4th International Conference on Pairing-Based Cryptography
Yamanaka Hot Spring, Japan — December 15, 2010

Compact hardware for computing the Tate pairing over 128-bit-security supersingular curves

Nicolas Estibals
CARAMEL project-team, LORIA, Nancy Université / CNRS / INRIA
Nicolas.Estibals@loria.fr

Outline of the talk

- Context
- Pairings-friendly curves with 128 bits of security
- Implementation and results

Hardware accelerators for pairing computation

- Pairings are (almost) everywhere!
  - wide range of targets and applications
    - low-resource environment (embedded systems, smart card, ...)
    - high-performance computation (bank server, ...)
  - non-trivial to compute
    - complex mathematical structure
    - finite field arithmetic
    - substantial amount of computation
- Needs in hardware implementation
  - computation not suited to general purpose processor
  - specific targets (*e.g.* smart card)

Hardware accelerators for pairing computation

- Pairings are (almost) everywhere!
  - wide range of targets and applications
    - low-resource environment (embedded systems, smart card, ...)
    - high-performance computation (bank server, ...)
  - non-trivial to compute
    - complex mathematical structure
    - finite field arithmetic
    - substantial amount of computation
- Needs in hardware implementation
  - computation not suited to general purpose processor
  - specific targets (*e.g.* smart card)
- Previous work on FPGA implementations
  - low-security pairings
  - most are performance-oriented designs
- Our goal:
  - AES-128 equivalent security
  - compact accelerator

Tate pairing

- Bilinear pairing:

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$$

Tate pairing

- E elliptic curve over $\mathbb{F}_q$
- ℓ large prime dividing $\#E(\mathbb{F}_q)$
  - in general, $\ell \approx \#E(\mathbb{F}_q)$
  - Hasse's bound : $|\#E(\mathbb{F}_q) - (q+1)| \leqslant 2\sqrt{q}$
  - thus, $\ell \approx q$
- $\mathbb{F}_q$-rational ℓ-torsion of E: $E(\mathbb{F}_q)[\ell] = \{P \in E(\mathbb{F}_q) \mid [\ell]P = \mathcal{O}\}$
- Tate pairing:

$$e : E(\mathbb{F}_q)[\ell] \times \mathbb{G}_2 \to \mathbb{G}_T$$

Tate pairing

- E elliptic curve over $\mathbb{F}_q$
- ℓ large prime dividing $\#E(\mathbb{F}_q)$
  - in general, $\ell \approx \#E(\mathbb{F}_q)$
  - Hasse's bound : $|\#E(\mathbb{F}_q) - (q+1)| \leqslant 2\sqrt{q}$
  - thus, $\ell \approx q$
- $\mathbb{F}_q$-rational ℓ-torsion of E: $E(\mathbb{F}_q)[\ell] = \{P \in E(\mathbb{F}_q) \mid [\ell]P = \mathcal{O}\}$
- Embedding degree: k, the smallest integer s. t. $\ell \mid q^k - 1$
- Tate pairing:

$$e : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_{q^k})[\ell] \to \mathbb{G}_T$$

Tate pairing

- E elliptic curve over $\mathbb{F}_q$
- ℓ large prime dividing $\#E(\mathbb{F}_q)$
  - in general, $\ell \approx \#E(\mathbb{F}_q)$
  - Hasse's bound : $|\#E(\mathbb{F}_q) - (q+1)| \leqslant 2\sqrt{q}$
  - thus, $\ell \approx q$
- $\mathbb{F}_q$-rational ℓ-torsion of E: $E(\mathbb{F}_q)[\ell] = \{P \in E(\mathbb{F}_q) \mid [\ell]P = \mathcal{O}\}$
- Embedding degree: k, the smallest integer s. t. $\ell \mid q^k - 1$
- Set of ℓ-th root of unity: $\mu_\ell = \{u \in \mathbb{F}_{q^k}^* \mid u^\ell = 1\}$
- Tate pairing:

$$e : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_{q^k})[\ell] \to \mu_\ell \subset \mathbb{F}_{q^k}^*$$

Tate pairing

- E elliptic curve over $\mathbb{F}_q$
- ℓ large prime dividing $\#E(\mathbb{F}_q)$
  - in general, $\ell \approx \#E(\mathbb{F}_q)$
  - Hasse's bound : $|\#E(\mathbb{F}_q) - (q+1)| \leqslant 2\sqrt{q}$
  - thus, $\ell \approx q$
- $\mathbb{F}_q$-rational ℓ-torsion of E: $E(\mathbb{F}_q)[\ell] = \{P \in E(\mathbb{F}_q) \mid [\ell]P = \mathcal{O}\}$
- Embedding degree: k, the smallest integer s. t. $\ell \mid q^k - 1$
- Set of ℓ-th root of unity: $\mu_\ell = \{u \in \mathbb{F}_{q^k}^* \mid u^\ell = 1\}$
- Tate pairing:

$$e : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_{q^k})[\ell] \to \mu_\ell \subset \mathbb{F}_{q^k}^*$$

