4

by performing smaller linear algebra steps.

1



What is simple ? To consider that:

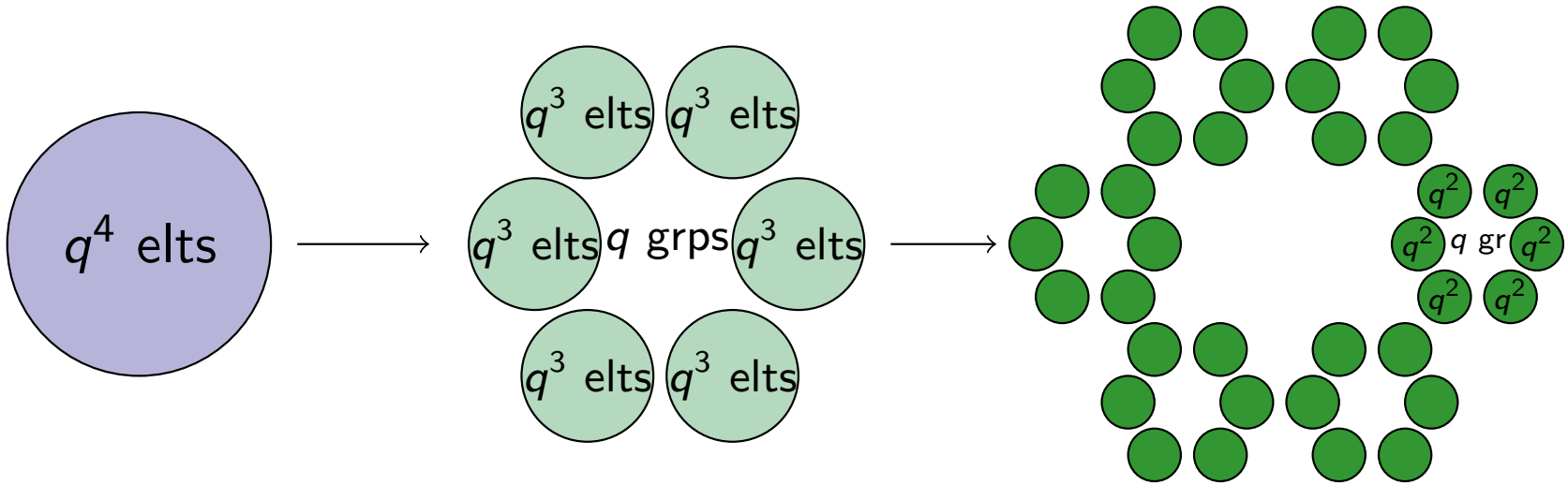
2 poly belongs to the same q^3 if same constant coefficient.

AND 2 poly belongs to the same q^2 if same coeff before X.

Extend the Factor Base to Degree 4

Our goal: extend the factor base to degree 4

1. by performing smaller linear algebra steps.



What is simple ? To consider that:

2 poly belongs to the same q^3 if same constant coefficient.

AND 2 poly belongs to the same q^2 if same coeff before X.

2. Given q^2, generate equations involving only poly in it and degree 1, 2 and 3 polynomials.

Extend the Factor Base to Degree 4

- How ? Previous techniques (bilinear descent from 4 to 3) + additional equations + systematic factors of $[A, B]_4$.

Extend the Factor Base to Degree 4

- How ? Previous techniques (bilinear descent from 4 to 3) + additional equations + systematic factors of $[A, B]_4$.
- Complexity of DLogs computation of ONE q^3:

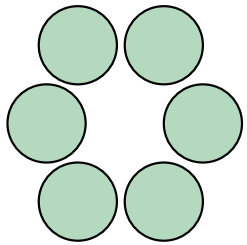
$$O((\underbrace{\#\ q^2 \text{ in } q^3}_{q}) \cdot (\underbrace{\#\ q^2}_{q^2})^2 \underbrace{(\#\text{entries})}_{q}) = O(q^6) \text{ ops.}$$

Extend the Factor Base to Degree 4

- How ? Previous techniques (bilinear descent from 4 to 3) + additional equations + systematic factors of $[A, B]_4$.
- Complexity of DLogs computation of ONE q^3:

$$O((\underbrace{\#\ q^2 \text{ in } q^3}_{q}) \cdot (\underbrace{\#\ q^2}_{q^2})^2 \underbrace{(\#\text{entries})}_{q}) = O(q^6) \text{ ops.}$$

- Final complexity dominated by the first q^3 computation:



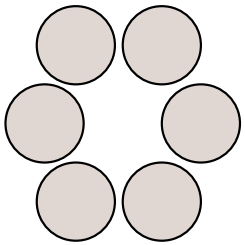
Extend the Factor Base to Degree 4

- How ? Previous techniques (bilinear descent from 4 to 3) + additional equations + systematic factors of $[A, B]_4$.
- Complexity of DLogs computation of ONE q^3:

$$O((\underbrace{\#\ q^2 \text{ in } q^3}_{q}) \cdot (\underbrace{\#\ q^2}_{q^2})^2 \underbrace{(\#\text{entries})}_{q}) = O(q^6) \text{ ops.}$$

- Final complexity dominated by the first q^3 computation:

Unknown



Extend the Factor Base to Degree 4

- How ? Previous techniques (bilinear descent from 4 to 3) + additional equations + systematic factors of $[A, B]_4$.
- Complexity of DLogs computation of ONE q^3:

