PhD defense — October 30th, 2013

Algorithms and arithmetic for the implementation of cryptographic pairings

Nicolas Estibals
CARAMEL project-team, LORIA, Université de Lorraine / CNRS / INRIA, France
Nicolas.Estibals@loria.fr
What is an elliptic curve?

\[ E/K : y^2 + h(x)y = f(x) \]
with \( \deg h \leq 1 \) and \( \deg f = 3 \)
What is an elliptic curve?

- Set of points $E(K)$ is a group

$E/K : y^2 + h(x)y = f(x)$
with $\deg h \leq 1$ and $\deg f = 3$
What is an elliptic curve?

Set of points $E(K)$ is a group

$E/K: y^2 + h(x)y = f(x)$

with $\deg h \leq 1$ and $\deg f = 3$
What is an elliptic curve?

- Set of points $E(K)$ is a group

$E/K : y^2 + h(x)y = f(x)$
with $\deg h \leq 1$ and $\deg f = 3$
What is an elliptic curve?

- Set of points $E(K)$ is a group

$E/K : y^2 + h(x)y = f(x)$
with $\deg h \leq 1$ and $\deg f = 3$
What is an elliptic curve?

- Set of points $E(K)$ is a group

$E/K : y^2 + h(x)y = f(x)$
with $\deg h \leq 1$ and $\deg f = 3$
What is an elliptic curve?

- Set of points $E(K)$ is a group
- In practice: $K$ is a finite field $\mathbb{F}_q$
- $E(\mathbb{F}_q)$ is a finite group

$$E/K : y^2 + h(x)y = f(x)$$
with $\deg h \leq 1$ and $\deg f = 3$
What is an elliptic curve?

- Set of points $E(K)$ is a group
- In practice: $K$ is a finite field $\mathbb{F}_q$
- $E(\mathbb{F}_q)$ is a finite group
- $[n]P = P + \cdots + P$ \(n\) times

$E/K : y^2 + h(x)y = f(x)$ with $\deg h \leq 1$ and $\deg f = 3$
What is an elliptic curve?

- Set of points $E(K)$ is a group
- In practice: $K$ is a finite field $\mathbb{F}_q$
- $E(\mathbb{F}_q)$ is a finite group
- $[n]P = P + \cdots + P$ $n$ times
- $\ell$: a large prime dividing $\#E(\mathbb{F}_q)$
- Use a cyclic subgroup of
  
  $E[\ell] = \{ P \mid [\ell]P = O \}$

$E/K : y^2 + h(x)y = f(x)$
with $\deg h \leq 1$ and $\deg f = 3$
What is an elliptic curve?

- Set of points $E(K)$ is a group
- In practice: $K$ is a finite field $\mathbb{F}_q$
- $E(\mathbb{F}_q)$ is a finite group
- $[n]P = P + \cdots + P$ (n times)
- $\ell$: a large prime dividing $\#E(\mathbb{F}_q)$
- Use a cyclic subgroup of $E[\ell] = \{P \mid [\ell]P = \mathcal{O}\}$

Our favorite curves: $E_3 : y^2 = x^3 - x \pm 1$
- characteristic 3
- supersingular
Discrete Logarithm Problem (DLP)

Let $\mathbb{G}$ be a cyclic group, $P$ a generator, given $Q \in \mathbb{G}$, it is supposed to be hard to compute $a$ such that

$$Q = [a]P$$
Elliptic Curve Cryptography

Discrete Logarithm Problem (DLP)
Let $\mathbb{G}$ be a cyclic group, $P$ a generator, given $Q \in \mathbb{G}$, it is supposed to be hard to compute $a$ such that

$$Q = [a]P$$

- Use this hard problem to design cryptographic protocols
- **Diffie–Hellman** key exchange:
  - Alice generates a secret integer $a$
  - Alice sends $[a]P$ to Bob
  - Alice computes $[a][b]P$
  - Bob generates a secret integer $b$
  - Bob sends $[b]P$ to Alice
  - Bob computes $[b][a]P$

  They both share the same secret: $[ab]P$
What is a pairing?

Pairing

$E[\ell]$

Pairing

$e(\ldots)$

$\ell$-th roots of unity

$\{u | u^\ell = 1\} \subset F_q$

Bilinear map:

$e(P + P', Q) = e(P, Q) \cdot e(P', Q)$

Cryptographic interest:

Mixing two secrets without having to know them

$e([a]P, [b]Q) = e(P, Q)^{ab}$

Useful for advanced protocols

• short signature
• electronic voting
• electronic money

DLP should be hard on all the groups involved
What is a pairing?

A pairing $e(\cdot, \cdot)$ is a bilinear map:

$$e(\cdot, \cdot) : E[\ell] \times E[\ell] \rightarrow \{ u \mid u^\ell = 1 \} \subset \overline{F_q}$$

- $\ell$-th roots of unity

Cryptographic interest:
- Mixing two secrets without knowing them
- $e([a]P, [b]Q) = e(P, Q)^{ab}$
- Useful for advanced protocols
  - short signature
  - electronic voting
  - electronic money
  - ...

DLP should be hard on all the groups involved.
What is a pairing?

A pairing is a bilinear map:

\[ e(P + P', Q) = e(P, Q) \cdot e(P', Q) \]

\[ e(P, Q + Q') = e(P, Q) \cdot e(P, Q') \]

pairing \( e(., .) \)

\( E[\ell] \)

\( \{ u \mid u^\ell = 1 \} \subset F_q \)

\( \ell \)-th roots of unity

Bilinear map:
What is a pairing?

\[ e(., .) \]
\[ E[\ell] \]
\[ \{ u \mid u^\ell = 1 \} \subseteq \mathbb{F}_q \]
\[ \ell \)-th roots of unity

**Bilinear map:**
\[ e(P + P', Q) = e(P, Q) \cdot e(P', Q) \]
\[ e(P, Q + Q') = e(P, Q) \cdot e(P, Q') \]

**Cryptographic interest:** Mixing two secrets without having to know them
\[ e([a]P, [b]Q) = e(P, Q)^{ab} \]
What is a pairing?

- **Bilinear map:**
  
  \[ e(P + P', Q) = e(P, Q) \cdot e(P', Q) \]
  
  \[ e(P, Q + Q') = e(P, Q) \cdot e(P, Q') \]

- **Cryptographic interest:** *Mixing two secrets without having to know them*

  \[ e([a]P, [b]Q) = e(P, Q)^{ab} \]

- **Useful for advanced protocols**
  - short signature
  - electronic voting
  - electronic money
  - ...
What is a pairing?

- **Bilinear map:**
  
  \[
  e(P + P', Q) = e(P, Q) \cdot e(P', Q) \\
  e(P, Q + Q') = e(P, Q) \cdot e(P, Q')
  \]

- **Cryptographic interest:** *Mixing two secrets without having to know them*
  
  \[
  e([a]P, [b]Q) = e(P, Q)^{ab}
  \]

- **Useful for advanced protocols**
  
  - short signature
  - electronic voting
  - electronic money
  - ...