- Compute thanks to Miller's iterative algorithm
  - number of iteration proportional to the size of the field
  - a multiplication over $\mathbb{F}_{q^k}$ at each iteration

General attacks

$$e : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_{q^k})[\ell] \to \mu_\ell \subset \mathbb{F}_{q^k}^*$$

- Pollard's ρ on the torsion subgroup $E[\ell]$
  - $\sqrt{\pi\ell/2} \approx \sqrt{\pi q/2}$ group operations
  - complexity exponential in q

General attacks

$$e : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_{q^k})[\ell] \to \mu_\ell \subset \mathbb{F}_{q^k}^*$$

- Pollard's ρ on the torsion subgroup $E[\ell]$
  - $\sqrt{\pi\ell/2} \approx \sqrt{\pi q/2}$ group operations
  - complexity exponential in q
- Discrete logarithm in finite field multiplicative group $\mathbb{F}_{q^k}^*$
  - FFS or NFS $\to L_{q^k}[1/3, c]$
  - complexity subexponential in q^k

General attacks

$$e : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_{q^k})[\ell] \to \mu_\ell \subset \mathbb{F}_{q^k}^*$$

- Pollard's ρ on the torsion subgroup $E[\ell]$
  - $\sqrt{\pi\ell/2} \approx \sqrt{\pi q/2}$ group operations
  - complexity exponential in q
- Discrete logarithm in finite field multiplicative group $\mathbb{F}_{q^k}^*$
  - FFS or NFS $\to L_{q^k}[1/3, c]$
  - complexity subexponential in q^k
- k acts as a cursor to balance the complexity of the two attacks
- $k = 12$: optimal for the 128-bit security level

Outline of the talk

- Context
- Pairings-friendly curves with 128 bits of security
- Implementation and results

Supersingular elliptic curves Vs. Barreto–Naehrig curves

- Definition:

$$E/\mathbb{F}_3 : \; y^2 = x^3 - x + b, b \neq 0$$

- Supersingular curve

- Definition:

$$E/\mathbb{F}_p : \qquad y^2 = x^3 + b, b \neq 0,$$
$$p = 36\alpha^4 - 36\alpha^3 + 24\alpha^2 - 6\alpha + 1$$

- Ordinary curve

$\Rightarrow$ Simpler curve arithmetic (efficient tripling formulae)

Supersingular elliptic curves Vs. Barreto–Naehrig curves

- Definition:

$$E/\mathbb{F}_3 : \; y^2 = x^3 - x + b, b \neq 0$$

- Supersingular curve

- Definition:

$$E/\mathbb{F}_p : \; y^2 = x^3 + b, b \neq 0,$$
$$p = 36\alpha^4 - 36\alpha^3 + 24\alpha^2 - 6\alpha + 1$$

- Ordinary curve

$\Rightarrow$ Simpler curve arithmetic (efficient tripling formulae)

- Distortion map, modified pairing:

$$\delta : E(\mathbb{F}_q)[\ell] \to E(\mathbb{F}_{q^k})[\ell]$$
$$\hat{e}(P, Q) = e(P, \delta(Q))$$

- No distortion map

$\Rightarrow$ Symmetric pairing (BN cannot be used with all protocols)

Supersingular elliptic curves Vs. Barreto–Naehrig curves

- Definition:

$$E/\mathbb{F}_3: \;\; y^2 = x^3 - x + b, b \neq 0$$

- Supersingular curve

- Definition:

$$E/\mathbb{F}_p: \qquad y^2 = x^3 + b, b \neq 0,$$
$$p = 36\alpha^4 - 36\alpha^3 + 24\alpha^2 - 6\alpha + 1$$

- Ordinary curve

$\Rightarrow$ Simpler curve arithmetic (efficient tripling formulae)

- Distortion map, modified pairing:

$$\delta : E(\mathbb{F}_q)[\ell] \to E(\mathbb{F}_{q^k})[\ell]$$
$$\hat{e}(P, Q) = e(P, \delta(Q))$$

- No distortion map

$\Rightarrow$ Symmetric pairing (BN cannot be used with all protocols)

