CHES 2009 — Lausanne, Switzerland

Hardware accelerator for the Tate pairing in characteristic three based on Karatsuba multipliers

Nicolas Estibals
CACAO, INRIA, Nancy, France

Joint work with:

Jean-Luc Beuchat	LCIS, University of Tsukuba, Japan
Jérémie Detrey	CACAO, INRIA, Nancy, France
Eiji Okamoto	LCIS, University of Tsukuba, Japan
Francisco Rodríguez-Henríquez	CINVESTAV, IPN, Mexico City, Mexico

Outline of the talk

Pairing-based cryptography

- Origin of pairings in cryptography
  - Menezes–Okamoto–Vanstone (1993) and Frey–Rück (1994)
  - attack against some elliptic curves
- Constructive properties
  - short signature
  - identity-based cryptography
  - ...
- Standardization in progress

Context

- Reduced Tate pairing
- Pairing on supersingular curves
  - \+ easier arithmetic on the curve
  - − lower security
- Characteristic 3
  - higher embedding degree
  - higher security

Context

- Reduced Tate pairing
- Pairing on supersingular curves
  + easier arithmetic on the curve
  – lower security
- Characteristic 3
  - higher embedding degree
  - higher security
- Need for dedicated hardware coprocessor
  - area optimized (RFID, embedded systems, ...)
  - speed optimized (bank servers, ...)

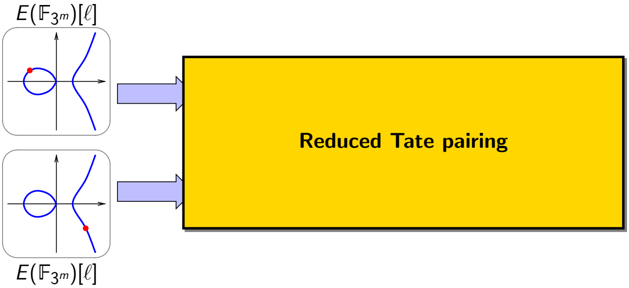
Context

- Reduced Tate pairing
- Pairing on supersingular curves
  + easier arithmetic on the curve
  – lower security
- Characteristic 3
  - higher embedding degree
  - higher security
- Need for dedicated hardware coprocessor
  - area optimized (RFID, embedded systems, . . .)
  - speed optimized (bank servers, . . .)

Reduced Tate pairing

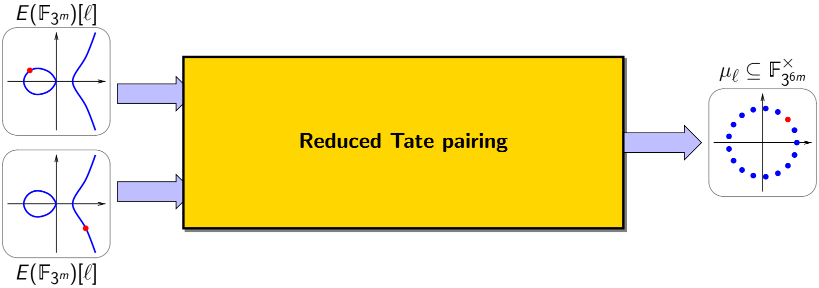
Reduced Tate pairing

Reduced Tate pairing



- Input: two points P and Q in $E(\mathbb{F}_{3^m})[\ell]$

Reduced Tate pairing

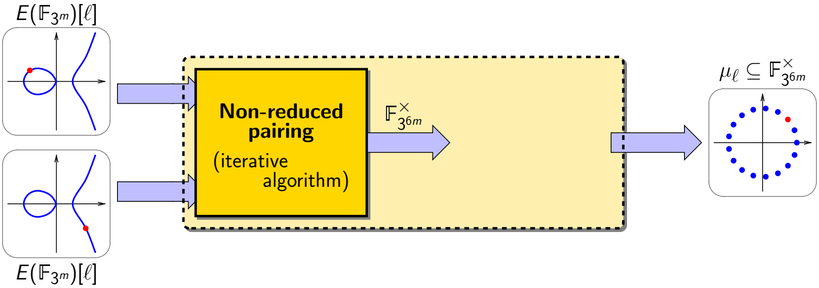


- Input: two points P and Q in $E(\mathbb{F}_{3^m})[\ell]$
- Output: an ℓ-th root of unity in the extension $\mathbb{F}^\times_{3^{6m}}$

Reduced Tate pairing

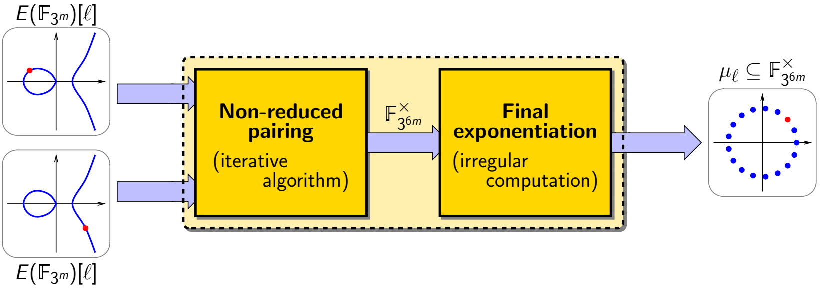
- Input: two points P and Q in $E(\mathbb{F}_{3^m})[\ell]$
- Output: an ℓ-th root of unity in the extension $\mathbb{F}_{3^{6m}}^\times$
- Two very different steps

Reduced Tate pairing



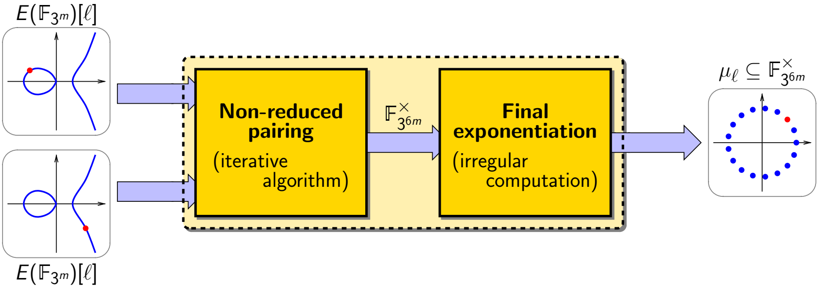
- Input: two points P and Q in $E(\mathbb{F}_{3^m})[\ell]$
- Output: an ℓ-th root of unity in the extension $\mathbb{F}_{3^{6m}}^\times$
- Two very different steps

Reduced Tate pairing



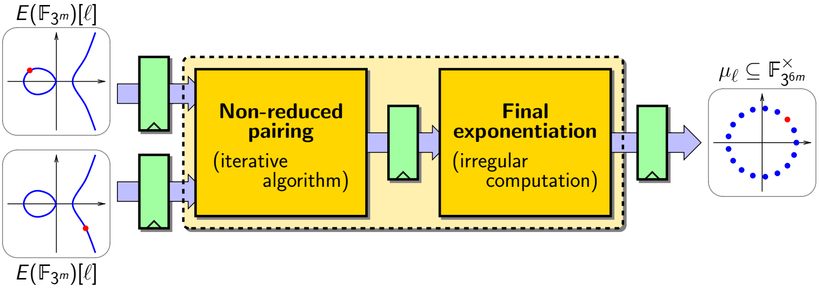
- Input: two points P and Q in $E(\mathbb{F}_{3^m})[\ell]$
- Output: an ℓ-th root of unity in the extension $\mathbb{F}_{3^{6m}}^{\times}$
- Two very different steps