- **DLP** should be hard on all the groups involved
Security considerations

- Security measurement
  - number of operations to break a cryptosystem
  - today’s recommendation: 128-bit security
    \[2^{128}\] operations
Security considerations

▶ Security measurement
  - number of operations to break a cryptosystem
  - today’s recommendation: 128-bit security
    \[ 2^{128} \text{ operations} \]

▶ Difficulty of the DLP on the curve
  - depends on the order \( \ell \)
  - roughly \( \sqrt{\ell} \) operations

\[ \ell \approx 2^{697} \]
\[ \sqrt{\ell} \approx 2^{349} \]

\[ 2 \cdot 509 = 1,018 \]

\[ \text{embedding degree}: k = 6 \]
\[ (F_{3^6}\cdot 509)^* \]

Subexponential algorithms exist
- function field sieve
  \[ \approx 2^{132} \text{ operations} \]
- very recent results (2013)
  Records by Joux and Göloğlu et al.
  Joux, Barbulescu, Gaudry, Joux, Thomé
  Adj, Menezes, Oliveira, Rodríguez-Henríquez
  \[ \approx 2^{75} \text{ operations} \]

For our favorite curve \( E_{3} \) over \( F_{3}^{509} \)
Security considerations

▶ Security measurement
  • number of operations to break a cryptosystem
  • today’s recommendation: 128-bit security
    \[ 2^{128} \text{ operations} \]

▶ Difficulty of the DLP on the curve
  • depends on the order \( \ell \)
  • roughly \( \sqrt{\ell} \) operations

For our favorite curve
\[ E_3 \text{ over } \mathbb{F}_{3509} \]
\[ \ell \approx 2^{697} \]
\[ \approx 2^{349} \text{ operations} \]
Security considerations

▶ Security measurement
  • number of operations to break a cryptosystem
  • today’s recommendation: 128-bit security
    \[2^{128}\] operations

▶ Difficulty of the DLP on the curve
  • depends on the order \(\ell\)
  • roughly \(\sqrt{\ell}\) operations

\[\ell \approx 2^{697}\]
\[\approx 2^{349}\] operations

▶ Difficulty of the DLP on the roots of unity
  • embedding degree: \(k\) such that all roots lie in \(\mathbb{F}_{q^k}\)
  • \(k = 6\), so DLP in \((\mathbb{F}_{36509})^*\)

For our favorite curve \(E_3\) over \(\mathbb{F}_{3509}\)
Security considerations

- **Security measurement**
  - number of operations to break a cryptosystem
  - today’s recommendation: 128-bit security
    
    $2^{128}$ operations

- **Difficulty of the DLP on the curve**
  - depends on the order $\ell$
  - roughly $\sqrt{\ell}$ operations
    
    $\ell \approx 2^{697}$
    
    $\approx 2^{349}$ operations

- **Difficulty of the DLP on the roots of unity**
  - embedding degree: $k$ such that all roots lie in $\mathbb{F}_{q^k}$
  - Subexponential algorithms exist
    
    ★ function field sieve
    
    $k = 6$, so DLP in $(\mathbb{F}_{3^{609}})^*$
    
    $\approx 2^{132}$ operations

For our favorite curve $E_3$ over $\mathbb{F}_{3509}$
Security considerations

- **Security measurement**
  - number of operations to break a cryptosystem
  - today’s recommendation: 128-bit security
    \[2^{128}\text{ operations}\]

- **Difficulty of the DLP on the curve**
  - depends on the order \(\ell\)
  - roughly \(\sqrt{\ell}\) operations
    \[\ell \approx 2^{697}\]
    \[\approx 2^{349}\text{ operations}\]

- **Difficulty of the DLP on the roots of unity**
  - embedding degree: \(k\) such that all roots lie in \(\mathbb{F}_{q^k}\)
  - Subexponential algorithms exist
    - function field sieve
    - very recent results (2013)
      - Records by Joux and Göloğlu et al. records
        - Joux
        - Barbulescu, Gaudry, Joux, Thomé
        - Adj, Menezes, Oliveira, Rodríguez-Henríquez
      \[\approx 2^{75}\text{ operations}\]
Why cryptography and hardware implementations?

- Growth of numeric exchanges
  - many applications
    - bank services
    - secure firmware updates
    - personal communications
    - ...
  - many targets
    - embedded electronics
    - smart cards
    - smartphones
    - computers, servers

- Security implies non-trivial computations

- Need for hardware implementations
  - CPUs may be inadequate
  - limited resources
Our target: Field Programmable Gate Array (FPGA)

- integrated circuit
- matrix of simple configurable logic cells
- programmable interconnection

Performance metric

- time (ms)
- area (slices)
- time–area product

Different designs for the same computation

- optimized for latency
- optimized for compactness
- optimized for throughput
Our target: **Field Programmable Gate Array (FPGA)**

- integrated circuit
- matrix of *simple* configurable logic cells
- programmable interconnection

Performance metric

- *time* (ms)
Hardware implementation

► Our target: Field Programmable Gate Array (FPGA)
  • integrated circuit
  • matrix of simple configurable logic cells
  • programmable interconnection

► Performance metric
  • time (ms)
  • area (slices)
Hardware implementation

► Our target: Field Programmable Gate Array (FPGA)
  • integrated circuit
  • matrix of simple configurable logic cells
  • programmable interconnection

► Performance metric
  • time (ms)
  • area (slices)

► Different designs for the same computation
  • optimized for latency
  • optimized for compactness
Our target: **Field Programmable Gate Array (FPGA)**
- integrated circuit
- matrix of *simple* configurable logic cells
- programmable interconnection

**Performance metric**
- time (ms)
- area (slices)

**Different designs** for the same computation
- optimized for latency
- optimized for compactness
Hardware implementation

Our target: Field Programmable Gate Array (FPGA)
- integrated circuit
- matrix of simple configurable logic cells
- programmable interconnection

Performance metric
- time (ms)
- area (slices)

Different designs for the same computation
- optimized for latency
- optimized for compactness

![Diagram of FPGA]

Area

Computation time
Our target: Field Programmable Gate Array (FPGA)
- integrated circuit
- matrix of simple configurable logic cells
- programmable interconnection

Performance metric
- time (ms)
- area (slices)

Different designs for the same computation
- optimized for latency
- optimized for compactness
Hardware implementation

- Our target: Field Programmable Gate Array (FPGA)
  - integrated circuit
  - matrix of simple configurable logic cells
  - programmable interconnection

- Performance metric
  - time (ms)
  - area (slices)
  - time–area product

- Different designs for the same computation
  - optimized for latency
  - optimized for compactness
  - optimized for throughput
Contributions

- **Fast accelerator** for pairings [CHES 2009, IEEE TC 2011]
  
  *Joint work with Beuchat, Detrey, Okamoto and Rodríguez-Henríquez*

  - parallel architecture
  - pipelined subquadratic multiplier

- **Compact design** for pairings reaching 128-bit security

  - composite extension fields [Paring 2010]
  - hyperelliptic curves [CT-RSA 2012]