- Small characteristic field arithmetic
- Modular arithmetic

$\Rightarrow$ No carry, better suited to hardware implementation

Supersingular elliptic curves Vs. Barreto–Naehrig curves

Supersingular elliptic curves

- Definition:

$$E/\mathbb{F}_3 : \; y^2 = x^3 - x + b, b \neq 0$$

- Supersingular curve

Barreto–Naehrig curves

- Definition:

$$E/\mathbb{F}_p : \quad y^2 = x^3 + b, b \neq 0,$$
$$p = 36\alpha^4 - 36\alpha^3 + 24\alpha^2 - 6\alpha + 1$$

- Ordinary curve

$\Rightarrow$ Simpler curve arithmetic (efficient tripling formulae)

Supersingular elliptic curves

- Distortion map, modified pairing:

$$\delta : E(\mathbb{F}_q)[\ell] \to E(\mathbb{F}_{q^k})[\ell]$$
$$\hat{e}(P, Q) = e(P, \delta(Q))$$

Barreto–Naehrig curves

- No distortion map

$\Rightarrow$ Symmetric pairing (BN cannot be used with all protocols)

Supersingular elliptic curves

- Small characteristic field arithmetic

Barreto–Naehrig curves

- Modular arithmetic

$\Rightarrow$ No carry, better suited to hardware implementation

Supersingular elliptic curves

- Small embedding degree ($k = 6$)

Barreto–Naehrig curves

- Optimal embedding degree ($k = 12$)

$\Rightarrow$ Larger field of definition for the same security level. For 128 bits of security:

Supersingular elliptic curves: $\mathbb{F}_q$ with $q \approx 3^{500}$

Barreto–Naehrig curves: $\mathbb{F}_p$ with p a 256-bit prime.

Supersingular elliptic curves

- Definition:

$$E/\mathbb{F}_3 : \; y^2 = x^3 - x + b, b \neq 0$$

- Supersingular curve
  $\Rightarrow$ Simpler curve arithmetic (efficient tripling formulae)
- Distortion map, modified pairing:

$$\delta : E(\mathbb{F}_q)[\ell] \rightarrow E(\mathbb{F}_{q^k})[\ell]$$
$$\hat{e}(P, Q) = e(P, \delta(Q))$$
$$\Rightarrow \text{Symmetric pairing}$$

- Small characteristic field arithmetic
  $\Rightarrow$ No carry, better suited to hardware implementation
- Small embedding degree ($k = 6$)
  $\Rightarrow$ Larger field of definition for the same security level.
  $\mathbb{F}_q$ with $q \approx 3^{500}$

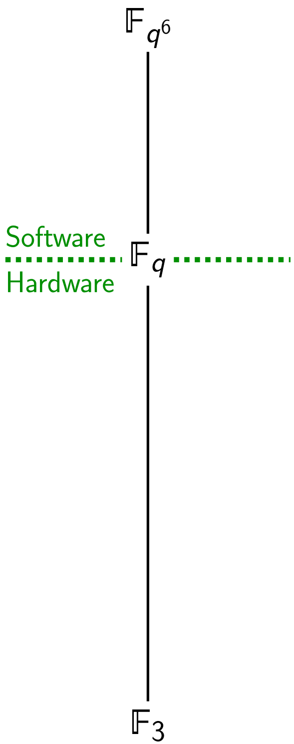
Which field of definition?

$$e : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_{q^6})[\ell] \to \mu_\ell \subset \mathbb{F}_{q^6}^*$$

- Arithmetic of $\mathbb{F}_{q^6}$ over $\mathbb{F}_q$:
  - tower field fixed by pairing construction
  - already optimized by previous works
  - Critical operation: products in $\mathbb{F}_q$

Which field of definition?

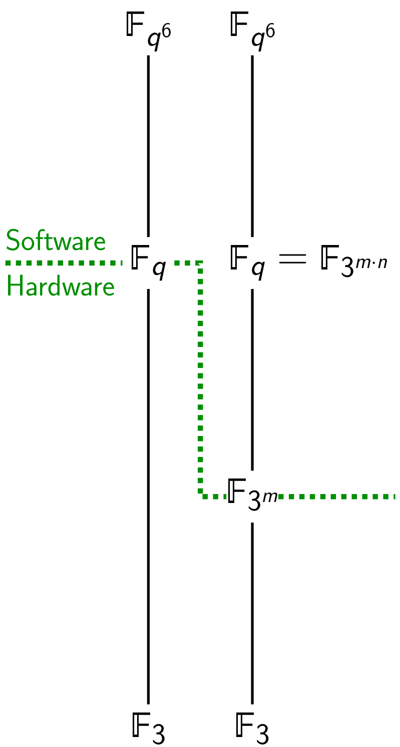
$$e : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_{q^6})[\ell] \rightarrow \mu_\ell \subset \mathbb{F}_{q^6}^*$$



- Arithmetic of $\mathbb{F}_{q^6}$ over $\mathbb{F}_q$:
  - tower field fixed by pairing construction
  - already optimized by previous works
  - Critical operation: products in $\mathbb{F}_q$
- Arithmetic of $\mathbb{F}_q$
  - traditionally implemented in hardware
  - does not scale to the 128-bit security level

Which field of definition?