Reduced Tate pairing



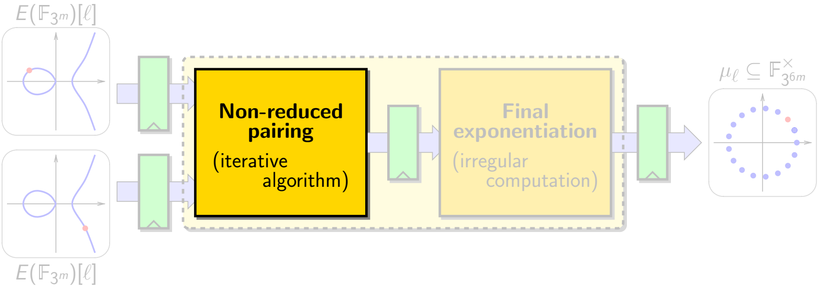
- Input: two points P and Q in $E(\mathbb{F}_{3^m})[\ell]$
- Output: an ℓ-th root of unity in the extension $\mathbb{F}^\times_{3^{6m}}$
- Two very different steps
- Idea: use two separate coprocessors

Reduced Tate pairing



- Input: two points P and Q in $E(\mathbb{F}_{3^m})[\ell]$
- Output: an ℓ-th root of unity in the extension $\mathbb{F}_{3^{6m}}^\times$
- Two very different steps
- Idea: use two separate coprocessors
  - pipeline the two computations
  - balance the latencies

Reduced Tate pairing



- Input: two points P and Q in $E(\mathbb{F}_{3^m})[\ell]$
- Output: an ℓ-th root of unity in the extension $\mathbb{F}_{3^{6m}}^\times$
- Two very different steps
- Idea: use two separate coprocessors
  - pipeline the two computations
  - balance the latencies

Computing the non-reduced pairing

- η_T pairing: shorter loop

for $i \leftarrow 0$ **to** $(m-1)/2$ **do**

end for

Computing the non-reduced pairing

- η_T pairing: shorter loop
- Based on Miller's algorithm:

for $i \leftarrow 0$ **to** $(m-1)/2$ **do**

$$x_P \leftarrow \sqrt[3]{x_P} \quad ; \quad y_P \leftarrow \sqrt[3]{y_P}$$
$$x_Q \leftarrow x_Q^3 \quad ; \quad y_Q \leftarrow y_Q^3$$

$$t \leftarrow x_P + x_Q \qquad u \leftarrow y_P y_Q$$
$$S \leftarrow -t^2 \pm u\sigma - t\rho - \rho^2$$

$$R \leftarrow R \cdot S$$

end for

Computing the non-reduced pairing

- η_T pairing: shorter loop
- Based on Miller's algorithm:
  - ① update of point coordinates

for $i \leftarrow 0$ **to** $(m-1)/2$ **do**

① $x_P \leftarrow \sqrt[3]{x_P}$; $y_P \leftarrow \sqrt[3]{y_P}$ — $2\ \sqrt[3]{\cdot}$

$x_Q \leftarrow x_Q^3$; $y_Q \leftarrow y_Q^3$ — $2\ (\cdot)^3$

$t \leftarrow x_P + x_Q \qquad u \leftarrow y_P y_Q$

$S \leftarrow -t^2 \pm u\sigma - t\rho - \rho^2$

$R \leftarrow R \cdot S$

end for

Computing the non-reduced pairing

- η_T pairing: shorter loop
- Based on Miller's algorithm:
  - ① update of point coordinates
  - ② computation of line equation

for $i \leftarrow 0$ **to** $(m-1)/2$ **do**

① $x_P \leftarrow \sqrt[3]{x_P}$; $y_P \leftarrow \sqrt[3]{y_P}$ — $2\ \sqrt[3]{\cdot}$
$x_Q \leftarrow x_Q^3$; $y_Q \leftarrow y_Q^3$ — $2\ (\cdot)^3$

② $t \leftarrow x_P + x_Q$; $u \leftarrow y_P y_Q$ — $2\times, 2+$
$S \leftarrow -t^2 \pm u\sigma - t\rho - \rho^2$

$R \leftarrow R \cdot S$

end for

Computing the non-reduced pairing

- η_T pairing: shorter loop
- Based on Miller's algorithm:
  - ① update of point coordinates
  - ② computation of line equation
  - ③ accumulation of the new factor

for $i \leftarrow 0$ **to** $(m-1)/2$ **do**

① $x_P \leftarrow \sqrt[3]{x_P}$; $y_P \leftarrow \sqrt[3]{y_P}$ — $2\ \sqrt[3]{\cdot}$
$x_Q \leftarrow x_Q^3$; $y_Q \leftarrow y_Q^3$ — $2\ (\cdot)^3$

② $t \leftarrow x_P + x_Q$; $u \leftarrow y_P y_Q$ — $2\times, 2+$
$S \leftarrow -t^2 \pm u\sigma - t\rho - \rho^2$

③ $R \leftarrow R \cdot S$ — $1\times\ (\mathbb{F}_{3^{6m}})$

end for

Computing the non-reduced pairing

- η_T pairing: shorter loop
- Based on Miller's algorithm:
  - ① update of point coordinates
  - ② computation of line equation
  - ③ accumulation of the new factor
- Multiplication is critical
- Fully parallel, pipelined multiplier over $\mathbb{F}_{3^m}$

for $i \leftarrow 0$ **to** $(m-1)/2$ **do**

①	$x_P \leftarrow \sqrt[3]{x_P}$; $y_P \leftarrow \sqrt[3]{y_P}$ $x_Q \leftarrow x_Q^3$; $y_Q \leftarrow y_Q^3$	$2\ \sqrt[3]{\cdot}$ $2\ (\cdot)^3$
②	$t \leftarrow x_P + x_Q$; $u \leftarrow y_P y_Q$ $S \leftarrow -t^2 \pm u\sigma - t\rho - \rho^2$	$2\times, 2+$
③	$R \leftarrow R \cdot S$	$1\times\ (\mathbb{F}_{3^{6m}})$

end for

Computing the non-reduced pairing

- η_T pairing: shorter loop
- Based on Miller's algorithm:
  - ① update of point coordinates
  - ② computation of line equation
  - ③ accumulation of the new factor
- Multiplication is critical
- Fully parallel, pipelined multiplier over $\mathbb{F}_{3^m}$
- Sparse multiplication over $\mathbb{F}_{3^{6m}}$

for $i \leftarrow 0$ **to** $(m-1)/2$ **do**

① $x_P \leftarrow \sqrt[3]{x_P}$; $y_P \leftarrow \sqrt[3]{y_P}$ — $2\ \sqrt[3]{\cdot}$
$x_Q \leftarrow x_Q^3$; $y_Q \leftarrow y_Q^3$ — $2\ (\cdot)^3$

② $t \leftarrow x_P + x_Q$; $u \leftarrow y_P y_Q$
$S \leftarrow -t^2 \pm u\sigma - t\rho - \rho^2$ — $2\times, 2+$

③ $R \leftarrow R \cdot S$ — $1\times\ (\mathbb{F}_{3^{6m}})$

end for

Computing the non-reduced pairing

- η_T pairing: shorter loop
- Based on Miller's algorithm:
  - ① update of point coordinates
  - ② computation of line equation
  - ③ accumulation of the new factor
- Multiplication is critical
- Fully parallel, pipelined multiplier over $\mathbb{F}_{3^m}$

for $i \leftarrow 0$ **to** $(m-1)/2$ **do**

①	$x_P \leftarrow \sqrt[3]{x_P}$; $y_P \leftarrow \sqrt[3]{y_P}$ $x_Q \leftarrow x_Q^3$; $y_Q \leftarrow y_Q^3$	$2\ \sqrt[3]{\cdot}$ $2\ (\cdot)^3$
②	$t \leftarrow x_P + x_Q$; $u \leftarrow y_P y_Q$ $S \leftarrow -t^2 \pm u\sigma - t\rho - \rho^2$	$2\times, 2+$
③	$R \leftarrow R \cdot S$	$12\times, 59+$

end for

- Sparse multiplication over $\mathbb{F}_{3^{6m}}$
  - ▸ $12\times$ and $59+$ over $\mathbb{F}_{3^m}$ (Gorla *et al.*, SAC 2007)