  *Joint work with Aranha, Beuchat and Detrey*

- **Formulae** for sub-quadratic multiplication [WAIFI 2012]

  *Joint work with Barbulescu, Detrey and Zimmermann*

  - exhaustive search
  - improved formulae for $\mathbb{F}_{3^{5m}}$
Contributions

▶ **Fast accelerator** for pairings [CHES 2009, IEEE TC 2011]
  
  *Joint work with Beuchat, Detrey, Okamoto and Rodríguez-Henríquez*

  - parallel architecture
  - pipelined subquadratic multiplier

▶ **Compact design** for pairings reaching 128-bit security

  - composite extension fields [Paring 2010]
  - hyperelliptic curves [CT-RSA 2012]

  *Joint work with Aranha, Beuchat and Detrey*

▶ **Formulae for sub-quadratic multiplication** [WAIFI 2012]

  *Joint work with Barbulescu, Detrey and Zimmermann*

  - exhaustive search
  - improved formulae for $\mathbb{F}_{35m}$
Outline of the talk

▶ Compact design through composite extension fields

▶ Pairing on genus-2 hyperelliptic curves

▶ Searching for efficient multiplication algorithms

▶ Conclusion and Perspectives
Computing the pairing: Miller’s algorithm

Computation of the pairing relies on
Miller functions: $f_{n,P}$
Computing the pairing: Miller’s algorithm

- Computation of the pairing relies on
  Miller functions: $f_{n,P}$

  - An inductive identity defined by
    
    \[
    f_{1,P} = 1
    \]
    
    \[
    f_{n+n',P} = f_{n,P} \cdot f_{n',P} \cdot g[n]P, [n']P
    \]
Computing the pairing: Miller’s algorithm

- Computation of the pairing relies on Miller functions: $f_{n,P}$
  - an inductive identity defined by
    
    $f_{1,P} = 1$
    $f_{n+n',P} = f_{n,P} \cdot f_{n',P} \cdot g[n]P,[n']P$
  
  - $g[n]P,[n']P$ derived from the addition of $[n]P$ and $[n']P$

![Diagram of Miller's algorithm for computing the pairing](image)
Computing the pairing: Miller’s algorithm

- Computation of the pairing relies on Miller functions: $f_{n,P}$
  - an inductive identity defined by
    
    $f_{1,P} = 1$
    $f_{n+n',P} = f_{n,P} \cdot f_{n',P} \cdot g[n]P, [n']P$

  - $g[n]P, [n']P$ derived from the addition of $[n]P$ and $[n']P$

- Tate pairing: $f_{\#E(\mathbb{F}_q), P}$
  - use an addition chain
  - in practice: double-and-add
    
    $\log_2 \#E(\mathbb{F}_q)$ iterations
Computing the pairing: Miller’s algorithm

Computation of the pairing relies on Miller functions: \( f_{n,P} \)

- an inductive identity defined by
  \[
  f_{1,P} = 1 \\
  f_{n+n',P} = f_{n,P} \cdot f_{n',P} \cdot g[n]P, [n']P
  \]

- \( g[n]P, [n']P \) derived from the addition of \([n]P\) and \([n']P\)

Tate pairing: \( f_{\#E(\mathbb{F}_q),P} \)

- use an addition chain
- in practice: double-and-add
  \[
  \log_2 \#E(\mathbb{F}_q) \text{ iterations}
  \]

<table>
<thead>
<tr>
<th>For ( E_3(\mathbb{F}_{3509}) )</th>
<th>Tate pairing</th>
</tr>
</thead>
<tbody>
<tr>
<td># iterations</td>
<td>509</td>
</tr>
<tr>
<td>\times</td>
<td>10330</td>
</tr>
<tr>
<td>+</td>
<td>45170</td>
</tr>
<tr>
<td>( (.)^3 )</td>
<td>8136</td>
</tr>
<tr>
<td>( (.)^{-1} )</td>
<td>2</td>
</tr>
</tbody>
</table>

For \( E_3(\mathbb{F}_{3509}) \)

\[
\# E_3(\mathbb{F}_{3509}) = 3^{509} + 3^{255} + 1
\]

- triple-and-add algorithm
Computing the pairing: Miller’s algorithm

- Computation of the pairing relies on Miller functions: $f_{n,P}$
  - an inductive identity defined by
    \[ f_{1,P} = 1 \]
    \[ f_{n+n',P} = f_{n,P} \cdot f_{n',P} \cdot g[n]P[n']P \]
  - $g[n]P[n']P$ derived from the addition of $[n]P$ and $[n']P$

- Tate pairing: $f_{\#E(F_q),P}$
  - use an addition chain
  - in practice: double-and-add
    \[ \log_2 \#E(F_q) \text{ iterations} \]

<table>
<thead>
<tr>
<th>For $E_3(F_{3509})$</th>
<th>Tate pairing</th>
<th>Eta T</th>
</tr>
</thead>
<tbody>
<tr>
<td># iterations</td>
<td>509</td>
<td>254</td>
</tr>
<tr>
<td>$\times$</td>
<td>10330</td>
<td>3638</td>
</tr>
<tr>
<td>$+$</td>
<td>45170</td>
<td>17240</td>
</tr>
<tr>
<td>$(.)^3$</td>
<td>8136</td>
<td>4068</td>
</tr>
<tr>
<td>$(.)^{-1}$</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

- $\#E_3(F_{3509}) = 3^{509} + 3^{255} + 1$
  - triple-and-add algorithm

- Many improvements
  - vertical elimination
  - use of some curve endomorphisms
    - Frobenius: Ate
    - Verschiebung: Eta, Eta T
An arithmetic coprocessor

- Only need arithmetic operations in $\mathbb{F}_{3^{509}}$
  - implement a specialized processor

- Multiplication is critical
  - separate linear operations and multiplications
  - careful scheduling to keep multiplier busy

<table>
<thead>
<tr>
<th>Operation count</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\times$</td>
<td>3638</td>
</tr>
<tr>
<td>$+$</td>
<td>17240</td>
</tr>
<tr>
<td>$(\cdot)^3$</td>
<td>4068</td>
</tr>
<tr>
<td>$(\cdot)^{-1}$</td>
<td>1</td>
</tr>
</tbody>
</table>
An arithmetic coprocessor

- Only need arithmetic operations in $\mathbb{F}_{3^{509}}$
  - implement a specialized processor

- Multiplication is critical
  - separate linear operations and multiplications
  - careful scheduling to keep multiplier busy

- Inverse is only needed once: Itoh–Tsujii algorithm
  - no need for hardware support

<table>
<thead>
<tr>
<th>Operation count</th>
<th>$\times$</th>
<th>3638</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$+$</td>
<td>17240</td>
</tr>
<tr>
<td></td>
<td>$(\cdot)^3$</td>
<td>4068</td>
</tr>
<tr>
<td></td>
<td>$(\cdot)^{-1}$</td>
<td>1</td>
</tr>
</tbody>
</table>
An arithmetic coprocessor

