Efficient multiplication over Extension Fields

Nadia El Mrabet[1] and Nicolas Gama[2]

[1]LIASD, Université Paris 8

[2]PRISM, Université de Versailles

July 18th 2012

Introduction

$$\mathbb{F}_{q^m} = (\mathbb{F}_q[X]/(X^m - \alpha), +, \times)$$

Goal

Make the multiplication efficient

- Lots of Crypto applications:
  - Elliptic curves, pairings, Fully Homomorphic Encryption, ...

Introduction

$$\mathbb{F}_{q^m} = (\mathbb{F}_q[X]/(X^m - \alpha), +, \times)$$

Goal

Make the multiplication efficient

- Lots of Crypto applications:
  - Elliptic curves, pairings, Fully Homomorphic Encryption, ...
- Possible improvements:
  - Multiplication in the base field $\mathbb{F}_q$
  - Multiplication in the extension of degree m

Fast Arithmetic in $\mathbb{F}_{q^m}$

1. We use a non classical representation of finite fields: the AMNS representation
   [Bajard, Imbert, Plantard 2004]
2. We combine it with the FFT multiplication method.
3. We improve previous construction of the AMNS representation.
4. We compare our results with NTL/GMP.

Outline

Choice 1: Polynomial

Example

Multiplication of $A \times B$

- $A = \sum_{i=0}^{m-1} \mathfrak{a}_i X^i$, $B = \sum_{i=0}^{m-1} \mathfrak{b}_i X^i$
- $A \times B \rightarrow \sum_{i=0}^{2m-1} c_i X^i$
- $\mod (X^m - \alpha) \rightarrow \sum_{i=0}^{m-1} c'_i X^i$

Analysis

- $O(m^2)$ generic multiplications in $\mathbb{F}_q$
- $O(m^2)$ additions in $\mathbb{F}_q$

Choice 1: Polynomial

Example

Multiplication of $A \times B$

- $A = \sum_{i=0}^{m-1} \mathfrak{a}_i X^i$, $B = \sum_{i=0}^{m-1} \mathfrak{b}_i X^i$
- $A \times B \rightarrow \sum_{i=0}^{2m-1} c_i X^i$
- $\mod (X^m - \alpha) \rightarrow \sum_{i=0}^{m-1} c'_i X^i$

Analysis

- $O(m^2)$ generic multiplications in $\mathbb{F}_q$
  - Karatsuba $\rightarrow O(m^{1.585})$/Tom Cook $\rightarrow O(m^{1.465})$ for specific m (powers of 2 and 3).
- $O(m^2)$ additions in $\mathbb{F}_q$

Choice 2: DFT

- $A = (A(\gamma^0), A(\gamma^1), \ldots, A(\gamma^{2m-1}))$
  $B = (B(\gamma^0), B(\gamma^1), \ldots, B(\gamma^{2m-1}))$
- $A \times B = (AB(\gamma^0), AB(\gamma^1), \ldots, AB(\gamma^{2m-1}))$
- $i\text{DFT} \rightarrow \sum_{i=0}^{2m-1} c_i X^i$
- $\mod (X^m - \alpha) \rightarrow \sum_{i=0}^{m-1} d_i X^i$
- $\text{DFT} \rightarrow (D(\gamma^0), D(\gamma^1), \ldots, D(\gamma^{2m-1}))$

Analysis

- $O(m^2)$ multiplications in $\mathbb{F}_q$
- $O(m^2)$ additions in $\mathbb{F}_q$
- worse?

DFT Matrices

$$\text{DFT:} \begin{bmatrix} 1 & \cdots & \gamma^{i} & \cdots & \gamma^{(2m-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \gamma^{j} & \cdots & \gamma^{ij} & \cdots & \gamma^{(2m-1)j} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \gamma^{(2m-1)} & \cdots & \gamma^{i(2m-1)} & \cdots & \gamma^{(2m-1)^2} \end{bmatrix}$$

$$\text{iDFT:} \frac{1}{m} \begin{bmatrix} 1 & \cdots & \gamma^{-i} & \cdots & \gamma^{-(2m-1)} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \gamma^{-j} & \cdots & \gamma^{-ij} & \cdots & \gamma^{-(2m-1)j} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \gamma^{-(2m-1)} & \cdots & \gamma^{-i(2m-1)} & \cdots & \gamma^{-(2m-1)^2} \end{bmatrix}$$