Computing the non-reduced pairing

- η_T pairing: shorter loop
- Based on Miller's algorithm:
  - ① update of point coordinates
  - ② computation of line equation
  - ③ accumulation of the new factor
- Multiplication is critical
- Fully parallel, pipelined multiplier over $\mathbb{F}_{3^m}$

for $i \leftarrow 0$ **to** $(m-1)/2$ **do**

① $x_P \leftarrow \sqrt[3]{x_P}$; $y_P \leftarrow \sqrt[3]{y_P}$ — $2\ \sqrt[3]{\cdot}$
$x_Q \leftarrow x_Q^3$; $y_Q \leftarrow y_Q^3$ — $2\ (\cdot)^3$

② $t \leftarrow x_P + x_Q$; $u \leftarrow y_P y_Q$ — $2\times, 2+$
$S \leftarrow -t^2 \pm u\sigma - t\rho - \rho^2$

③ $R \leftarrow R \cdot S$ — $15\times, 29+$

end for

- Sparse multiplication over $\mathbb{F}_{3^{6m}}$
  - $12\times$ and $59+$ over $\mathbb{F}_{3^m}$ (Gorla *et al.*, SAC 2007)
  - $15\times$ and $29+$ over $\mathbb{F}_{3^m}$ (Beuchat *et al.*, ARITH 18)

Computing the non-reduced pairing

- η_T pairing: shorter loop
- Based on Miller's algorithm:
  - ① update of point coordinates
  - ② computation of line equation
  - ③ accumulation of the new factor
- Multiplication is critical
- Fully parallel, pipelined multiplier over $\mathbb{F}_{3^m}$

for $i \leftarrow 0$ **to** $(m-1)/2$ **do**

① $x_P \leftarrow \sqrt[3]{x_P}$; $y_P \leftarrow \sqrt[3]{y_P}$ — $2\ \sqrt[3]{\cdot}$
$x_Q \leftarrow x_Q^3$; $y_Q \leftarrow y_Q^3$ — $2\ (\cdot)^3$

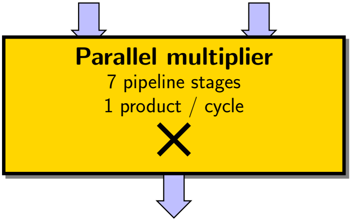
② $t \leftarrow x_P + x_Q$; $u \leftarrow y_P y_Q$ — $2\times, 2+$
$S \leftarrow -t^2 \pm u\sigma - t\rho - \rho^2$

③ $R \leftarrow R \cdot S$ — $15\times, 29+$

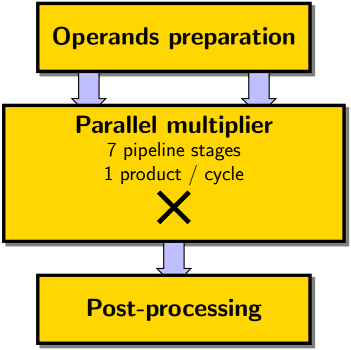
end for

- Sparse multiplication over $\mathbb{F}_{3^{6m}}$
  - $12\times$ and $59+$ over $\mathbb{F}_{3^m}$ (Gorla *et al.*, SAC 2007)
  - $15\times$ and $29+$ over $\mathbb{F}_{3^m}$ (Beuchat *et al.*, ARITH 18)
- Objective: keep the multiplier pipeline busy
  - 7-stage pipeline
  - one product per cycle
  - 17 cycles per iteration

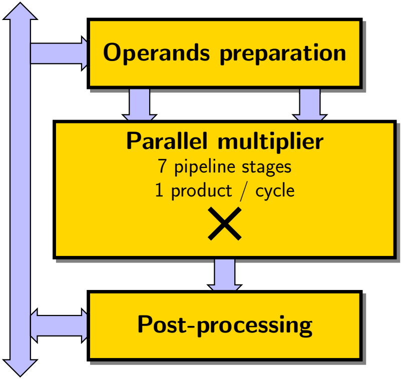
Coprocessor for the non-reduced pairing



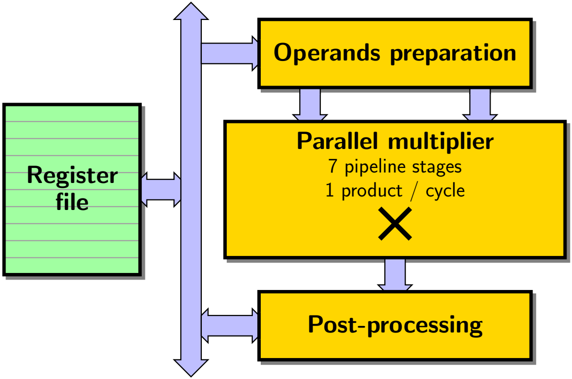
Coprocessor for the non-reduced pairing



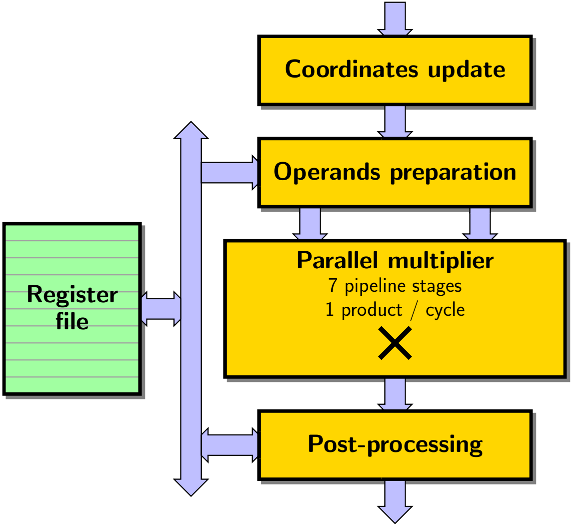
Coprocessor for the non-reduced pairing



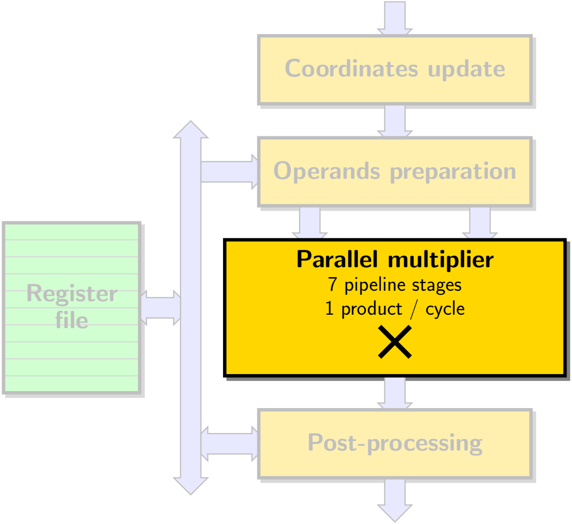
Coprocessor for the non-reduced pairing



Coprocessor for the non-reduced pairing



Coprocessor for the non-reduced pairing



Our parallel multiplier

- Polynomial basis:

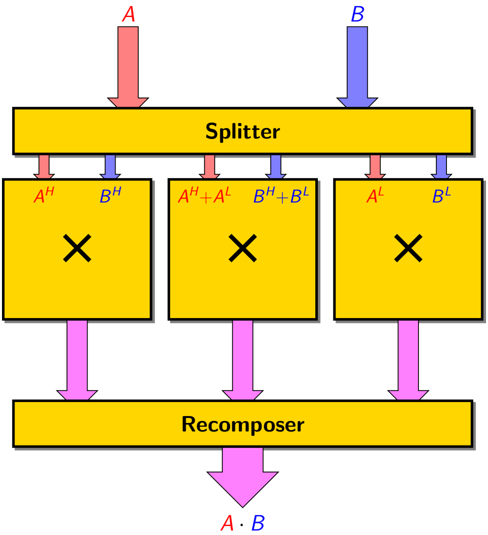
$$\mathbb{F}_{3^m} \cong \mathbb{F}_3[x]/(f(x))$$