Only need arithmetic operations in $\mathbb{F}_{3^{509}}$
- implement a specialized processor

Multiplication is critical
- separate linear operations and multiplications
- careful scheduling to keep multiplier busy

Inverse is only needed once: Itoh–Tsujii algorithm
- no need for hardware support

Synthesis results for $\mathbb{F}_{3^{509}}$: 9625 slices
- almost fully occupy a Virtex 6 LX 75 T (82%)
- computation time: $\approx 4$ ms

Operation count

\[
\begin{array}{c|c}
\times & 3638 \\
+ & 17240 \\
(\cdot)^3 & 4068 \\
(\cdot)^{-1} & 1 \\
\end{array}
\]
Field of composite extension degree

\[ F_{3^{6\cdot509}} \]

Software

Hardware

\[ F_{3^{5\cdot97}} \]

\[ F_{3^{6\cdot5\cdot97}} \]

- Provides some arithmetic advantages
  - Smaller datapath
  - Efficient multiplication algorithm
- Allows Weil Descent based attacks on the curve
  - GHS: Using the composite extension degree \( \approx 2^{279} \) operations
  - SDHP: Granger's algorithm \( \approx 2^{142} \) operations
  - Limited effect on security
- Results
  - 1848 slices of the same Virtex 6 LX (15%)
  - 2.5 times smaller
  - Compute a pairing in 1.6 ms
    - 2.5 times faster
Field of composite extension degree

- Provides some arithmetic advantages
  - smaller datapath

\[
\begin{align*}
F_3 & \quad \rightarrow \quad F_{3 \cdot 5 \cdot 97} \\
\rightarrow & \quad \text{Software} \quad \rightarrow \quad F_{3 \cdot 509} \\
\rightarrow & \quad \text{Hardware} \quad \rightarrow \quad F_{3 \cdot 5 \cdot 97} \\
\end{align*}
\]
Field of composite extension degree

- Provides some arithmetic advantages
  - smaller datapath
  - efficient multiplication algorithm

\[
F_{3^{6 \cdot 509}}, \quad F_{3^{6 \cdot 5 \cdot 97}}
\]

Subquadratic multiplication

Quadratic multiplication

Software

Hardware
**Field of composite extension degree**

- Provides some *arithmetic advantages*
  - smaller datapath
  - efficient multiplication algorithm

- Allows *Weil Descent based attacks* on the curve
  - GHS: using the composite extension degree
    \[
    \approx 2^{279} \text{ operations}
    \]
  - SDHP: Granger’s algorithm
    \[
    \approx 2^{142} \text{ operations}
    \]
  - limited effect on security
Field of composite extension degree

- Provides some arithmetic advantages
  - smaller datapath
  - efficient multiplication algorithm

- Allows Weil Descent based attacks on the curve
  - GHS: using the composite extension degree
    \[ \approx 2^{279} \ \text{operations} \]
  - SDHP: Granger’s algorithm
    \[ \approx 2^{142} \ \text{operations} \]
  - limited effect on security

- Results
  - 1848 slices of the same Virtex 6 LX (15%)
    5.2 times smaller
  - compute a pairing in 1.6 ms
    2.5 times faster
Pairing implementations at 128 bits of security on Virtex 6
Benchmarks

Pairing implementations at 128 bits of security on Virtex 6

- $F_{3509}$

Diagram showing computation time versus area for different implementations.

- [Est10]

N. Estibals — Algorithms and arithmetic for the implementation of cryptographic pairings
Pairing implementations at 128 bits of security on Virtex 6

Computation time [ms]

Area $[\times 10^3 \text{ slices}]$
Pairing implementations at 128 bits of security on Virtex 6
Pairing implementations at 128 bits of security on Virtex 6

- [Est10]
- [FVV12]
- [GVR13]
- [Che+11]
- [Yao+13]
- [AHN13]
- [GRD11]
- [AHN13]
Outline of the talk

- Compact design through composite extension fields
- Pairing on genus-2 hyperelliptic curves
- Searching for efficient multiplication algorithms
- Conclusion and Perspectives
Genus-2 hyperelliptic curves

\[ C/K : y^2 + h(x)y = f(x) \]
with \( \deg h \leq 2 \) and \( \deg f = 5 \)
Genus-2 hyperelliptic curves

$C(K)$ not a group!

$C/K : y^2 + h(x)y = f(x)$
with $\deg h \leq 2$ and $\deg f = 5$
Genus-2 hyperelliptic curves

$C(K)$ not a group!

But pairs of points $\{P_1, P_2\}$

$C/K : y^2 + h(x)y = f(x)$
with $\deg h \leq 2$ and $\deg f = 5$
Genus-2 hyperelliptic curves

- $C(K)$ not a group!
- But pairs of points $\{P_1, P_2\}$

$$C/K : y^2 + h(x)y = f(x)$$
with $\deg h \leq 2$ and $\deg f = 5$
Genus-2 hyperelliptic curves

- $C(K)$ not a group!
- But pairs of points $\{P_1, P_2\}$

\[ C/K : y^2 + h(x)y = f(x) \]
with $\deg h \leq 2$ and $\deg f = 5$
Genus-2 hyperelliptic curves

- $C(K)$ not a group!

- But pairs of points \( \{P_1, P_2\} \)

\[
C/K : y^2 + h(x)y = f(x)
\]
with \( \deg h \leq 2 \) and \( \deg f = 5 \)

\[
\{P_1, P_2\} + \{Q_1, Q_2\} = \{R_1, R_2\}
\]
Genus-2 hyperelliptic curves

- $C(K)$ not a group!
- But pairs of points $\{P_1, P_2\}$
- More formally
  - Jacobian of the curve $\text{Jac}_C$
  - is a group

$C/K : y^2 + h(x)y = f(x)$
with $\deg h \leq 2$ and $\deg f = 5$

$\{P_1, P_2\} + \{Q_1, Q_2\} = \{R_1, R_2\}$
Genus-2 hyperelliptic curves

- $C(K)$ not a group!

