Short Memory Method on Koblitz Curves

Katsuyuki Okeya
Tsuyoshi Takagi
Camille Vuillaume
Outline

Elliptic curves
- Interest of ECC
- Binary Method
- NAF Method

Koblitz curves
- Binary Curves
- $\tau$ Expansions
- Binary vs. Koblitz

Short Memory
- Normal vs. Polynomial Basis
- Short Memory on Normal Basis
- Short Memory on Polynomial Basis
Outline

Elliptic curves

- Interest of ECC
- Binary Method
- NAF Method

Koblitz curves

Short Memory
RSA vs. Elliptic Curves

Speed
ECC are 30 times faster than RSA...

Memory
…and require 6.5 less memory
Binary Method

Operations:

Scalar: \( d = 105 = (1 0 1 0 1 0) \)

Input: \( P \)

Output: \( 105P \)

On average: \((n-1)D + n/2A\)
NAF<sub>w</sub> Method

Scalar: \( d=105=(1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1)_2 \)

NAF<sub>w</sub> recoding, \( w=3 \)

\[
\begin{array}{cccccccc}
1 & 0 & 0 & 0 & -3 & 0 & 0 & 1 \\
\end{array}
\]

Input:

Output: 105P

Operations:

On average: \( nD + \frac{n}{(w+1)}A + (2^{w-2}-1)A \)
Comparison – NAFw Method

Optimal width $w=5$

How to get rid of point doublings?

Maximal speed-up: less than 20% 😞
Binary Curves

- **Koblitz Curves**
  - $y^2 + xy = x^3 + ax + 1, \ a = \{0, 1\}$

- **Binary Curves**
  - $y^2 = x^3 + ax^2 + b, \ a, b \in F_{2^m}$

- Coprocessor-less architecture
  - Lower production cost, cheaper design 😊

- Can use AES acceleration hardware
  - Re-use existing design 😊

- Slow without AES hardware or fast processor
  - With coprocessor, prime curves are faster 😞

- Well-suited for mobile phone CPU (32-bit RISC)
  - Faster than general binary curves 😊😊
Binary and $\tau$-Expansions

**Binary curves**

$P = (x,y) \rightarrow 2P = (x',y')$

$\begin{array}{c}
P \\
2P \\
4P \\
8P \\
P
\end{array}$

$\begin{array}{c}
\tau \ P \\
\tau \ P + P \\
\tau \ P + 2P \\
\tau \ P + 3P \\
\tau \ P + 4P \\
\tau \ P + 5P \\
\tau \ P + 6P \\
\tau \ P + 7P \\
\tau \ P + 8P \\
\tau \ P + 9P \\
\end{array}$

$\begin{array}{c}
P \\
\tau \ P \\
\tau \ P + P \\
\tau \ P + 3P \\
\tau \ P + 5P \\
\tau \ P + 7P \\
\tau \ P + 9P \\
\end{array}$

$d = 9 = (1001_2)$

$\tau \text{-and-add: no point doubling}$

**Koblitzz curves**

$P = (x,y) \rightarrow 2P = (x',y')$

$\begin{array}{c}
P \\
\tau \ P \\
\tau \ P + \ P \\
\tau \ P + 2\ P \\
\tau \ P + 3\ P \\
\tau \ P + 4\ P \\
\tau \ P + 5\ P \\
\tau \ P + 6\ P \\
\end{array}$

$\begin{array}{c}
\tau \ P \\
\tau \ P + \ P \\
\tau \ P + 3\ P \\
\tau \ P + 5\ P \\
\tau \ P + 7\ P \\
\tau \ P + 9\ P \\
\end{array}$

$\begin{array}{c}
P \\
\tau \ P \\
\tau \ P + P \\
\tau \ P + \tau \ P \\
\tau \ P + 3P \\
\tau \ P + 5P \\
\tau \ P + 7P \\
\tau \ P + 9P \\
\end{array}$

$\begin{array}{c}
\tau = (1 + \sqrt{-1})/7 \\
\tau^2 = -1 \\
1*\tau^0 \\
1*\tau^1 \\
1*\tau^2 \\
1*\tau^3 \\
1*\tau^4 \\
1*\tau^5 \\
1*\tau^6 \\
\end{array}$

$\begin{array}{c}
d = 9 = (11111001_\tau) \\
1*\tau^0 \\
1*\tau^1 \\
1*\tau^2 \\
1*\tau^3 \\
1*\tau^4 \\
1*\tau^5 \\
1*\tau^6 \\
1*\tau^7 \\
\end{array}$

10x faster

$2P = -\tau^2 \ P + \mu \ P$

$\tau = (1 + \sqrt{-1})/7$
Binary vs. Koblitz Curves

163-bit Scalar Multiplication

- **Binary curves**
  - w=5 optimal...
  - but requires a lot of memory

- **Koblitz curves**
Outline

Short Memory Method on Koblitz Curves

Elliptic curves

Koblitz curves

Short Memory

Normal vs. Polynomial Basis

Short Memory on Normal Basis

Short Memory with Mixed Bases
## Normal vs. Polynomial Basis

<table>
<thead>
<tr>
<th>Polynomial Basis</th>
<th>Normal Basis</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p = p_{m-1}X^{m-1} + \ldots + p_1X + p_0$, where $p_i \in {0, 1}$</td>
<td>$b = b_0\beta_0 + b_1\beta_1 + \ldots + b_{m-1}\beta_{m-1}$, where $\beta_i^2 = \beta_{i+1}$ and $\beta_{m-1}^2 = \beta_0$</td>
</tr>
</tbody>
</table>