Our parallel multiplier

- Polynomial basis:

$$\mathbb{F}_{3^m} \cong \mathbb{F}_3[x]/(f(x))$$

- Karatsuba scheme
- Fully parallel

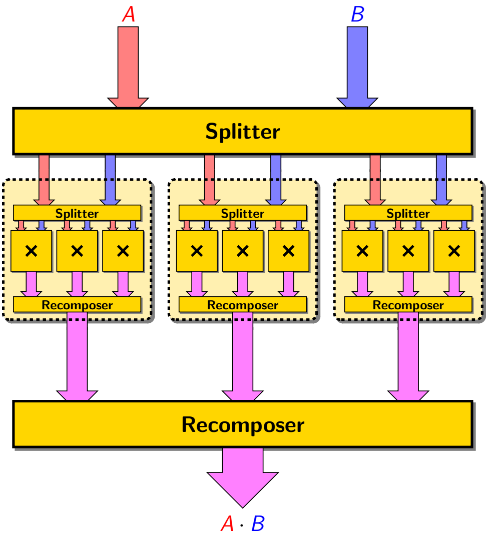


Our parallel multiplier

- Polynomial basis:

$$\mathbb{F}_{3^m} \cong \mathbb{F}_3[x]/(f(x))$$

- Karatsuba scheme
- Fully parallel
- Recursive scheme

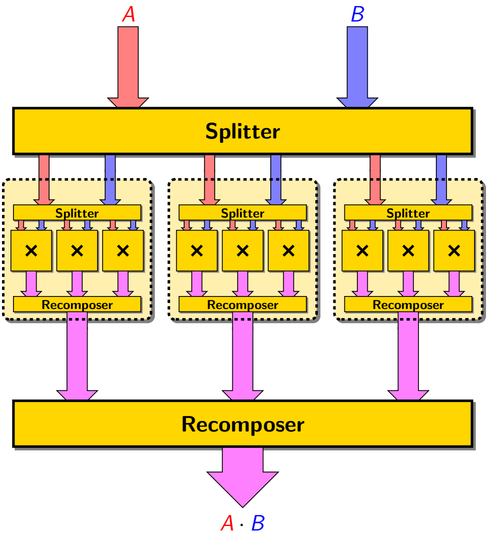


Our parallel multiplier

- Polynomial basis:

$$\mathbb{F}_{3^m} \cong \mathbb{F}_3[x]/(f(x))$$

- Karatsuba scheme
- Fully parallel
- Recursive scheme
- Some variations:
  - 3-way split
  - odd–even split
  - select the best method for each stage

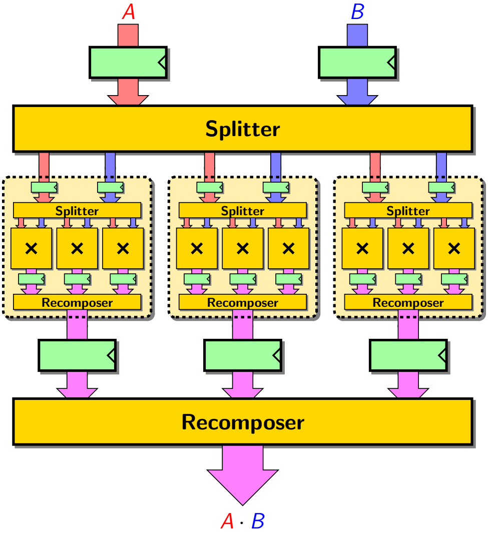


Our parallel multiplier

- Polynomial basis:

$$\mathbb{F}_{3^m} \cong \mathbb{F}_3[x]/(f(x))$$

- Karatsuba scheme
- Fully parallel
- Recursive scheme
- Some variations:
  - 3-way split
  - odd–even split
  - select the best method for each stage
- Pipelined: optional registers

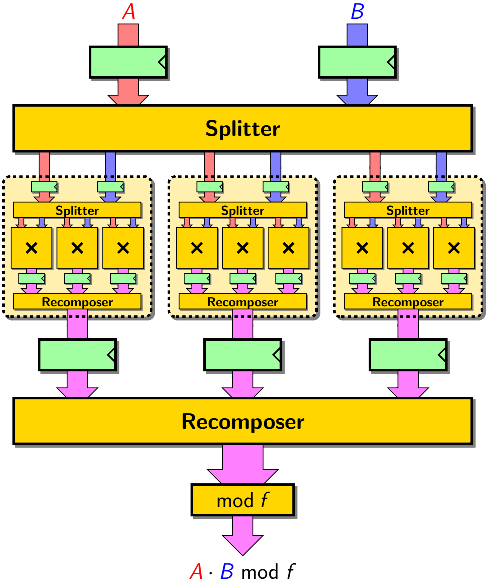


Our parallel multiplier

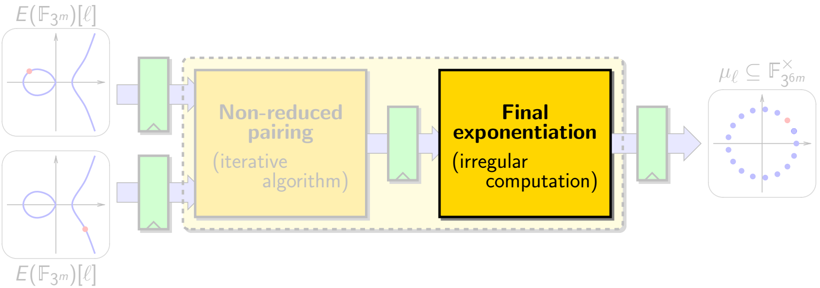
- Polynomial basis:

  $$\mathbb{F}_{3^m} \cong \mathbb{F}_3[x]/(f(x))$$

- Karatsuba scheme
- Fully parallel
- Recursive scheme
- Some variations:
  - 3-way split
  - odd–even split
  - select the best method for each stage
- Pipelined: optional registers
- Final reduction modulo f



Final exponentiation



- Design rationale:
  - as small as possible
  - at least as fast as the computation of the non-reduced pairing

Coprocessor for the final exponentiation

- Highly sequential computation
- Very heterogeneous

Coprocessor for the final exponentiation

- Highly sequential computation
- Very heterogeneous

$\Rightarrow$ general-purpose finite-field arithmetic processor

Coprocessor for the final exponentiation

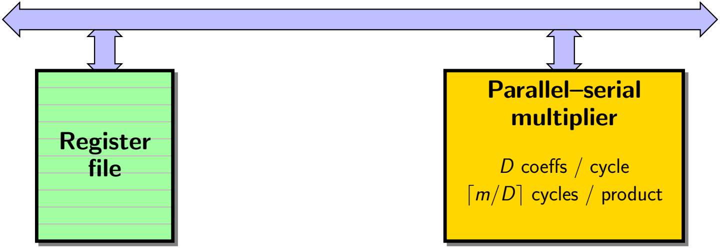
- Highly sequential computation
- Very heterogeneous

$\Rightarrow$ general-purpose finite-field arithmetic processor

Coprocessor for the final exponentiation

- Highly sequential computation
- Very heterogeneous

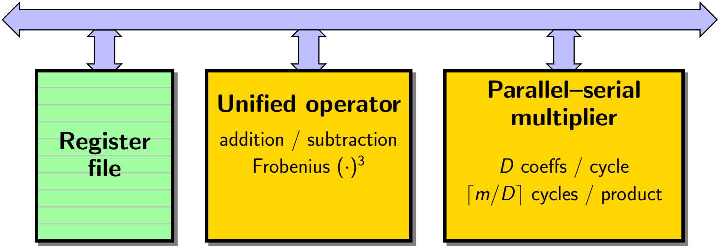
$\Rightarrow$ general-purpose finite-field arithmetic processor



Coprocessor for the final exponentiation

- Highly sequential computation
- Very heterogeneous

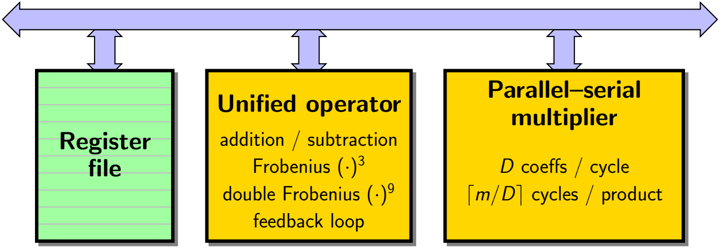
$\Rightarrow$ general-purpose finite-field arithmetic processor