- But pairs of points \( \{ P_1, P_2 \} \)

- More formally
  - Jacobian of the curve $\text{Jac}_C$
  - is a group

- Chosen curves
  - $H_2 : y^2 + y = x^5 + x^3 + d$, with $d \in \{0, 1\}$
  - characteristic 2
  - supersingular

\[ C/K : y^2 + h(x)y = f(x) \]
with $\deg h \leq 2$ and $\deg f = 5$

\[ \{ P_1, P_2 \} + \{ Q_1, Q_2 \} = \{ R_1, R_2 \} \]
Optimal Eta

Parameters for 128-bit security

- Embedding degree $k = 12$
- Field: $\mathbb{F}_{2^{367}}$
Optimal Eta

- Parameters for 128-bit security
  - Embedding degree $k = 12$
  - Field: $\mathbb{F}_{2^{367}}$
  - $\# \text{Jac}_C(\mathbb{F}_{2^{367}}) = 2^{734} - 2^{551} - 2^{367} + 2^{184} + 1$

- Our pairing algorithm

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Tate (double-and-add)</th>
</tr>
</thead>
<tbody>
<tr>
<td># iterations</td>
<td>734</td>
</tr>
</tbody>
</table>
Optimal Eta

► Parameters for 128-bit security
  • Embedding degree $k = 12$
  • Field: $\mathbb{F}_{2^{367}}$
  • $\# \text{Jac}_C(\mathbb{F}_{2^{367}}) = 4 \cdot 8^{244} - 4 \cdot 2^{183} - 2 \cdot 8^{122} + 1$

► Our pairing algorithm
  • Efficient octupling formula: octuple-and-add

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Tate (double-and-add)</th>
<th>Tate (octuple-and-add)</th>
</tr>
</thead>
<tbody>
<tr>
<td># iterations</td>
<td>734</td>
<td>245</td>
</tr>
</tbody>
</table>
Optimal Eta

- Parameters for 128-bit security
  - Embedding degree $k = 12$
  - Field: $\mathbb{F}_{2^{367}}$
  - $\# \text{Jac}_C(\mathbb{F}_{2^{367}}) = 4 \cdot 8^{244} - 4 \cdot 2^{183} - 2 \cdot 8^{122} + 1$

- Our pairing algorithm
  - Efficient octupling formula: octuple-and-add
  - Adapted Verschiebung: Eta $T$

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Tate (double-and-add)</th>
<th>Tate (octuple-and-add)</th>
<th>Eta $T$</th>
</tr>
</thead>
<tbody>
<tr>
<td># iterations</td>
<td>734</td>
<td>245</td>
<td>184</td>
</tr>
</tbody>
</table>
Optimal Eta

▸ Parameters for 128-bit security
  • Embedding degree $k = 12$
  • Field: $\mathbb{F}_{2^{367}}$
  • $\# \text{Jac}_C(\mathbb{F}_{2^{367}}) = 4 \cdot 8^{244} - 4 \cdot 2^{183} - 2 \cdot 8^{122} + 1$

▸ Our pairing algorithm
  • Efficient octupling formula: octuple-and-add
  • adapted Verschiebung: Eta T
  • Vercauteren’s optimal technique: optimal Eta

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Tate (double-and-add)</th>
<th>Tate (octuple-and-add)</th>
<th>Eta T</th>
<th>Optimal Eta</th>
</tr>
</thead>
<tbody>
<tr>
<td># iterations</td>
<td>734</td>
<td>245</td>
<td>184</td>
<td>123</td>
</tr>
</tbody>
</table>
Optimal Eta

- Parameters for 128-bit security
  - Embedding degree $k = 12$
  - Field: $\mathbb{F}_{2^{367}}$
  - $\# \text{Jac}_C(\mathbb{F}_{2^{367}}) = 4 \cdot 8^{244} - 4 \cdot 2^{183} - 2 \cdot 8^{122} + 1$

- Our pairing algorithm
  - Efficient octupling formula: octuple-and-add
  - adapted Verschiebung: Eta T
  - Vercauteren’s optimal technique: optimal Eta

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Tate (double-and-add)</th>
<th>Tate (octuple-and-add)</th>
<th>Eta T</th>
<th>Optimal Eta</th>
</tr>
</thead>
<tbody>
<tr>
<td># iterations</td>
<td>734</td>
<td>245</td>
<td>184</td>
<td>123</td>
</tr>
</tbody>
</table>

- Implementation on the previous coprocessor adapted for $\mathbb{F}_{2^{367}}$
  - 1366 slices on the same Virtex 6 LX (12%)
  - 3.2 ms
  - comparable performances with the elliptic case
Benchmarks

Pairing implementations at 128 bits of security on Virtex 6

N. Estibals — Algorithms and arithmetic for the implementation of cryptographic pairings
Outline of the talk

- Compact design through composite extension fields
- Pairing on genus-2 hyperelliptic curves
- Searching for efficient multiplication algorithms
- Conclusion and Perspectives
Origin of the problem

▶ Polynomial multiplication is an expensive arithmetic operation

▶ Schoolbook algorithm: quadratic cost
Origin of the problem

- Polynomial multiplication is an expensive arithmetic operation
- Schoolbook algorithm: quadratic cost
- Karatsuba (1962): first subquadratic multiplication algorithm
  \[(a_0 + a_1X)(b_0 + b_1X) = a_0b_0 + (a_0b_1 + a_1b_0)X + a_1b_1X^2\]

- Well-studied problem
  - asymptotic complexity
  - theoretical bilinear complexity
  - small and "cryptographic" size
  - ad-hoc formulae
  - exhaustive search for five-term multiplication
  - non-exhaustive search for six and seven-term multiplications

- Our approach: improve the search algorithm
Origin of the problem

- Polynomial multiplication is an expensive arithmetic operation
- Schoolbook algorithm: quadratic cost
- Karatsuba (1962): first subquadratic multiplication algorithm
  \[(a_0 + a_1X)(b_0 + b_1X) = a_0b_0 + (a_0b_1 + a_1b_0)X + a_1b_1X^2\]
  \[= a_0b_0 + ((a_0 + a_1)(b_0 + b_1) - a_0b_0 - a_1b_1)X + a_1b_1X^2\]
Origin of the problem

- Polynomial multiplication is an **expensive** arithmetic operation

- **Schoolbook algorithm**: quadratic cost

- **Karatsuba (1962)**: first subquadratic multiplication algorithm

\[(a_0 + a_1 X)(b_0 + b_1 X) = a_0 b_0 + (a_0 b_1 + a_1 b_0)X + a_1 b_1 X^2\]

\[= a_0 b_0 + ((a_0 + a_1)(b_0 + b_1) - a_0 b_0 - a_1 b_1)X + a_1 b_1 X^2\]

- Well-studied problem
  - asymptotic complexity
  - theoretical bilinear complexity
  - small and “cryptographic” size

  - ad-hoc formulae
  - exhaustive search for five-term multiplication
  - non-exhaustive search for six and seven-term multiplications

- Our approach: **improve the search algorithm**
Generalization of the problem

▶ Model of a multiplication algorithm

\[ a_{n-1}X^{n-1} + a_1X + a_0 \]

\[ b_{n-1}X^{n-1} + b_1X + b_0 \]

Linear combinations

Product
Generalization of the problem

▶ **Model of a multiplication algorithm**

\[
a_{n-1}X^{n-1} + a_1X + a_0
\]

\[
b_{n-1}X^{n-1} + b_1X + b_0
\]

**Linear combinations**

**Product**

▶ Also true for any bilinear application

- multiplication in extension fields
- sparse products
- matrix multiplications
- . . .
Formal framework

Formulation in terms of vector space for an \( n \times m \) multiplication over a given field \( K \)

- Represent the coefficients of the result and the products as elements of

\[
V \text{ the } nm\text{-dimensional } K\text{-vector space generated by } \{a_i b_j\}_{0 \leq i < n, 0 \leq j < m}
\]

where the \( a_i b_j \)'s are seen as formal elements
Formal framework

Formulation in terms of vector space for an $n \times m$ multiplication over a given field $K$