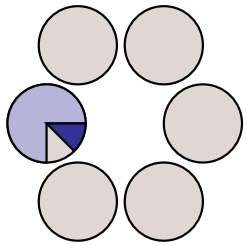
$$O((\underbrace{\#\ q^2 \text{ in } q^3}_{q}) \cdot (\underbrace{\#\ q^2}_{q^2})^2 \underbrace{(\#\text{entries})}_{q}) = O(q^6) \text{ ops.}$$

- Final complexity dominated by the first q^3 computation:

Unknown

Reducible

Bili. desc. $4 \to 3$

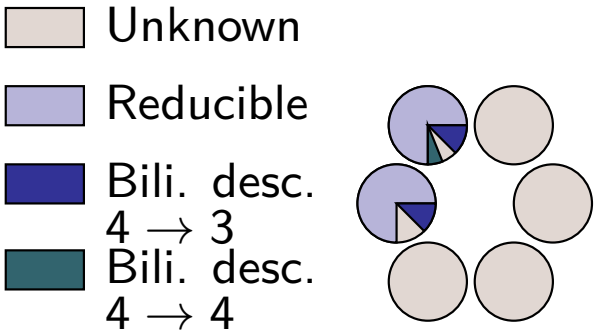


Extend the Factor Base to Degree 4

- How ? Previous techniques (bilinear descent from 4 to 3) + additional equations + systematic factors of $[A, B]_4$.
- Complexity of DLogs computation of ONE q^3:

$$O((\underbrace{\#\ q^2 \text{ in } q^3}_{q}) \cdot (\underbrace{\#\ q^2}_{q^2})^2 \underbrace{(\#\text{entries})}_{q}) = O(q^6) \text{ ops.}$$

- Final complexity dominated by the first q^3 computation:



Extend the Factor Base to Degree 4

- How ? Previous techniques (bilinear descent from 4 to 3) + additional equations + systematic factors of $[A, B]_4$.
- Complexity of DLogs computation of ONE q^3:

$$O(\underbrace{(\#\ q^2 \text{ in } q^3)}_{q} \cdot \underbrace{(\#\ q^2)^2}_{q^2}\ \underbrace{(\#\text{entries})}_{q}) = O(q^6) \text{ ops.}$$

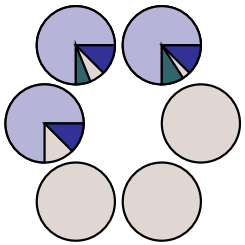
- Final complexity dominated by the first q^3 computation:

Unknown

Reducible

Bili. desc. $4 \to 3$

Bili. desc. $4 \to 4$



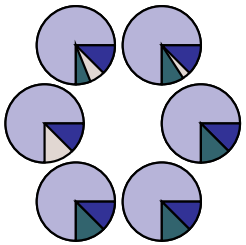
Extend the Factor Base to Degree 4

- How ? Previous techniques (bilinear descent from 4 to 3) + additional equations + systematic factors of $[A, B]_4$.
- Complexity of DLogs computation of ONE q^3:

$$O((\underbrace{\#\ q^2 \text{ in } q^3}_{q}) \cdot (\underbrace{\#\ q^2}_{q^2})^2 \underbrace{(\#\text{entries})}_{q}) = O(q^6) \text{ ops.}$$

- Final complexity dominated by the first q^3 computation:

Unknown
Reducible
Bili. desc. $4 \to 3$
Bili. desc. $4 \to 4$



$\Rightarrow$ Final complexity of extension to deg 4 in $O(q^6)$ operations.

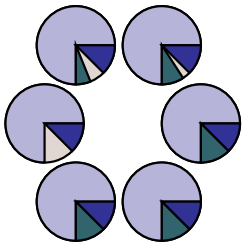
Extend the Factor Base to Degree 4

- How ? Previous techniques (bilinear descent from 4 to 3) + additional equations + systematic factors of $[A, B]_4$.
- Complexity of DLogs computation of ONE q^3:

$$O(\underbrace{(\#\ q^2 \text{ in } q^3)}_{q} \cdot \underbrace{(\#\ q^2)^2}_{q^2} \underbrace{(\#\text{entries})}_{q}) = O(q^6) \text{ ops.}$$

- Final complexity dominated by the first q^3 computation:

Unknown
Reducible
Bili. desc. $4 \to 3$
Bili. desc. $4 \to 4$

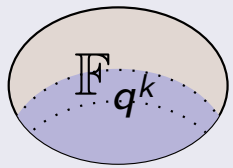


$\Rightarrow$ Final complexity of extension to deg 4 in $O(q^6)$ operations.

Main Result

Final asymptotic complexity of the three first phases:

$O(q^6)$ operations – to be compared with previous $O(q^7)$.



And in practice ?

- New record in characteristic 3 on $\mathbb{F}_{3^{5\cdot 479}}$, a finite field of cardinality a 3796-bit integer.
  - Not a special extension field such as Kummer extension !
  - Make use of the Dual Frobenius Representation combined with the useful variant (both not presented here).
- To be compared with previous record in characteristic 3 by Adj, Menezes, Oliveira and Rodriguez-Henriquez on a 1551-bit finite field.
- Time : 8600 CPU-hours $\approx$ 1 CPU-year

Thank you for your attention !