Coprocessor for the final exponentiation

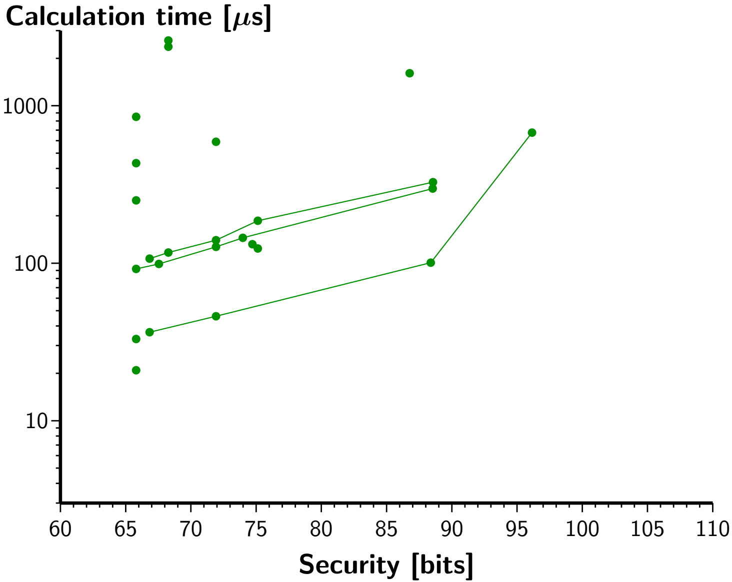
- Highly sequential computation
- Very heterogeneous

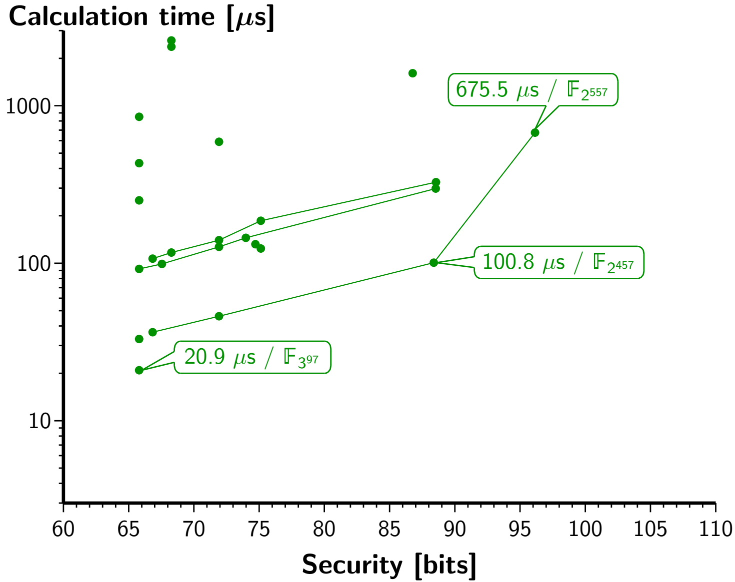
$\Rightarrow$ general-purpose finite-field arithmetic processor



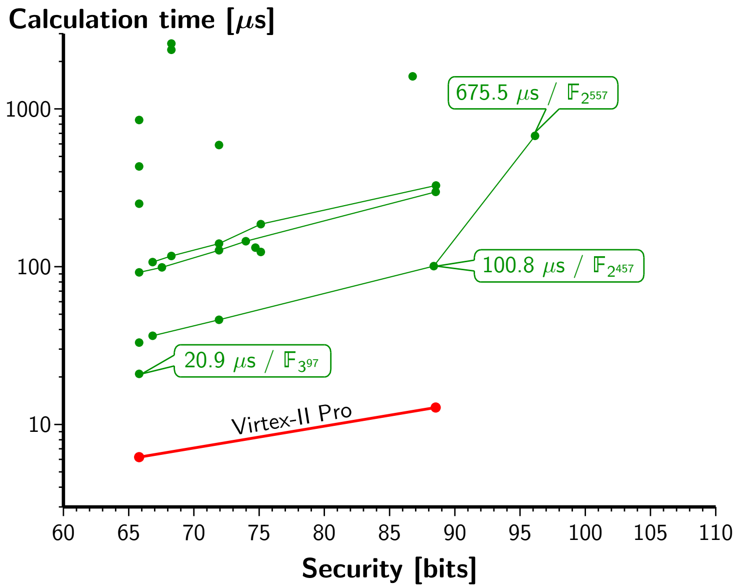
Experimental setup

- Full Tate pairing computation:
  - non-reduced pairing and
  - final exponentiation
- Prototyped on Xilinx Virtex-II Pro and Virtex-4 LX FPGAs
- Post-place-and-route timing and area estimations

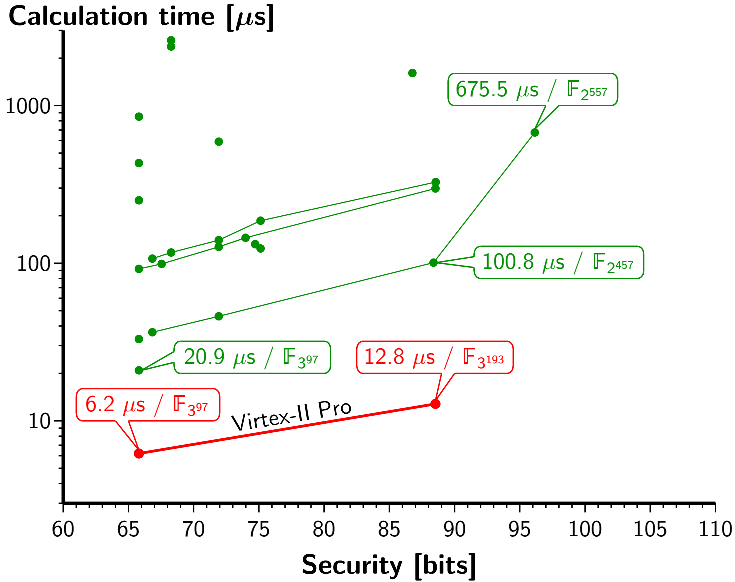

Calculation time [μs]
1000
100
10
60
65
70
75
80
85
90
95
100
105
110
Security [bits]



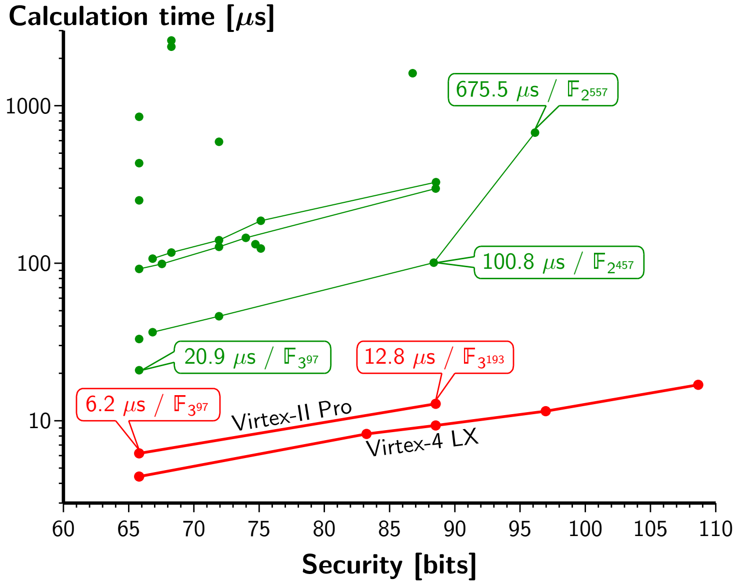
Calculation time [μs]
1000
100
10
675.5 μs / $\mathbb{F}_{2^{557}}$
100.8 μs / $\mathbb{F}_{2^{457}}$
20.9 μs / $\mathbb{F}_{3^{97}}$
60
65
70
75
80
85
90
95
100
105
110
Security [bits]