Choice 2: DFT

- $A = (A(\gamma^0), A(\gamma^1), \ldots, A(\gamma^{2m-1}))$
  $B = (B(\gamma^0), B(\gamma^1), \ldots, B(\gamma^{2m-1}))$
- $A \times B = (AB(\gamma^0), AB(\gamma^1), \ldots, AB(\gamma^{2m-1}))$
- $i\text{DFT} \rightarrow \sum_{i=0}^{2m-1} c_i X^i$
- $\mod (X^m - \alpha) \rightarrow \sum_{i=0}^{m-1} d_i X^i$
- $\text{DFT} \rightarrow (D(\gamma^0), D(\gamma^1), \ldots, D(\gamma^{2m-1}))$

Analysis

- $O(m)$ generic multiplications in $\mathbb{F}_q$
- $O(m^2)$ multiplications **by powers of** γ in $\mathbb{F}_q$
- $O(m^2)$ additions in $\mathbb{F}_q$

Choice 3: FFT

- $A = (A(\gamma^0), A(\gamma^1), \ldots, A(\gamma^{2m-1}))$
  $B = (B(\gamma^0), B(\gamma^1), \ldots, B(\gamma^{2m-1}))$
- $A \times B = (AB(\gamma^0), AB(\gamma^1), \ldots, AB(\gamma^{2m-1}))$
- $i\text{FFT} \rightarrow \sum_{i=0}^{2m-1} c_i X^i$
- $\mod (X^m - \alpha) \rightarrow \sum_{i=0}^{m-1} d_i X^i$
- $\text{FFT} \rightarrow (D(\gamma^0), D(\gamma^1), \ldots, D(\gamma^{2m-1}))$

Analysis

- $O(m)$ generic multiplications in $\mathbb{F}_q$
- $O(m \cdot \log\, m)$ multiplications **by powers of** γ in $\mathbb{F}_q$
- $O(m \cdot \log\, m)$ additions in $\mathbb{F}_q$

Representation

Question:

Is there a representation of $\mathbb{F}_q$ where multiplication by powers of γ is as cheap as additions?

Representation

Question:

Is there a representation of $\mathbb{F}_q$ where multiplication by powers of γ is as cheap as additions?

Yes

Represent polynomials in γ

Representation of $\mathbb{F}_q$

- Parameters: $\gamma^n = -1 \bmod q$ and $\rho < 2^{32}$
- Representation:
  $(a_0, \ldots, a_{n-1}) = \sum_{i=0}^{n-1} a_i \gamma^i \bmod q$
  $\|(a_0, \ldots, a_{n-1})\|_\infty \leq \rho$

Operations

$\times\gamma^k$: Fast (rotation/negation of coefficients)

$+, -$: pairwise addition.

$\times$: OK.

$/$: Slow (must evaluate)

$==$: Slow (must evaluate)

Representation of $\mathbb{F}_q$

- Parameters: $\gamma^n = -1 \bmod q$ and $\rho < 2^{32}$
- Representation:
  $(a_0, \ldots, a_{n-1}) = \sum_{i=0}^{n-1} a_i \gamma^i \bmod q$
  $\|(a_0, \ldots, a_{n-1})\|_\infty \leq \rho$

Operations

$\times\gamma^k$: Fast (rotation/negation of coefficients)

$+, -$: pairwise addition. **Coefs may increase** (2ρ)

$\times$: OK. **Coefs increase a lot** $(n\rho^2)$

$/$: Slow (must evaluate)

$==$: Slow (must evaluate)

Representation of $\mathbb{F}_q$

- Parameters: $\gamma^n = -1 \bmod q$ and $\rho < 2^{32}$
- Representation:
  $(a_0, \ldots, a_{n-1}) = \sum_{i=0}^{n-1} a_i \gamma^i \bmod q$
  $\|(a_0, \ldots, a_{n-1})\|_\infty \leq \rho$

Operations

$\times\gamma^k$: Fast (rotation/negation of coefficients)

$+,-$: pairwise addition. **Coefs may increase** (2ρ)

$\times$: OK. **Coefs increase a lot** $(n\rho^2)$

$/$: Slow (must evaluate)

$==$: Slow (must evaluate)

