$GF(2^m)$ Finite-Field Multipliers with Reduced Activity Variations

Danuta Pamuła, Arnaud Tisserand

Silesian University of Technology
IRISA / Université de Rennes 1

18.07.2012
1. Problem introduction - security of Elliptic Curve Cryptography systems
### Elliptic Curve Cryptography protocols

#### Key pair generation

<table>
<thead>
<tr>
<th>IN: ( (p, E, P, n) )</th>
<th>OUT: ( (Q, d) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Select ( d \in \mathbb{R} ) ([1, n-1])</td>
<td>2. Compute ( Q = dP )</td>
</tr>
<tr>
<td>3. Return ( (Q, d) )</td>
<td></td>
</tr>
</tbody>
</table>

#### ElGamal encryption

<table>
<thead>
<tr>
<th>IN: ( (p, E, P, n), Q, m )</th>
<th>OUT: ( (Q, d) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Represent ( m ) as a point ( M \in E(\mathbb{F}_p) )</td>
<td></td>
</tr>
<tr>
<td>2. Select ( k \in \mathbb{R} ) ([1, n-1])</td>
<td></td>
</tr>
<tr>
<td>3. Compute ( C_1 = kP )</td>
<td></td>
</tr>
<tr>
<td>4. Compute ( C_2 = M + kQ )</td>
<td></td>
</tr>
<tr>
<td>5. Return ( (C_1, C_2) )</td>
<td></td>
</tr>
</tbody>
</table>

#### ElGamal decryption

<table>
<thead>
<tr>
<th>IN: ( (p, E, P, n), d, (C_1, C_2) )</th>
<th>OUT: ( m )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Compute ( M = C_2 - dC_1 ) and extract ( m ) from ( M )</td>
<td></td>
</tr>
<tr>
<td>2. Return ( m )</td>
<td></td>
</tr>
</tbody>
</table>

\((p, E, P, n)\) - elliptic curve parameters, \( Q, d \) - key pair: private and public, \( m \) - message, plaintext; \( (C_1, C_2) \) - ciphertext
Typical ECC Computations

- **E**: \( y^2 = x^3 + 4x + 20 \) over \( \text{GF}(1009) \)
- Points on \( E \): \( P, Q = (x, y) \) or \( (x, y, z) \)
- Coordinates: \( x, y, z \in \text{GF}(\cdot) \)
- \( \text{GF}(p), \text{GF}(2^m), t : 160-600 \) bits
- \( k = (k_{t-1} k_{t-2} \ldots k_1 k_0)_2 \in \mathbb{N} \)

**Scalar multiplication operation**

- for \( i \) from 0 to \( t - 1 \) do
  - if \( k_i = 1 \) then \( Q = \text{ADD}(P, Q) \)
  - \( P = \text{DBL}(P) \)

**Point addition/doubling operations**

- Sequence of finite field operations
  - \( \text{DBL}: v_1 = z_1^2, v_2 = x_1 - v_1, \ldots \)
  - \( \text{ADD}: v_1 = z_1^2, v_2 = z_1 \times v_1, \ldots \)

**GF(\(p\)) or GF(\(2^m\)) operations**

- Operation modulo: large prime (GF(\(p\))) or irreducible polynomial (GF(\(2^m\)))
Basic Power Analysis Attack on ECC

Scalar multiplication operation

for $i$ from 0 to $t-1$ do
  if $k_i = 1$ then $Q = \text{ADD}(P, Q)$
  $P = \text{DBL}(P)$
Exemplary SCA countermeasures

1. Key recoding:
   - redundant number system
   - double base representation DBNS
   \[ k = \sum_{i=0}^{m-1} k_i 2^{a_j} 3^{b_j} \]

2. Algorithm modifications
2. **Activity variations in hardware arithmetic operators**

- Information leakage evaluation method
- Activity variations of efficient hardware $GF(2^m)$ multipliers
- Proposed architectural and algorithmic modifications
\[ P_{DD}(t) = i_{DD}(t) \times V_{DD} \]

- **static power**
  - does not depend on circuit activity
- **dynamic power**
  - depends on circuit activity
    - charging and discharging short circuit currents
    - load/parasitic capacitances

- instantaneous power
- instantaneous current
- power supply

Danuta Pamuła, Arnaud Tisserand

\( GF(2^m) \) Multipliers with Reduced Activity Variations
Dynamic power

useful activity

parasitic activity

input → clk → logic → reg. → x(t)

\[
\begin{align*}
x(t) & \quad 00110101 \\
x(t + 1) & \quad 01101011
\end{align*}
\]