- Represent the coefficients of the result and the products as elements of $V$ the $nm$-dimensional $K$-vector space generated by \{a_i b_j\}_{0 \leq i < n, 0 \leq j < m}$ where the $a_i b_j$'s are seen as formal elements

- Our target: the coefficients of the result is a family $\mathcal{T} \subset V$ that spans the target subspace $T = \text{Span} \mathcal{T}$ of $V$
Formal framework

Formulation in terms of vector space for an $n \times m$ multiplication over a given field $K$

- Represent the coefficients of the result and the products as elements of $V$ the $nm$-dimensional $K$-vector space generated by $\{a_ib_j\}_{0 \leq i < n, 0 \leq j < m}$

where the $a_ib_j$’s are seen as formal elements

- Our target: the coefficients of the result is a family $T \subset V$ that spans the target subspace $T = \text{Span } T$ of $V$

- The set $G$ of the potential products to use in a formula: the generators
Formal framework

Formulation in terms of vector space for an $n \times m$ multiplication over a given field $K$

- Represent the coefficients of the result and the products as elements of $V$ the $nm$-dimensional $K$-vector space generated by $\{a_ib_j\}_{0 \leq i < n, 0 \leq j < m}$ where the $a_ib_j$'s are seen as formal elements

- Our target: the coefficients of the result is a family $T \subset V$ that spans the target subspace $T = \text{Span} \ T$ of $V$

- The set $G$ of the potential products to use in a formula: the generators

- Goal: find the optimal formulae (i.e. with a minimum number of products)
  - for increasing $k$ until a solution is found
  - find each subset $W \subset G$ of exactly $k$ products
  - which gives a valid formula (i.e. that linearly generates the coefficients of the result)

$$T \subset \text{Span} \ W$$
**Resolution**

- **Naive** approach: test each subset of $k$ potential products

```latex
\text{expand\_family}(\emptyset, \mathcal{G})

\textbf{procedure} expand\_family(\mathcal{W}, \mathcal{H})
  \textbf{if} \#\mathcal{W} = k \textbf{ then}
    \textbf{if} \mathcal{T} \subset \text{Span} \mathcal{W} \textbf{ then}
      \mathcal{W} \text{ is a solution}
    \textbf{else}

    \textbf{while} \mathcal{H} \neq \emptyset \textbf{ do}
      \text{Pick a } h \text{ in } \mathcal{H}
      \mathcal{H} \leftarrow \mathcal{H} \setminus \{h\}
      \text{expand\_family}(\mathcal{W} \cup \{h\}, \mathcal{H})
  \textbf{end procedure}
```

- **Complexity** depends on

$$\binom{\#\mathcal{G}}{k}$$
Resolution

- Naive approach: test each subset of \( k \) potential products

- Better approach: test each vector space of dimension \( k \) generated by potential products

```plaintext
procedure expand_subspace(W, \( \mathcal{H} \))
    if \( \dim W = k \) then
        if \( T \subset W \) then
            \( W \) is a solution
        end if
    else
        \( \mathcal{H} \leftarrow \mathcal{H} \setminus W \)
        while \( \mathcal{H} \neq \emptyset \) do
            Pick a \( h \) in \( \mathcal{H} \)
            \( \mathcal{H} \leftarrow \mathcal{H} \setminus \{h\} \)
            expand_subspace(\( W \oplus \text{Span}(h), \mathcal{H} \))
        end while
    end if
end procedure
```

- Complexity still depends on
  \[
  \binom{\#G}{k}
  \]
Resolution

- **Naive** approach: test each subset of $k$ potential products

- **Better** approach: test each vector space of dimension $k$ generated by potential products