ReduceCoefs: $(n\rho^2 \to \rho)$

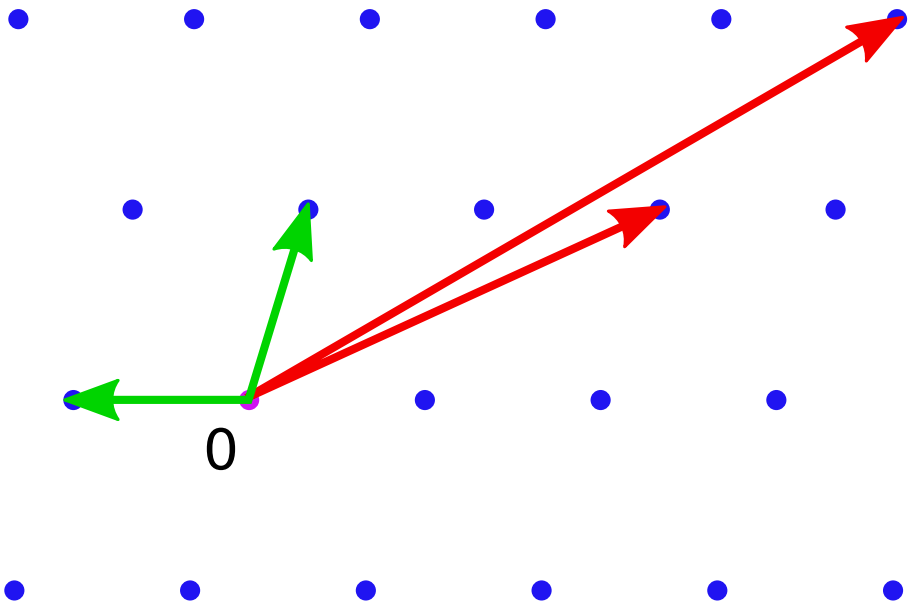
ReduceCoefs

ReduceCoefs

- **Input:**
  $(a_0, \ldots, a_{n-1}) \in [-n\rho^2, n\rho^2]^n$
- **Output:**
  $(a'_0, \ldots, a'_{n-1}) \in [-\rho, \rho]^n$
  such that
  $\sum_{i=0}^{n-1} a_i \gamma^i = \sum_{i=0}^{n-1} a'_i \gamma^i \bmod q$

It looks like a Closest Vector Problem

Lattices

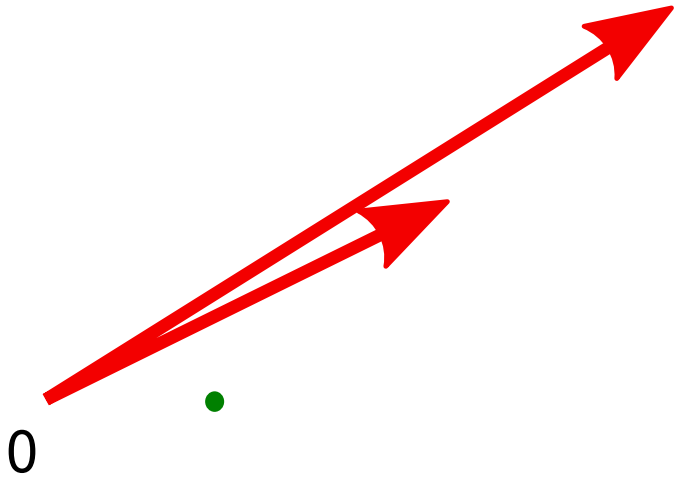


Definitions

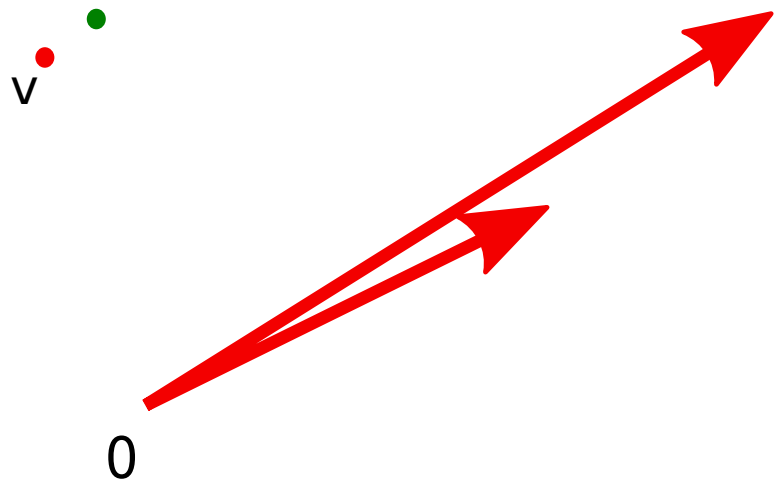
- Lattice
- Basis

SVP/CVP

SVP: Given a lattice basis B, find the shortest non-zero vector of $\mathcal{L}(B)$