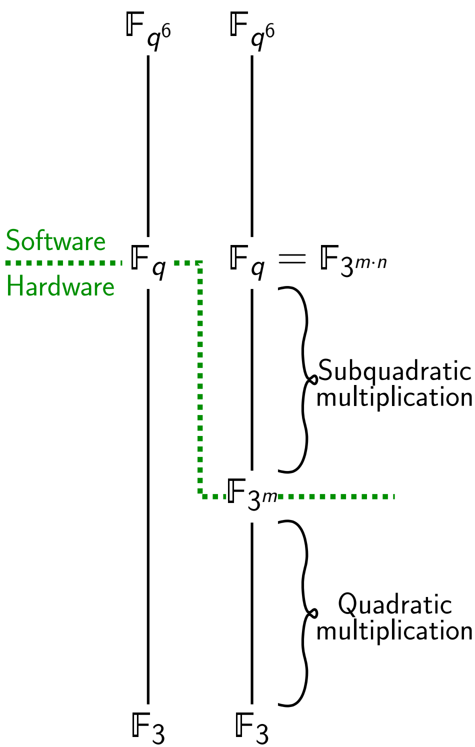
$$e : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_{q^6})[\ell] \to \mu_\ell \subset \mathbb{F}_{q^6}^*$$



- Arithmetic of $\mathbb{F}_{q^6}$ over $\mathbb{F}_q$:
  - tower field fixed by pairing construction
  - already optimized by previous works
  - Critical operation: products in $\mathbb{F}_q$
- Arithmetic of $\mathbb{F}_q$
  - traditionally implemented in hardware
  - does not scale to the 128-bit security level
- Idea: lower the soft/hardware frontier
  - insert $\mathbb{F}_{3^m}$ in the tower field
  - implement it in hardware

Which field of definition?

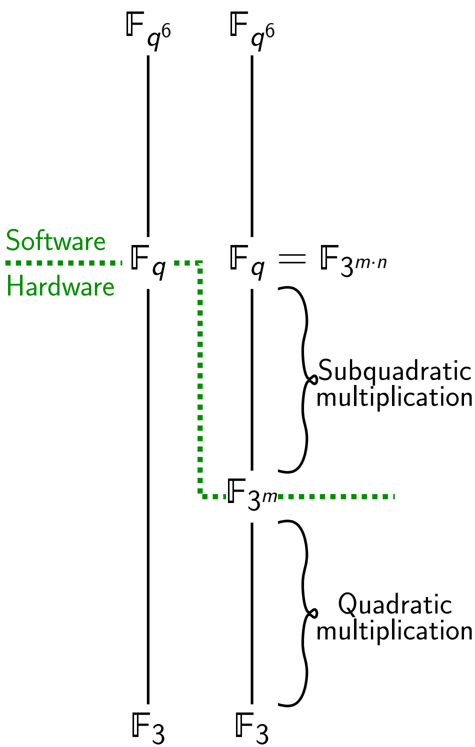
$$e : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_{q^6})[\ell] \to \mu_\ell \subset \mathbb{F}_{q^6}^*$$



- Arithmetic of $\mathbb{F}_{q^6}$ over $\mathbb{F}_q$:
  - tower field fixed by pairing construction
  - already optimized by previous works
  - Critical operation: products in $\mathbb{F}_q$
- Arithmetic of $\mathbb{F}_q$
  - traditionally implemented in hardware
  - does not scale to the 128-bit security level
- Idea: lower the soft/hardware frontier
  - insert $\mathbb{F}_{3^m}$ in the tower field
  - implement it in hardware
  - use subquadratic multiplication algorithm for $\mathbb{F}_q$ over $\mathbb{F}_{3^m}$

Which field of definition?

$$e : E(\mathbb{F}_q)[\ell] \times E(\mathbb{F}_{q^6})[\ell] \to \mu_\ell \subset \mathbb{F}_{q^6}^*$$