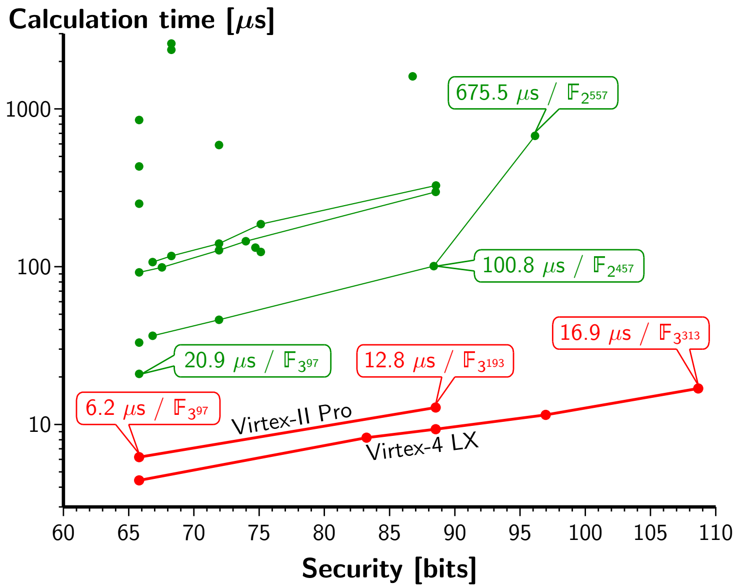


Calculation time [μs]
1000
100
10
675.5 μs / $\mathbb{F}_{2^{557}}$
100.8 μs / $\mathbb{F}_{2^{457}}$
20.9 μs / $\mathbb{F}_{3^{97}}$
Virtex-II Pro
60 65 70 75 80 85 90 95 100 105 110
Security [bits]

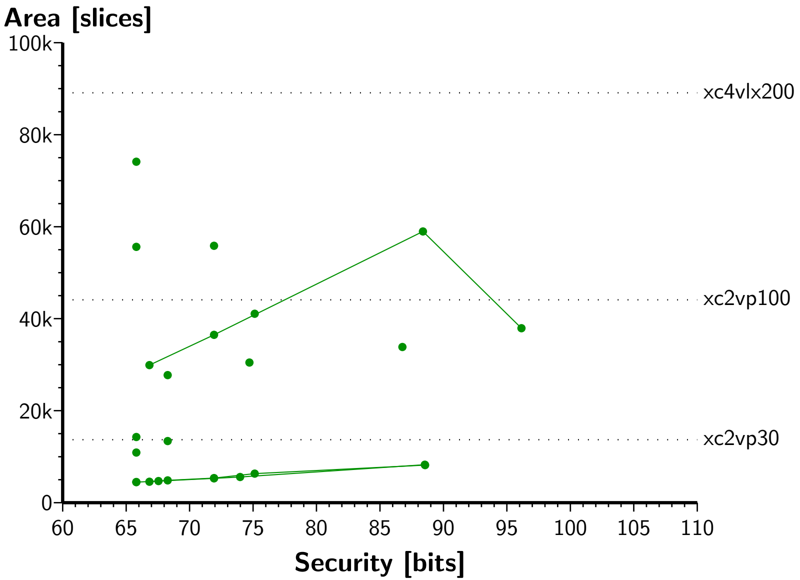


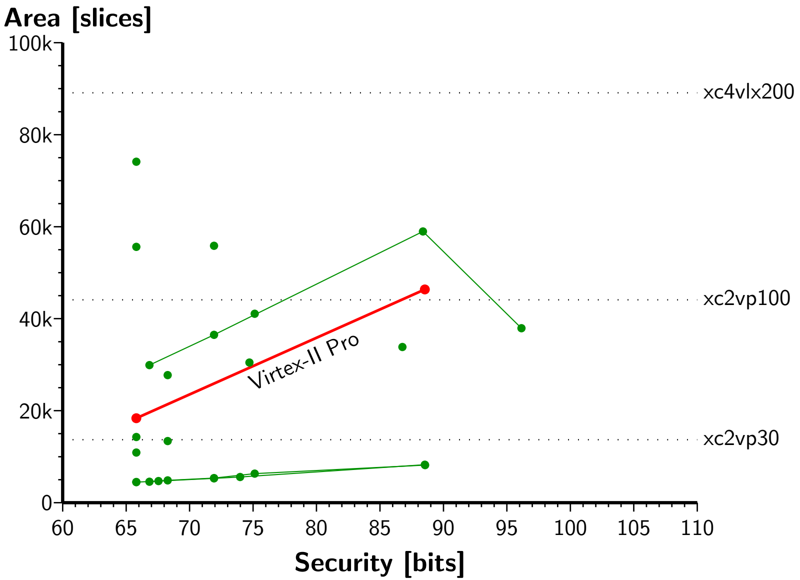
Calculation time [µs]
1000
100
10
675.5 µs / $\mathbb{F}_{2^{557}}$
100.8 µs / $\mathbb{F}_{2^{457}}$
20.9 µs / $\mathbb{F}_{3^{97}}$
12.8 µs / $\mathbb{F}_{3^{193}}$
6.2 µs / $\mathbb{F}_{3^{97}}$
Virtex-II Pro
60 65 70 75 80 85 90 95 100 105 110
Security [bits]

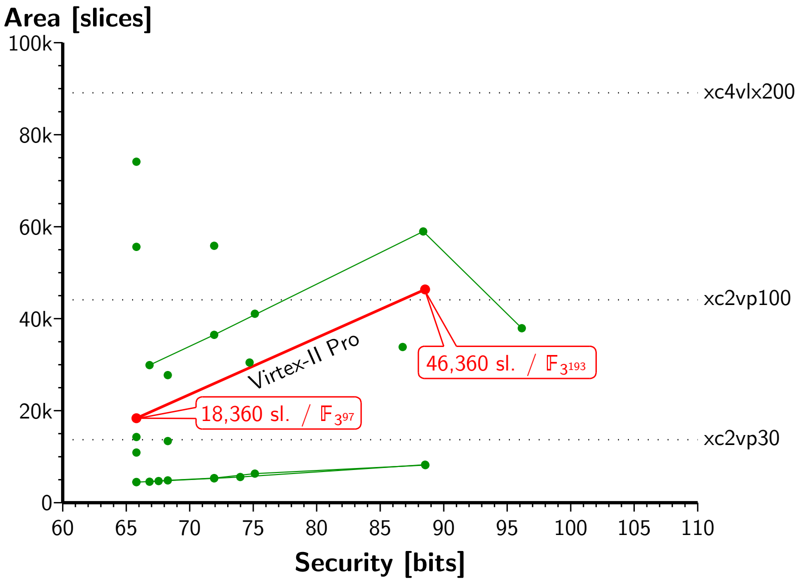

Calculation time [μs]
1000
100
10
60
65
70
75
80
85
90
95
100
105
110
Security [bits]
675.5 μs / $\mathbb{F}_{2^{557}}$
100.8 μs / $\mathbb{F}_{2^{457}}$
20.9 μs / $\mathbb{F}_{3^{97}}$
12.8 μs / $\mathbb{F}_{3^{193}}$
6.2 μs / $\mathbb{F}_{3^{97}}$
Virtex-II Pro
Virtex-4 LX