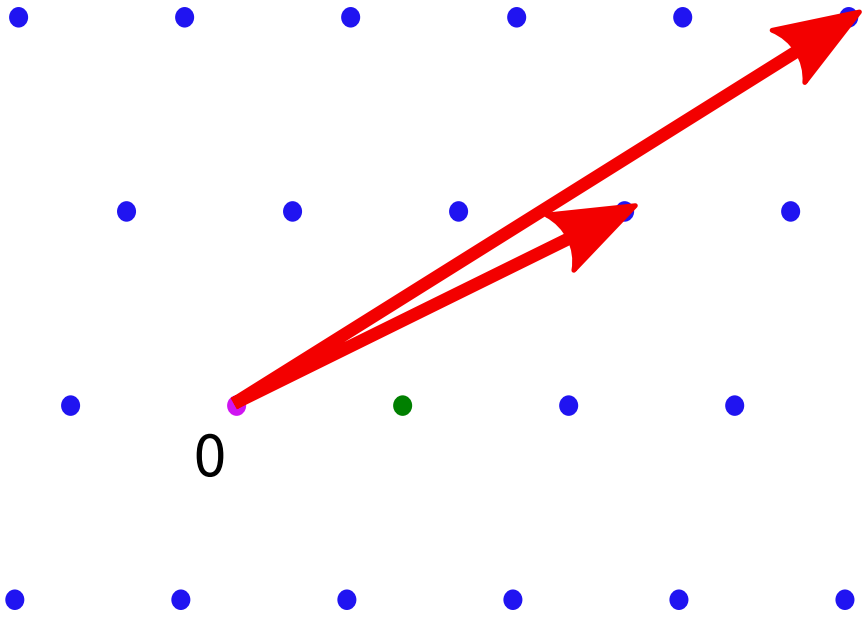


CVP: Given a lattice basis $B \in \mathcal{M}_{n,m}(\mathbb{Z})$ and a target vector $\mathfrak{v} \in \mathbb{R}^m$, find the lattice vector of $\mathcal{L}(B)$ closest to $\mathfrak{v}$

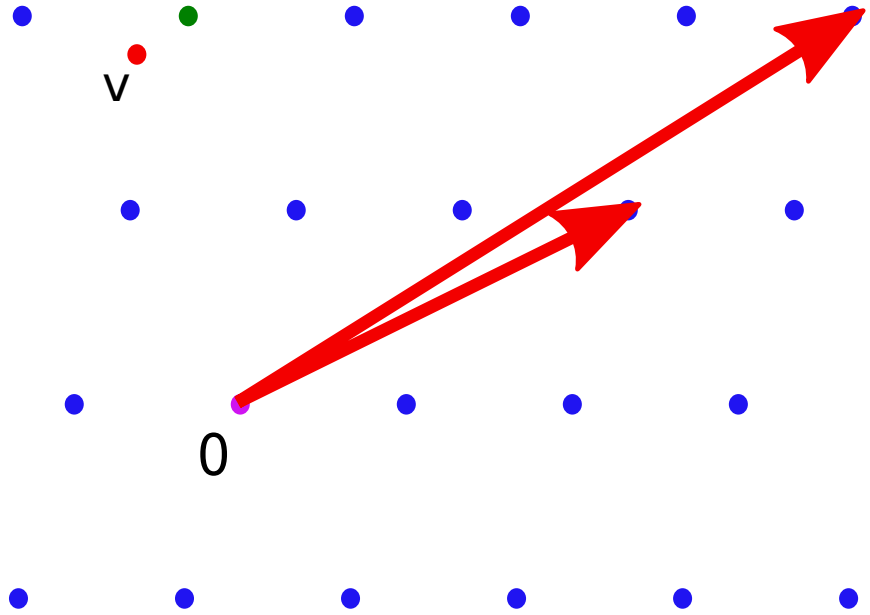


SVP/CVP

SVP: Given a lattice basis B, find the shortest non-zero vector of $\mathcal{L}(B)$