- **Even better** approach: part of the solution is already known, use incomplete basis theorem

```plaintext
Expand_subspace($T, G$)

Procedure expand_subspace($W, H$)
  if dim $W = k$ then
    if rank($W \cap G$) = $k$ then
      $W$ is a solution
    else
      $H \leftarrow H \setminus W$
      while $H \neq \emptyset$ do
        Pick a $h$ in $H$
        $H \leftarrow H \setminus \{h\}$
        expand_subspace($W \oplus \text{Span}(h), H$)
  end procedure

- Complexity now depends on

\[
\left( \begin{array}{c}
\#G \\
\binom{k - \text{rank} T}
\end{array} \right)
\]
### Some results

- **Multiplication of** \( n \times m \) **term binary polynomials**

<table>
<thead>
<tr>
<th>Ring</th>
<th>( n \times m )</th>
<th>#G</th>
<th>( k )</th>
<th># of tests</th>
<th># of solutions</th>
<th># of formulae</th>
<th>Computation time (1 core)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathbb{F}_2[X] )</td>
<td>2 \times 2</td>
<td>9</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>3 \times 3</td>
<td>49</td>
<td>6</td>
<td>9</td>
<td>3</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>4 \times 4</td>
<td>225</td>
<td>9</td>
<td>6.60 \cdot 10^3</td>
<td>4</td>
<td>4</td>
<td>30 ms</td>
</tr>
<tr>
<td></td>
<td>5 \times 5</td>
<td>961</td>
<td>13</td>
<td>9.65 \cdot 10^9</td>
<td>27</td>
<td>27</td>
<td>2 d 15 h</td>
</tr>
<tr>
<td></td>
<td>6 \times 6</td>
<td>3969</td>
<td>14</td>
<td>4.37 \cdot 10^9</td>
<td>—</td>
<td>—</td>
<td>7 d</td>
</tr>
<tr>
<td></td>
<td>(Sym.) 6 \times 6</td>
<td>63</td>
<td>17</td>
<td>8.08 \cdot 10^6</td>
<td>6</td>
<td>54</td>
<td>18 s</td>
</tr>
<tr>
<td></td>
<td>(Sym.) 7 \times 7</td>
<td>127</td>
<td>22</td>
<td>3.38 \cdot 10^{12}</td>
<td>2618</td>
<td>19550</td>
<td>184 d</td>
</tr>
</tbody>
</table>
## Some results

- Multiplication of $n \times m$ term binary polynomials

<table>
<thead>
<tr>
<th>Ring</th>
<th>$n \times m$</th>
<th>$#G$</th>
<th>$k$</th>
<th>$#$ of tests</th>
<th>$#$ of solutions</th>
<th>$#$ of formulae</th>
<th>Computation time (1 core)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_2[X]$</td>
<td>$2 \times 2$</td>
<td>9</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>$3 \times 3$</td>
<td>49</td>
<td>6</td>
<td>9</td>
<td>3</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>$4 \times 4$</td>
<td>225</td>
<td>9</td>
<td>$6.60 \cdot 10^3$</td>
<td>4</td>
<td>4</td>
<td>30 ms</td>
</tr>
<tr>
<td></td>
<td>$5 \times 5$</td>
<td>961</td>
<td>13</td>
<td>$9.65 \cdot 10^9$</td>
<td>27</td>
<td>27</td>
<td>2 d 15 h</td>
</tr>
<tr>
<td></td>
<td>$6 \times 6$</td>
<td>3969</td>
<td>14</td>
<td>$4.37 \cdot 10^9$</td>
<td>—</td>
<td>—</td>
<td>7 d</td>
</tr>
<tr>
<td></td>
<td>$6 \times 6$ (Sym.)</td>
<td>63</td>
<td>17</td>
<td>$8.08 \cdot 10^6$</td>
<td>6</td>
<td>54</td>
<td>18 s</td>
</tr>
<tr>
<td></td>
<td>$7 \times 7$ (Sym.)</td>
<td>127</td>
<td>22</td>
<td>$3.38 \cdot 10^{12}$</td>
<td>2618</td>
<td>19,550</td>
<td>184 d</td>
</tr>
</tbody>
</table>
Some results

- Multiplication of $n \times m$ term binary polynomials

<table>
<thead>
<tr>
<th>Ring</th>
<th>$n \times m$</th>
<th>$# G$</th>
<th>$k$</th>
<th>$#$ of tests</th>
<th>$#$ of solutions</th>
<th>$#$ of formulae</th>
<th>Computation time (1 core)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbb{F}_2[X]$</td>
<td>2 × 2</td>
<td>9</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>3 × 3</td>
<td>49</td>
<td>6</td>
<td>9</td>
<td>3</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>4 × 4</td>
<td>225</td>
<td>9</td>
<td>$6.60 \cdot 10^3$</td>
<td>4</td>
<td>4</td>
<td>30 ms</td>
</tr>
<tr>
<td></td>
<td>5 × 5</td>
<td>961</td>
<td>13</td>
<td>$9.65 \cdot 10^9$</td>
<td>27</td>
<td>27</td>
<td>2 d 15 h</td>
</tr>
<tr>
<td></td>
<td>6 × 6</td>
<td>3 969</td>
<td>14</td>
<td>$4.37 \cdot 10^9$</td>
<td>—</td>
<td>—</td>
<td>7 d</td>
</tr>
<tr>
<td></td>
<td>6 × 6 (Sym.)</td>
<td>63</td>
<td>17</td>
<td>$8.08 \cdot 10^6$</td>
<td>6</td>
<td>54</td>
<td>18 s</td>
</tr>
<tr>
<td></td>
<td>7 × 7 (Sym.)</td>
<td>127</td>
<td>22</td>
<td>$3.38 \cdot 10^{12}$</td>
<td>2 618</td>
<td>19 550</td>
<td>184 d</td>
</tr>
</tbody>
</table>
Some results

- Multiplication of $n \times m$ term binary polynomials

<table>
<thead>
<tr>
<th>Ring</th>
<th>$n \times m$</th>
<th>#G</th>
<th>$k$</th>
<th># of tests</th>
<th># of solutions</th>
<th># of formulae</th>
<th>Computation time (1 core)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$F_2[X]$</td>
<td>2 × 2</td>
<td>9</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>3 × 3</td>
<td>49</td>
<td>6</td>
<td>9</td>
<td>3</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>4 × 4</td>
<td>225</td>
<td>9</td>
<td>$6.60 \cdot 10^3$</td>
<td>4</td>
<td>4</td>
<td>30 ms</td>
</tr>
<tr>
<td></td>
<td>5 × 5</td>
<td>961</td>
<td>13</td>
<td>$9.65 \cdot 10^9$</td>
<td>27</td>
<td>27</td>
<td>2 d 15 h</td>
</tr>
<tr>
<td></td>
<td>6 × 6</td>
<td>3 969</td>
<td>14</td>
<td>$4.37 \cdot 10^9$</td>
<td>—</td>
<td>—</td>
<td>7 d</td>
</tr>
<tr>
<td></td>
<td>(Sym.) 63</td>
<td></td>
<td>17</td>
<td>$8.08 \cdot 10^6$</td>
<td>6</td>
<td>54</td>
<td>18 s</td>
</tr>
<tr>
<td></td>
<td>(Sym.) 127</td>
<td></td>
<td>22</td>
<td>$3.38 \cdot 10^{12}$</td>
<td>2 618</td>
<td>19 550</td>
<td>184 d</td>
</tr>
</tbody>
</table>
Some results

- Multiplication of $n \times m$ term binary polynomials

<table>
<thead>
<tr>
<th>Ring</th>
<th>$n \times m$</th>
<th>$#G$</th>
<th>$k$</th>
<th>$#$ of tests</th>
<th>$#$ of solutions</th>
<th>$#$ of formulae</th>
<th>Computation time (1 core)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbb{F}_2[X]$</td>
<td>2 $\times$ 2</td>
<td>9</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>3 $\times$ 3</td>
<td>49</td>
<td>6</td>
<td>9</td>
<td>3</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>4 $\times$ 4</td>
<td>225</td>
<td>9</td>
<td>$6.60 \cdot 10^3$</td>
<td>4</td>
<td>4</td>
<td>30 ms</td>
</tr>
<tr>
<td></td>
<td>5 $\times$ 5</td>
<td>961</td>
<td>13</td>
<td>$9.65 \cdot 10^9$</td>
<td>27</td>
<td>27</td>
<td>2 d 15 h</td>
</tr>
<tr>
<td></td>
<td>6 $\times$ 6</td>
<td>3969</td>
<td>14</td>
<td>$4.37 \cdot 10^9$</td>
<td>—</td>
<td>—</td>
<td>7 d</td>
</tr>
<tr>
<td></td>
<td>6 $\times$ 6 (Sym.)</td>
<td>63</td>
<td>17</td>
<td>$8.08 \cdot 10^6$</td>
<td>6</td>
<td>54</td>
<td>18 s</td>
</tr>
<tr>
<td></td>
<td>7 $\times$ 7 (Sym.)</td>
<td>127</td>
<td>22</td>
<td>$3.38 \cdot 10^{12}$</td>