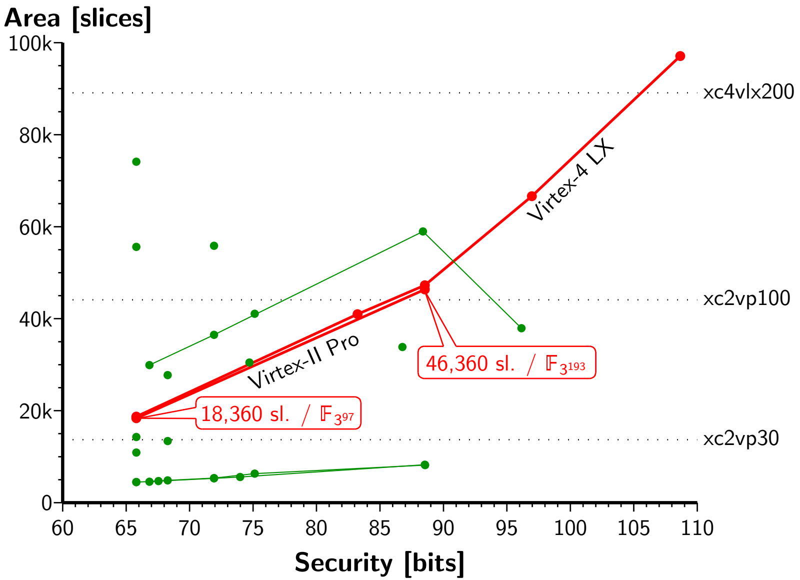


Calculation time [μs]
1000
100
10
675.5 μs / $\mathbb{F}_{2^{557}}$
100.8 μs / $\mathbb{F}_{2^{457}}$
20.9 μs / $\mathbb{F}_{3^{97}}$
16.9 μs / $\mathbb{F}_{3^{313}}$
12.8 μs / $\mathbb{F}_{3^{193}}$
6.2 μs / $\mathbb{F}_{3^{97}}$
Virtex-II Pro
Virtex-4 LX
60 65 70 75 80 85 90 95 100 105 110
Security [bits]

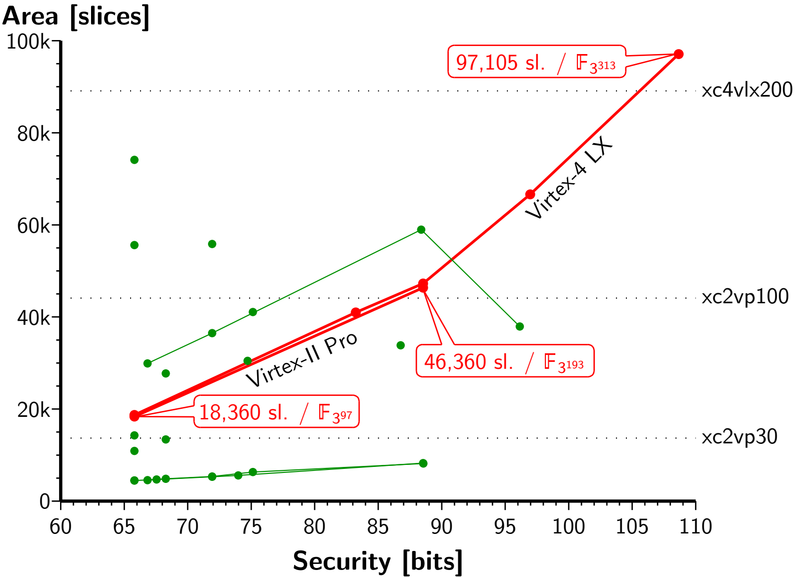

Area [slices]
100k
80k
60k
40k
20k
0
xc4vlx200
xc2vp100
xc2vp30
60
65
70
75
80
85
90
95
100
105
110
Security [bits]

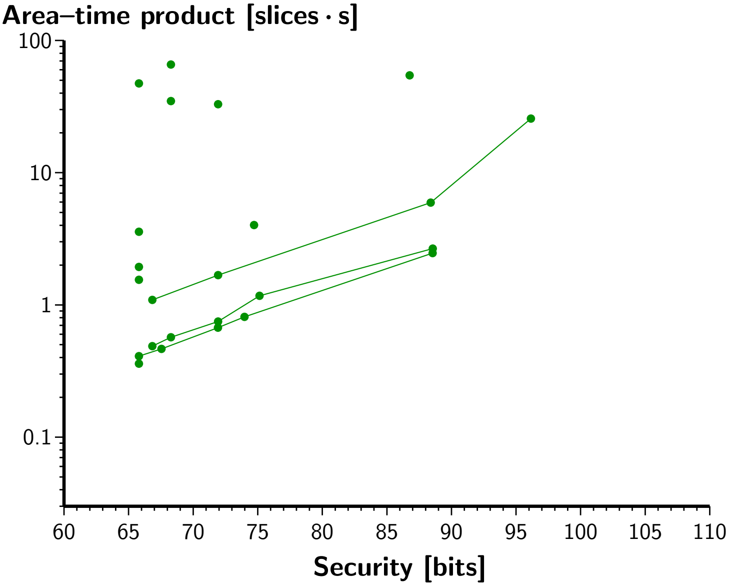

Area [slices]
100k
80k
60k
40k
20k
0
xc4vlx200
xc2vp100
xc2vp30
Virtex-II Pro
60
65
70
75
80
85
90
95
100
105
110
Security [bits]

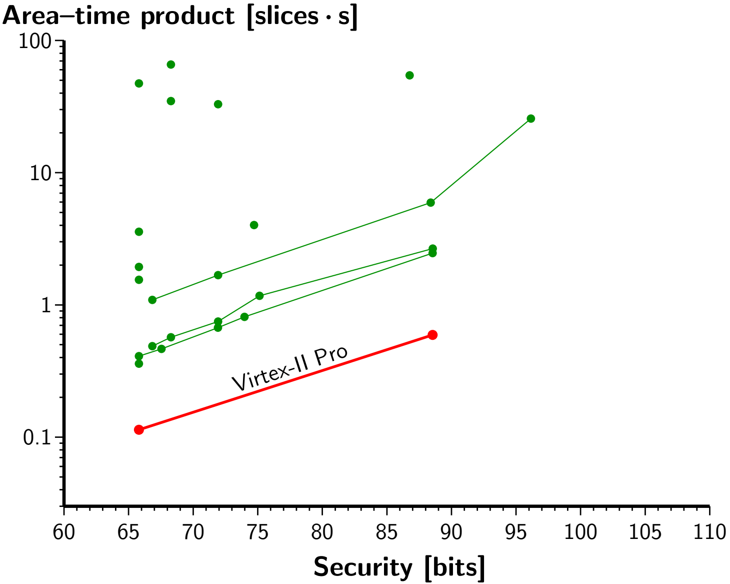

Area [slices]
100k
80k
60k
40k
20k
0
xc4vlx200
xc2vp100
xc2vp30
Virtex-II Pro
46,360 sl. / $\mathbb{F}_{3^{193}}$
18,360 sl. / $\mathbb{F}_{3^{97}}$
60
65
70
75
80
85
90
95
100
105
110
Security [bits]