CVP: Given a lattice basis $B \in \mathcal{M}_{n,m}(\mathbb{Z})$ and a target vector $\mathfrak{v} \in \mathbb{R}^m$, find the lattice vector of $\mathcal{L}(B)$ closest to $\mathfrak{v}$



Our problem

- Lattice:
  $\mathcal{L} = \{(x_0, \ldots, x_{n-1}) \in \mathbb{Z}^n \text{ s.t. } \sum_{i=1}^{n} x_i \gamma^i = 0 \bmod q\}$
- HNF basis:
$$\begin{bmatrix} q & 0 & \cdots & 0 \\ -\gamma & 1 & \ddots & \vdots \\ \vdots & 0 & \ddots & 0 \\ -\gamma^{n-1} & 0 & 0 & 1 \end{bmatrix}$$
- CVP Target: $\mathfrak{a} = (a_0, \ldots, a_{n-1})$

RedCoefs:

$\mathfrak{c} \leftarrow \mathrm{CVP}(\mathfrak{a}, \mathcal{L})$
$\mathfrak{a}' = \mathfrak{a} - \mathfrak{c}$ is the AMNS representative of $\mathfrak{a}$

Bad news

- ✘ CVP is NP-Hard

RedCoefs:

$\mathfrak{c} \leftarrow \mathbf{Approx}\mathrm{CVP}(\mathfrak{a}, \mathcal{L})$
$\mathfrak{a}' = \mathfrak{a} - \mathfrak{c}$ is the AMNS representative of $\mathfrak{a}$

Bad news

✘ CVP is NP-Hard

Good news

✔ ApproxCVP is easy with a trapdoor:

- a short basis of $\mathcal{L}$
- simply one short vector, since $\mathcal{L}$ is anti-cyclic.

✔ The trapdoor can be precomputed

Complexity

Let $\mathfrak{m}$ a short vector of $\mathcal{L}$

Highest Significant bits

- $\mathrm{CVP}(\mathfrak{c}, \mathcal{L}) \approx \lfloor c \cdot \mathfrak{m}^{-1} \rceil \cdot \mathfrak{m}$
- $\mathfrak{c} - \mathrm{CVP}(\mathfrak{c}, \mathcal{L})$ is small $\rightarrow$AMNS of c
- $\mathfrak{m}^{-1} \bmod q$ has large entries!
  (use floating point computation?)

Least Significant bits

- $\mathrm{CVP}'(\mathfrak{c}, \mathcal{L}) \approx (c \cdot \mathfrak{m}^{-1} \bmod \phi) \cdot \mathfrak{m}$
- $\mathfrak{c} - \mathrm{CVP}'(\mathfrak{c}, \mathcal{L})$ is divisible by ϕ
  $\Longrightarrow$ $\frac{1}{\phi}(\mathfrak{c} - \mathrm{CVP}'(\mathfrak{c}, \mathcal{L}))$ is small $\rightarrow$AMNS of $\frac{c}{\phi}$

AMNS Representation of $\mathbb{F}_q$

AMNS Parameters

- $\gamma^n = -1 \bmod q$,
- a radius ρ,
- a small $\mathfrak{m}$ such that $\mathfrak{m}(\gamma) = 0 \bmod q$,
- a modulus ϕ and $\mathfrak{m}\mathfrak{I} = \mathfrak{m}^{-1} \bmod \phi$.

AMNS Representation:

- $(a_0, \ldots, a_{n-1}) = \sum_{i=0}^{n-1} a_i \gamma^i \bmod q$
  $\|(a_0, \ldots, a_{n-1})\|_\infty \leq \rho$

AMNS Operations

Fast AMNS Operations

$\times\gamma^k$: Fast (rotation/negation of coefficients)

$+,-$: pairwise addition. **Coefs may increase** (2ρ)

$\times$: OK. **Coefs increase a lot** $(n\rho^2)$

$/\phi$: **ReduceCoefs:** $(n\rho^2 \rightarrow \rho)$

$a \times b/\phi$: OK. (Montgomery multiplication)

AMNS Operations

Fast AMNS Operations

$\times\gamma^k$: Fast (rotation/negation of coefficients)

$+,-$: pairwise addition. **Coefs may increase** (2ρ)

$\times$: OK. **Coefs increase a lot** $(n\rho^2)$

$/\phi$: **ReduceCoefs:** $(n\rho^2 \to \rho)$

$a \times b/\phi$: OK. (Montgomery multiplication)