5 useful transitions

parasitic transition on z

Danuta Pamuła, Arnaud Tisserand

GF(2^m) Multipliers with Reduced Activity Variations
Dynamic power

**useful activity**

**parasitic activity**

\[ x(t) \rightarrow \text{logic} \rightarrow \text{reg.} \rightarrow x(t) \]

\[
\begin{align*}
x(t) & = 00110101 \\
x(t+1) & = 01101011
\end{align*}
\]

5 useful transitions

parasitic transition on z

Danuta Pamuła, Arnaud Tisserand

\( GF(2^m) \) Multipliers with Reduced Activity Variations
→ circuit activity monitoring method: → → add activity monitors

**Activity monitor**  $\Rightarrow$ counts the number of useful transitions in the circuit

$s(t)$ monitored signal

transition $s(t+1)$ vs. $s(t)$

total number of transitions

the activity counter outputs are analysed using ChipScope tool.
Efficient hardware $GF(2^m)$ multipliers

Proposed $GF(2^m)$ multipliers are based on:

- Classic algorithm combined with optimised divide-and-conquer method (Karatsuba-Ofman method)
- Mastrovito matrix method
- Montgomery algorithm

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$w$-o activity monitors</th>
<th>with activity monitors</th>
<th>AT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>size</td>
<td>$f$</td>
<td>clock</td>
</tr>
<tr>
<td>Classic</td>
<td>3638</td>
<td>302</td>
<td>234</td>
</tr>
<tr>
<td>Montgomery</td>
<td>2178</td>
<td>323</td>
<td>270</td>
</tr>
<tr>
<td>Mastrovito</td>
<td>3760</td>
<td>297</td>
<td>75</td>
</tr>
</tbody>
</table>

multipliers were implemented in Xilinx Virtex-6 device.
Classic algorithm

---

**Problem introduction - security of ECC systems**

Activity variations in Hardware Arithmetic Operators

Results and conclusions

---

Information leakage evaluation method

Activity variations of efficient $GF(2^m)$ operators

Proposed of architectural and arithmetic modifications

---

**Danuta Pamuła, Arnaud Tisserand**

$GF(2^m)$ Multipliers with Reduced Activity Variations
Montgomery algorithm

Complete multiplication process (with conversion from Montgomery representation)
Mastrovito algorithm

Danuta Pamuła, Arnaud Tisserand

GF(2^m) Multipliers with Reduced Activity Variations
Conclusions

observable circuit initialisation (operation start)

observable reduction (operation end)

observable peak and drop of activity at the beginning and at the end of multiplication
Modification goals

- **reduce activity variations** - *uniformise* activity trace shapes
- **reduce activity peaks and drops** occurring during circuit activity

**activity trace randomisation** - activity trace for each multiplication is different
Classic algorithm - modifications

**Problems**

activity peak at the beginning of operation - initialisation/reset of circuit’s registers

**Proposed modifications**

- spreading reinitialisation over several clock cycles
- filling the registers with random and variable values (instead of zeroes or constants)
- removal of auxiliary and temporary registers - algorithm optimisation, reduction optimisation

*multiple steps of the algorithm were replaced by chain of XOR operation; power consumption reduction - smaller number of simultaneously switching registers*
Activity variations in Hardware Arithmetic Operators

Information leakage evaluation method

Proposed of architectural and arithmetic modifications

GF($2^m$) Multipliers with Reduced Activity Variations

Danuta Pamuła, Arnaud Tisserand
 Montgomery algorithm

Problems

activity drops due to reduction - concerning the Montgomery method idea the problem can be neglected

Montgomery algorithm

Danuta Pamuła, Arnaud Tisserand

GF(2^m) Multipliers with Reduced Activity Variations
Mastrovito algorithm

**Problems**

Specific “step-wave” shape

**Modification goals**

- activity variations uniformisation
- activity variations randomisation

*masking the operation, hindering operation identification*

Mastrovito algorithm

\[ c = ab \mod f = Mb \]
### Specificity of proposed architecture based on Mastrovito algorithm