<td>2 618</td>
<td>19 550</td>
<td>184 d</td>
</tr>
</tbody>
</table>

$G = \{ a_0 \cdot b_0, \quad a_1 \cdot b_0, \quad (a_0 + a_1) \cdot b_0, \quad a_2 \cdot b_0, \quad (a_0 + a_2) \cdot b_0, \ldots \}$

$\mathbb{F}_3$
Some results

- Multiplication of $n \times m$ term binary polynomials

<table>
<thead>
<tr>
<th>Ring</th>
<th>$n \times m$</th>
<th>$#G$</th>
<th>$k$</th>
<th>$#$ of tests</th>
<th>$#$ of solutions</th>
<th>$#$ of formulae</th>
<th>Computation time (1 core)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbb{F}_2[X]$</td>
<td>$2 \times 2$</td>
<td>9</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>$3 \times 3$</td>
<td>49</td>
<td>6</td>
<td>9</td>
<td>3</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>$4 \times 4$</td>
<td>225</td>
<td>9</td>
<td>$6.60 \cdot 10^3$</td>
<td>4</td>
<td>4</td>
<td>30 ms</td>
</tr>
<tr>
<td></td>
<td>$5 \times 5$</td>
<td>961</td>
<td>13</td>
<td>$9.65 \cdot 10^9$</td>
<td>27</td>
<td>27</td>
<td>2 d 15 h</td>
</tr>
<tr>
<td></td>
<td>$6 \times 6$</td>
<td>3969</td>
<td>14</td>
<td>$4.37 \cdot 10^9$</td>
<td>—</td>
<td>—</td>
<td>7 d</td>
</tr>
<tr>
<td></td>
<td>$6 \times 6$ (Sym.)</td>
<td>63</td>
<td>17</td>
<td>$8.08 \cdot 10^6$</td>
<td>6</td>
<td>54</td>
<td>18 s</td>
</tr>
<tr>
<td></td>
<td>$7 \times 7$ (Sym.)</td>
<td>127</td>
<td>22</td>
<td>$3.38 \cdot 10^{12}$</td>
<td>2 618</td>
<td>19 550</td>
<td>184 d</td>
</tr>
</tbody>
</table>

$G = \{a_0 \cdot b_0, a_1 \cdot b_0, (a_0 + a_1) \cdot b_0, a_2 \cdot b_0, (a_0 + a_2) \cdot b_0, \ldots \}$

Optimal formulae for sparse multiplication useful in pairing computation

• in the genus-2 pairing, from 11 to 9 subproducts

Optimal multiplication for the extensions $\mathbb{F}_{3^m}$

• 11 subproducts instead of 12 previously

• yields a 5% improvement for the pairing on $E_{3^m}$
Some results

- Multiplication of $n \times m$ term binary polynomials

<table>
<thead>
<tr>
<th>Ring</th>
<th>$n \times m$</th>
<th>$# G$</th>
<th>$k$</th>
<th>$#$ of tests</th>
<th>$#$ of solutions</th>
<th>$#$ of formulae</th>
<th>Computation time (1 core)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbb{F}_2[X]$</td>
<td>2 × 2</td>
<td>9</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>3 × 3</td>
<td>49</td>
<td>6</td>
<td>9</td>
<td>3</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>4 × 4</td>
<td>225</td>
<td>9</td>
<td>6.60 × 10^3</td>
<td>4</td>
<td>4</td>
<td>30 ms</td>
</tr>
<tr>
<td></td>
<td>5 × 5</td>
<td>961</td>
<td>13</td>
<td>9.65 × 10^9</td>
<td>27</td>
<td>27</td>
<td>2 d 15 h</td>
</tr>
<tr>
<td></td>
<td>6 × 6</td>
<td>3969</td>
<td>14</td>
<td>4.37 × 10^9</td>
<td>—</td>
<td>—</td>
<td>7 d</td>
</tr>
<tr>
<td></td>
<td>6 × 6 (Sym.)</td>
<td>63</td>
<td>17</td>
<td>8.08 × 10^6</td>
<td>6</td>
<td>54</td>
<td>18 s</td>
</tr>
<tr>
<td></td>
<td>7 × 7 (Sym.)</td>
<td>127</td>
<td>22</td>
<td>3.38 × 10^{12}</td>
<td>2618</td>
<td>19550</td>
<td>184 d</td>
</tr>
</tbody>
</table>

- Optimal formulae for sparse multiplication useful in pairing computation
  - in the genus-2 pairing, from 11 to 9 subproducts
Some results

Multiplication of $n \times m$ term binary polynomials

<table>
<thead>
<tr>
<th>Ring</th>
<th>$n \times m$</th>
<th>$#G$</th>
<th>$k$</th>
<th># of tests</th>
<th># of solutions</th>
<th># of formulae</th>
<th>Computation time (1 core)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbb{F}_2[X]$</td>
<td>$2 \times 2$</td>
<td>9</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>$3 \times 3$</td>
<td>49</td>
<td>6</td>
<td>9</td>
<td>3</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>$4 \times 4$</td>
<td>225</td>
<td>9</td>
<td>$6.60 \cdot 10^3$</td>
<td>4</td>
<td>4</td>
<td>30 ms</td>
</tr>
<tr>
<td></td>
<td>$5 \times 5$</td>
<td>961</td>
<td>13</td>
<td>$9.65 \cdot 10^9$</td>
<td>27</td>
<td>27</td>
<td>2 d 15 h</td>
</tr>
<tr>
<td></td>
<td>$6 \times 6$</td>
<td>3969</td>
<td>14</td>
<td>$4.37 \cdot 10^9$</td>
<td>—</td>
<td>—</td>
<td>7 d</td>
</tr>
<tr>
<td></td>
<td>(Sym.) $6 \times 6$</td>
<td>63</td>
<td>17</td>
<td>$8.08 \cdot 10^6$</td>
<td>6</td>
<td>54</td>
<td>18 s</td>
</tr>
<tr>
<td></td>
<td>(Sym.) $7 \times 7$</td>
<td>127</td>
<td>22</td>
<td>$3.38 \cdot 10^{12}$</td>
<td>2 618</td>
<td>19 550</td>
<td>184 d</td>
</tr>
</tbody>
</table>

Optimal formulae for sparse multiplication useful in pairing computation
- in the genus-2 pairing, from 11 to 9 subproducts

Optimal multiplication for the extensions $\mathbb{F}_{3^m}$
- 11 subproducts instead of 12 previously
- yields a 5% improvement for the pairing on $E_3$
Outline of the talk

- Compact design through composite extension fields
- Pairing on genus-2 hyperelliptic curves
- Searching for efficient multiplication algorithms
- Conclusion and Perspectives
Conclusion

- Hardware implementations of pairing
- An algorithm to search for multiplication formulae
Conclusion

- Hardware implementations of pairing
- An algorithm to search for multiplication formulae
- Unified framework for constructing pairing algorithms
  - lot of literature on pairing algorithms
  - generally concepts and results only for specific cases
  - covers both elliptic and hyperelliptic cases
  - covers the different variants of the Tate pairing:
    - Ate, Eta, Eta T, optimal Ate, ...
Conclusion

- Hardware implementations of pairing
- An algorithm to search for multiplication formulae
- Unified framework for constructing pairing algorithms
  - lot of literature on pairing algorithms
  - generally concepts and results only for specific cases
  - covers both elliptic and hyperelliptic cases
  - covers the different variants of the Tate pairing:
    - Ate, Eta, Eta T, optimal Ate, ...
- General method for cryptographic implementations
  - study mathematical structures
  - fix parameters thanks to cryptanalysis
  - algorithmic optimizations
  - choose the right arithmetic representation
  - implement different hardware accelerators
Perspectives

- **Lower-level architecture**
  - FPGA is a good **prototyping platform**
  - but with **limited uses** in real-life devices
  - develop skills in **ASIC** designs
  - **power consumption** awareness

- **Integrate side-channel counter-measures**
  - **side-channel attacks** are very effective threats
  - **embedded systems** need to be protected

- **Use this method on different cryptographic primitives**
  - **scalar multiplication** on hyperelliptic curves
  - **lattice**-based cryptography