Slow operations

$/$: Slow (bezout or evaluate)

$==$: Slow (evaluate)

Optimization

1. γ should be a n-th root of -1 (so $2n$ must divide $q-1$)
2. Every number must have an AMNS representation:
   $\rho >$covering radius of $\mathcal{L} \approx q^{1/n}$
3. Computations should fit in raw integer types:
   $2n\rho^2 \leq 2^{64}$ or 2^{128}
4. $\mathfrak{m}$ must be a very short vector of $\mathcal{L}$
   and must be invertible $\mod \phi$.
5. ϕ must be large enough to reduce coefficients: $2n\rho^2/\phi \leq \rho$
6. ϕ should be a power of 2
   (there are a lot of exact divisions by ϕ)

Combining it together

Is it possible to satisfy all these constraints? [BIP2004]

1. choose γ and p
2. Reduce the lattice $\mathcal{L}$ to find a short vector $\mathfrak{m}$
   $\rightarrow \rho = n \left\| \mathfrak{m} \right\|_\infty$
3. Choose ϕ prime with $\mathrm{Res}(\mathfrak{m}, X^n + 1)$
4. Compute $\mathfrak{m}^{-1} \bmod \phi$

Combining it together

Is it possible to satisfy all these constraints? [BIP2004]

1. choose γ and p
2. Reduce the lattice $\mathcal{L}$ to find a short vector $\mathfrak{m}$
   $\rightarrow \rho = n \left\|\mathfrak{m}\right\|_\infty$
3. Choose ϕ prime with $\mathrm{Res}(\mathfrak{m}, X^n + 1)$
4. Compute $\mathfrak{m}^{-1} \bmod \phi$

Questions

- ✘ problem of lattice reduction (high dimension)
  - poly lattice reduction $\rightarrow$ exponential ρ
  - exp lattice reduction $\rightarrow$ tight ρ
- ✘ ϕ may not be a power of 2 (if $\mathrm{Res}(\mathfrak{m}, X^n + 1)$ is even)

Optimal parameters

Theorem

Every basis of $\mathcal{L}$ contains at least a vector $\mathfrak{m}$ such that $\mathrm{Res}(\mathfrak{m}, X^n + 1)$ is odd

Consequences

- From any short basis, one can extract $\mathfrak{m}$ such that $\mathfrak{m}$ invertible mod powers of 2.
- ϕ can always be a power of 2.

New Construction 1

Construction 1

1. choose γ and q, n, and $\phi = 2^k$
2. Reduce the lattice $\mathcal{L}$ to find a short basis
3. extract a short vector $\mathfrak{m}$ invertible $\mod \phi$

New Construction 1

Construction 1

1. choose γ and q, n, and $\phi = 2^k$
2. Reduce the lattice $\mathcal{L}$ to find a short basis
3. extract a short vector $\mathfrak{m}$ invertible $\mod \phi$

Remarks

- ✔ ϕ is a power of 2
- ✘ Still an expensive lattice reduction

New Construction 2

Construction 2

1. choose $\phi = 2^k$, ρ and n
2. choose a small vector $\mathfrak{m}$ at random in $[0 - \rho]^n$.
3. Compute $\mathrm{Res}(\mathfrak{m}, X^n + 1)$, repeat until it is prime or contains a large prime factor q.
4. compute γ a $2n$-th root of q

Remarks

- ✔ ϕ is a power of 2
- ✔ poly-time construction, no more lattice reduction
- q cannot be freely chosen
  (Can be problematic for some crypto constructions)

Implementation of $\times$

Implementations of $(\mathbb{F}_{q^m}, \times)$

NTL

- Uses GMP for large integer operations
- Has ZZ_pE module when the extension is fixed in advance (allows preprocessing).