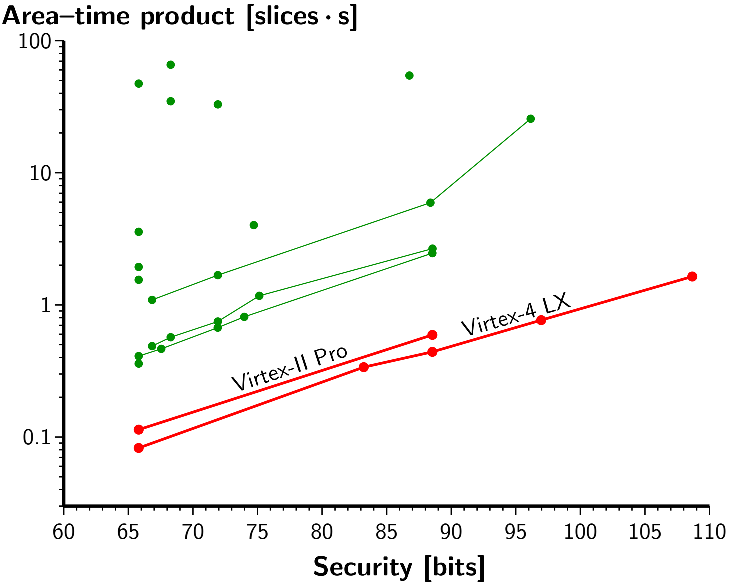


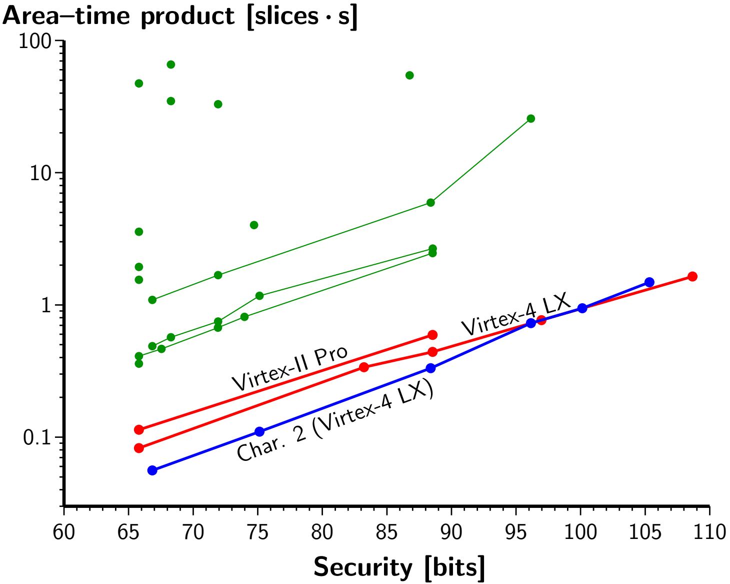
Area [slices]
100k
80k
60k
40k
20k
0
60
65
70
75
80
85
90
95
100
105
110
Security [bits]
xc4vlx200
xc2vp100
xc2vp30
Virtex-4 LX
Virtex-II Pro
46,360 sl. / $\mathbb{F}_{3^{193}}$
18,360 sl. / $\mathbb{F}_{3^{97}}$


Area [slices]
100k
80k
60k
40k
20k
0
60
65
70
75
80
85
90
95
100
105
110
Security [bits]
97,105 sl. / $\mathbb{F}_{3^{313}}$
xc4vlx200
Virtex-4 LX
xc2vp100
Virtex-II Pro
46,360 sl. / $\mathbb{F}_{3^{193}}$
18,360 sl. / $\mathbb{F}_{3^{97}}$
xc2vp30

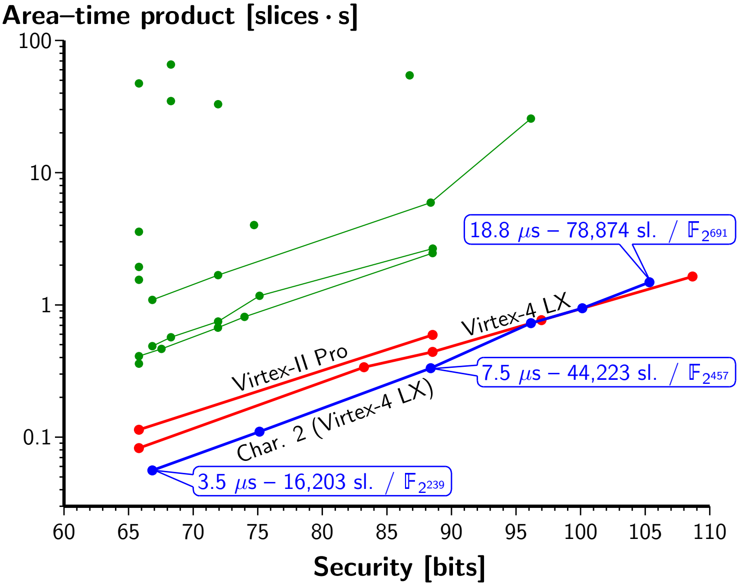

Area–time product [slices · s]
100
10
1
0.1
60
65
70
75
80
85
90
95
100
105
110
Security [bits]


Area–time product [slices · s]
100
10
1
0.1
Virtex-II Pro
60
65
70
75
80
85
90
95
100
105
110
Security [bits]

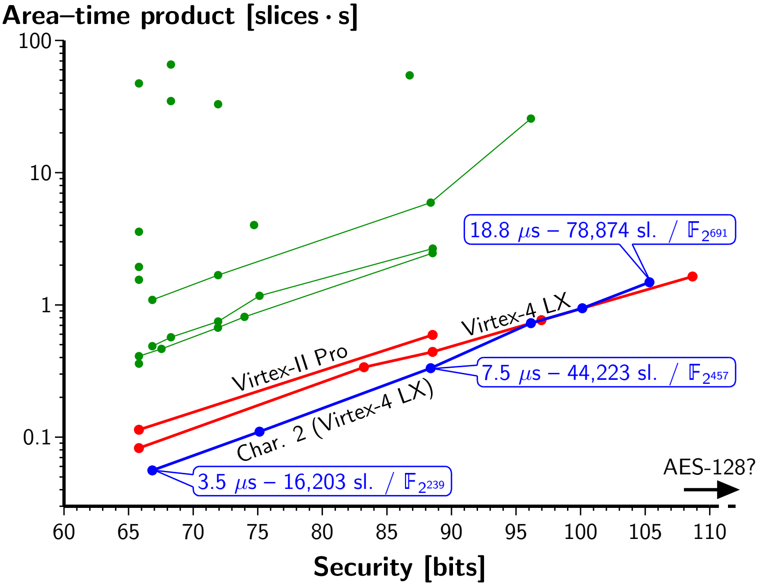

Area–time product [slices · s]
100
10
1
0.1
Virtex-II Pro
Virtex-4 LX
60
65
70
75
80
85
90
95
100
105
110
Security [bits]



Area–time product [slices · s]
100
10
1
0.1
60
65
70
75
80
85
90
95
100
105
110
Security [bits]
Virtex-II Pro
Virtex-4 LX
Char. 2 (Virtex-4 LX)



Area–time product [slices · s]
100
10
1
0.1
60
65
70
75
80
85
90
95
100
105
110
Security [bits]
18.8 μs – 78,874 sl. / $\mathbb{F}_{2^{691}}$
7.5 μs – 44,223 sl. / $\mathbb{F}_{2^{457}}$
3.5 μs – 16,203 sl. / $\mathbb{F}_{2^{239}}$
Virtex-4 LX
Virtex-II Pro
Char. 2 (Virtex-4 LX)


Area–time product [slices · s]
100
10
1
0.1
60
65
70
75
80
85
90
95
100
105
110
Security [bits]
18.8 µs – 78,874 sl. / $\mathbb{F}_{2^{691}}$
Virtex-4 LX
Virtex-II Pro
7.5 µs – 44,223 sl. / $\mathbb{F}_{2^{457}}$
Char. 2 (Virtex-4 LX)
AES-128?
3.5 µs – 16,203 sl. / $\mathbb{F}_{2^{239}}$

Conclusion

- A new architecture for pairing computation
  - two specialized coprocessors
  - bet on parallelizing multiplier
  - based on Karatsuba multiplication scheme
  - importance of architecture–algorithm co-design
  - careful bubble-free scheduling of Miller's loop

Conclusion

- A new architecture for pairing computation
  - two specialized coprocessors
  - bet on parallelizing multiplier
  - based on Karatsuba multiplication scheme
  - importance of architecture–algorithm co-design
  - careful bubble-free scheduling of Miller's loop
- High-performance accelerator
  - the fastest coprocessor (17 μs for 109 bits of security)
  - the best area–time trade-off
  - scales to higher security levels

Future work

- Fully parallel multipliers
  - try other algorithms: Toom–Cook, Montgomery's formulae

Future work

- Fully parallel multipliers
  - try other algorithms: Toom–Cook, Montgomery's formulae
- Final-exponentiation coprocessor
  - full-featured finite-field processor
  - compute the full pairing with it (work in progress)

Future work

- Fully parallel multipliers
  - try other algorithms: Toom–Cook, Montgomery's formulae
- Final-exponentiation coprocessor
  - full-featured finite-field processor
  - compute the full pairing with it (work in progress)
- Toward AES-128 security level
  - characteristic 2 (recently submitted)
  - genus-2 supersingular curves in characteristic 2 (work in progress)
  - Barreto–Naehrig curves (next talks!)

Thank you for your attention

Questions?