<table>
<thead>
<tr>
<th>M</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M1</td>
<td>M1</td>
<td>M1</td>
<td>M1</td>
<td>M1</td>
<td>Mc</td>
</tr>
<tr>
<td>1</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M1</td>
<td>M1</td>
<td>M1</td>
<td>M1</td>
<td>Mc</td>
</tr>
<tr>
<td>2</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M1</td>
<td>M1</td>
<td>M1</td>
<td>M1</td>
<td>Mc</td>
</tr>
<tr>
<td>3</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M1</td>
</tr>
<tr>
<td>4</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
</tr>
<tr>
<td>5</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
</tr>
<tr>
<td>6</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
</tr>
<tr>
<td>7</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
</tr>
<tr>
<td>8</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
</tr>
<tr>
<td>9</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
</tr>
<tr>
<td>10</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
<td>M2</td>
</tr>
<tr>
<td>11</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
</tr>
<tr>
<td>12</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
</tr>
<tr>
<td>13</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
<td>M0</td>
</tr>
<tr>
<td>14</td>
<td>Mr</td>
<td>Mr</td>
<td>Mr</td>
<td>Mr</td>
<td>Mr</td>
<td>Mr</td>
<td>Mr</td>
<td>Mr</td>
<td>Mr</td>
<td>Mr</td>
<td>Mr</td>
<td>Mr</td>
<td>Mr</td>
<td>Mr</td>
<td>Mr</td>
</tr>
</tbody>
</table>
Activity variations

Uniformisation
- Utilisation of same number of registers in one clock cycle.
- Without changing computation time.

Randomisation
- Introducing additional controls for the sub-multipliers
- Randomisation of starting moments of each sub-multiplier used - random start sequence generator based on LFSR

LFSR is initialised with values depending on certain bits of one of the operands. The computation times becomes random.
Problem introduction - security of ECC systems
Activity variations in Hardware Arithmetic Operators
Results and conclusions

Information leakage evaluation method
Activity variations of efficient $GF(2^m)$ operators
Proposed of architectural and arithmetic modifications

Danuta Pamuła, Arnaud Tisserand

$GF(2^m)$ Multipliers with Reduced Activity Variations
3. Results and conclusions

- Protected operators - summary
- Further work
Evaluation of activity variation reduction using signal processing tools

measured activity traces (time domain) $\Rightarrow$ frequency domain using Fast Fourier Transform (FFT),

**spectral flatness measure (SFM) - numerical measure**

$$\text{SFM} = \sqrt{\prod_{i=1}^{n} p(i)} \frac{1}{\frac{1}{n} \sum_{i=1}^{n} p(i)} \in [0, 1]$$
### Implementation results of protected multipliers

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>balanced</th>
<th>area</th>
<th>speed</th>
<th># clock cycles</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>LUT</td>
<td>MHz</td>
<td>LUT</td>
<td>MHz</td>
</tr>
<tr>
<td>Classical</td>
<td>2868</td>
<td>270</td>
<td>2778</td>
<td>228</td>
</tr>
<tr>
<td>× α factor</td>
<td>0.79</td>
<td>0.89</td>
<td>0.76</td>
<td>0.75</td>
</tr>
<tr>
<td>Montgomery</td>
<td>2099</td>
<td>323</td>
<td>2093</td>
<td>338</td>
</tr>
<tr>
<td>× α factor</td>
<td>0.96</td>
<td>1.00</td>
<td>0.96</td>
<td>1.05</td>
</tr>
<tr>
<td>Mastrovito v1</td>
<td>3463</td>
<td>414</td>
<td>3439</td>
<td>343</td>
</tr>
<tr>
<td>× α factor</td>
<td>0.99</td>
<td>1.50</td>
<td>0.98</td>
<td>1.24</td>
</tr>
<tr>
<td>Mastrovito v2</td>
<td>3700</td>
<td>306</td>
<td>3667</td>
<td>253</td>
</tr>
<tr>
<td>× α factor</td>
<td>1.06</td>
<td>1.11</td>
<td>1.05</td>
<td>0.92</td>
</tr>
<tr>
<td>Mastrovito v3</td>
<td>3903</td>
<td>319</td>
<td>3837</td>
<td>250</td>
</tr>
<tr>
<td>× α factor</td>
<td>1.12</td>
<td>1.16</td>
<td>1.10</td>
<td>0.91</td>
</tr>
</tbody>
</table>

modified $= \alpha \times$ original
Further work

- analysis of activity variations of other $GF(2^m)$ arithmetic operators - division, inversion
- analysis of activity variations of chain of $GF(2^m)$ arithmetic operations needed to perform $2P$ and $P + Q$ operations
Thank You :}
Thank You :)
Problem introduction - security of ECC systems
Activity variations in Hardware Arithmetic Operators
Results and conclusions
Protected operators - summary
Further work

Danuta Pamuła, Arnaud Tisserand

$GF(2^m)$ Multipliers with Reduced Activity Variations
Problem introduction - security of ECC systems
Activity variations in Hardware Arithmetic Operators
Results and conclusions
Protected operators - summary
Further work

Danuta Pamuła, Arnaud Tisserand

$\text{GF}(2^m)$ Multipliers with Reduced Activity Variations