AMNS

- raw C/C++ implementation (int64, int32)
- DFT and FFT
- Consistency checks (in debug version)

Complexity

Implementations of $(\mathbb{F}_{q^m}, \times)$

The complexity of NTL increases with

- the base field size (q)
- the degree of extension (m)

The complexity of AMNS increases with

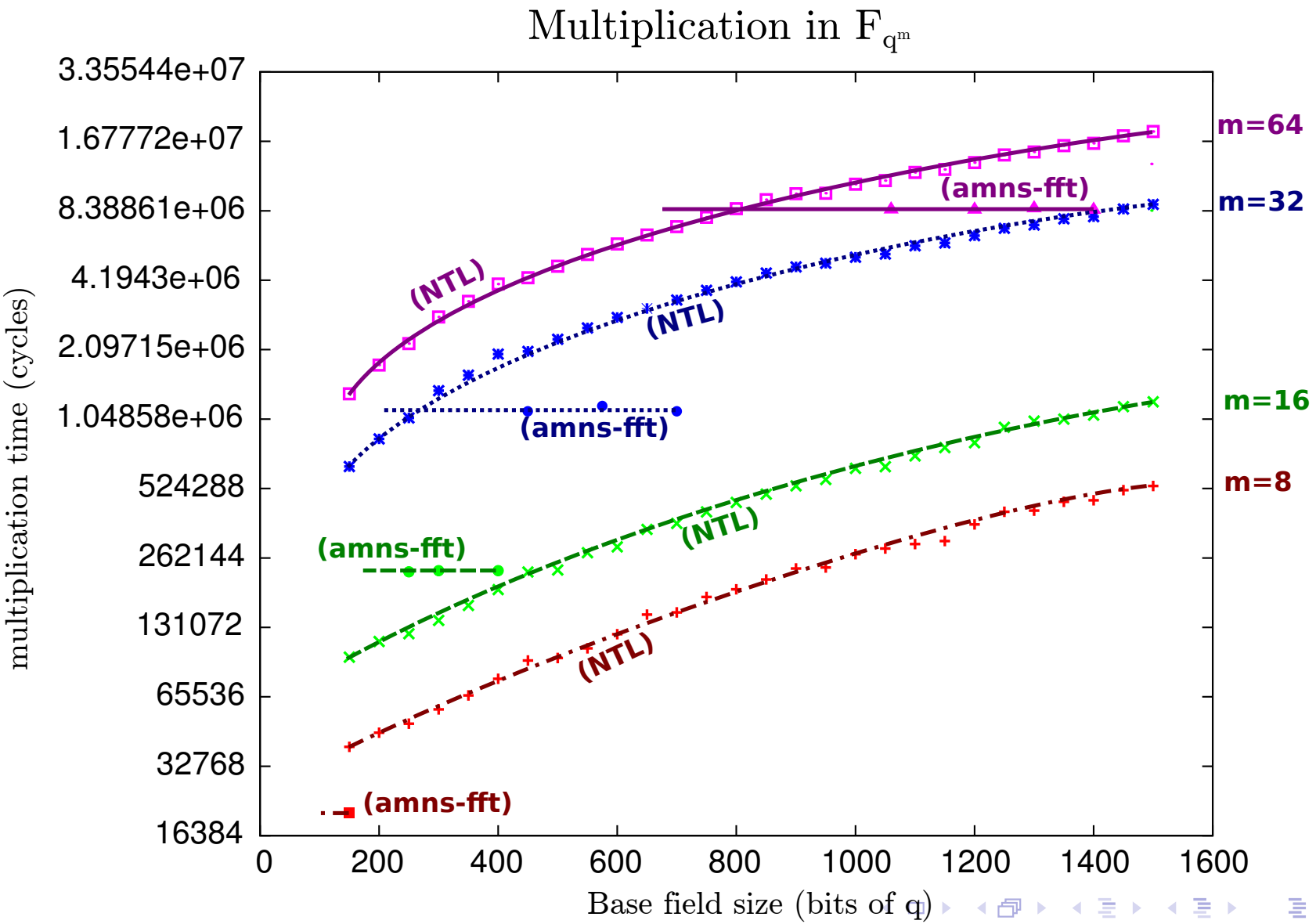
- The dimension (n)
  But there is a maximum (q, m) which can be represented in dim n

Choice of parameters

Choice of parameters

- $n =$ power of 2 (for FFT)
- $\rho \approx \phi \lessapprox 2^{32}/\sqrt{2n}$ (native int64 on 32-bit machines)
- $m \leq n \rightarrow$ take the equality
- $q \leq \rho^n \rightarrow$ take as high as possible

Experimental Timings



Conclusion

- We proposed efficient routines to efficiently construct AMNS bases
  - An easy one when the characteristic can be freely chosen
  - A slower one when q is constrained
- We proposed the first implementation which runs faster than GMP over large fields

Open questions

- fast equality test or inversion?
- Implement a full cryptosystem using AMNS?