- Fast reduction with trinomials or pentanomials
- Fast multiplications
- No reduction
- Software
- Hardware

### Fast squares

- $b = b_0\beta_0 + b_1\beta_1 + \ldots + b_{m-2}\beta_{m-2} + b_{m-1}\beta_{m-1}$
- $b^2 = b_0\beta_0^2 + b_1\beta_1^2 + \ldots + b_{m-2}\beta_{m-2}^2 + b_{m-1}\beta_{m-1}^2$
- $= b_{m-1}\beta_0 + b_0\beta_1 + b_1\beta_2 + \ldots + b_{m-2}\beta_{m-1}$

Interesting for Koblitz curves ($\tau$)

**Mixed approach?**
Short Memory - Normal Basis

Standard method

\[
\begin{array}{cccccccc}
-1 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 1 \\
\end{array}
\]

\[
P
\]

\[
3P
\]

\[
\tau
\]

\[
\tau
\]

\[
\tau
\]

\[
\tau
\]

\[
\tau
\]

\[
\tau
\]

\[
\tau
\]

Short memory

\[
\begin{array}{cccccccc}
-1 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 1 \\
\end{array}
\]

\[
\tau^8P
\]

\[
\tau^8
\]

\[
\tau^8P+P
\]

\[
P
\]

\[
3\tau^3P
\]

\[
\tau^3
\]

\[
9P
\]

\[
-\tau^8P+P
\]

\[
3P
\]
# Sequential Precomputations

Short Memory Method on Koblitz Curves

<table>
<thead>
<tr>
<th>$u$</th>
<th>$\alpha_u = u \mod \tau^5$</th>
<th>Binary representation of $\alpha_u$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>$1$</td>
</tr>
<tr>
<td>3</td>
<td>$\tau - 3$</td>
<td>$\tau - 1$</td>
</tr>
<tr>
<td>5</td>
<td>$\tau - 1$</td>
<td>$\tau + 1$</td>
</tr>
<tr>
<td>7</td>
<td>$\tau + 1$</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>$2\tau - 3$</td>
<td>$-\tau - 1$</td>
</tr>
<tr>
<td>11</td>
<td>$2\tau - 1$</td>
<td>$2\tau + 1$</td>
</tr>
<tr>
<td>13</td>
<td>$2\tau + 1$</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>$-3\tau + 1$</td>
<td></td>
</tr>
</tbody>
</table>

### Precomputations

- $\alpha_4 P = P$
- $\alpha_5 P = tP - P$
- $\alpha_6 P = \tau P + P$
- $\alpha_{13} P = -2 \alpha_5 P + P$
- $\alpha_{11} P = -\tau^3 P + \alpha_3 P$
- $\alpha_7 P = \tau P + P$
- $\alpha_{15} P = \alpha_{11} P - \tau P$
- $\alpha_9 P = -\tau^4 P - \alpha_7 P$
Performance, Hardware

Short Memory Method on Koblitz Curves

163-bit binary curve
163-bit Koblitz curve
163-bit normal basis

Cost (elliptic operations)

Memory (bits)

Performance, Hardware

163-bit Koblitz curve

163-bit binary curve

Cost (elliptic operations)

Memory (bits)

Performance, Hardware

Short Memory Method on Koblitz Curves
Short Memory - Mixed Bases

- Short Memory Method on Koblitz Curves

\[
\begin{array}{cccccccc}
-1 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 1 \\
\end{array}
\]

\[
\tau^8P \quad \tau^8P \\
\tau^8P \quad \tau^8P \\
\tau^3P \quad \tau^3P \\
\tau^3P \quad \tau^3P \\
\end{array}
\]

\[
\begin{array}{cccccccc}
P & P & P & P & P & P & P & P \\
\end{array}
\]

\[
\begin{array}{cccccccc}
\text{polynomial basis} & \text{normal basis} \\
\end{array}
\]

\[
u=1, u=3
\]
Change of Basis

Original representation:

1
1
0
1
1
0
1
1

New representation:

Change of basis matrix:

0 1 0 0 0 1
⊕
1 1 0 1 0 0
⊕
1 1 0 0 1 1
⊕
0 0 1 1 0 0
⊕
0 1 0 1 0 1
⊕
1 1 1 1 1 0 0
⊕

Cyclic shift not explicitly computed
Performance, Software

Short Memory Method on Koblitz Curves

163-bit binary curve

163-bit Koblitz curve

163-bit polynomial basis

Memory (bytes)

Cost (multiplications)

0 50 100 150 200 250 300 350

0 100 200 300 400 500 600 700 800 900 1000 1100 1200 1300 1400

0 42 84 126 168 210 252 294

0 200 400 600 800 1000 1200 1400

Memory (bytes) -- Cost (M)

☺☺
Extensions, open problems

- Side channel & fault attacks
- Change of basis
- Other curves
Recap

"Le beurre et l’argent du beurre"
Questions & Comments