- Arithmetic of $\mathbb{F}_{q^6}$ over $\mathbb{F}_q$:
  - tower field fixed by pairing construction
  - already optimized by previous works
  - Critical operation: products in $\mathbb{F}_q$
- Arithmetic of $\mathbb{F}_q$
  - traditionally implemented in hardware
  - does not scale to the 128-bit security level
- Idea: lower the soft/hardware frontier
  - insert $\mathbb{F}_{3^m}$ in the tower field
  - implement it in hardware
  - use subquadratic multiplication algorithm for $\mathbb{F}_q$ over $\mathbb{F}_{3^m}$
- Problem:
  - field with composite extension degree
  - allows some additional attacks

Weil Descent-based attacks

- We now consider:

$$E(\mathbb{F}_{3^{m\cdot n}})[\ell] \text{ with } m \text{ prime and } n \text{ small}$$

- Weil descent (or Weil restriction to scalar) apply:

$$E(\mathbb{F}_{3^{m\cdot n}}) \cong W_E(\mathbb{F}_{3^m})$$

Weil Descent-based attacks

- We now consider:

$$E(\mathbb{F}_{3^{m\cdot n}})[\ell] \text{ with } m \text{ prime and } n \text{ small}$$

- Weil descent (or Weil restriction to scalar) apply:

$$E(\mathbb{F}_{3^{m\cdot n}}) \cong W_E(\mathbb{F}_{3^m})$$

- Gaudry–Hess–Smart attack:
  - $W_E(\mathbb{F}_{3^m})$ might map to $\mathrm{Jac}(\mathcal{C})$, with $\mathcal{C}$ a curve of genus at least n
  - index calculus algorithm: solve DLP in $\tilde{O}((3^m)^{2-\frac{2}{n}})$

Weil Descent-based attacks

- We now consider:

$$E(\mathbb{F}_{3^{m \cdot n}})[\ell] \text{ with } m \text{ prime and } n \text{ small}$$

- Weil descent (or Weil restriction to scalar) apply:

$$E(\mathbb{F}_{3^{m \cdot n}}) \cong W_E(\mathbb{F}_{3^m})$$

- Gaudry–Hess–Smart attack:
  - $W_E(\mathbb{F}_{3^m})$ might map to $\mathrm{Jac}(\mathcal{C})$, with $\mathcal{C}$ a curve of genus at least n
  - index calculus algorithm: solve DLP in $\tilde{O}((3^m)^{2-\frac{2}{n}})$
- Static Diffie–Hellman problem
  - leakage when reusing private key (*e.g.* ElGamal encryption)
  - Granger's attack: complexity in $\tilde{O}((3^m)^{1-\frac{1}{n+1}})$
  - revoke key after a certain amount of use is an effective workaround

Suitable curves for 128-bit security level

			Cost of the attacks (bits)			
p^m	n	$\log_2 \ell$	Pollard's ρ	FFS		
3^{503}	1	697	342	132		
3^{97}	5	338	163	130		
3^{67}	7	612	300	129		
3^{53}	11	672	330	140		
3^{43}	13	764	376	138		

Suitable curves for 128-bit security level

			Cost of the attacks (bits)			
p^m	n	$\log_2 \ell$	Pollard's ρ	FFS	GHS	SDH
3^{503}	1	697	342	132	–	–
3^{97}	5	338	163	130	245	128
3^{67}	7	612	300	129	182	92
3^{53}	11	672	330	140	152	77
3^{43}	13	764	376	138	125	63

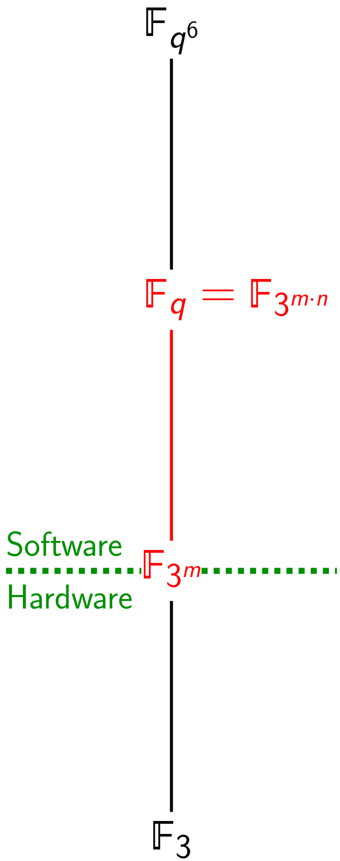
Suitable curves for 128-bit security level

			Cost of the attacks (bits)			
p^m	n	$\log_2 \ell$	Pollard's ρ	FFS	GHS	SDH
3^{503}	1	697	342	132	–	–
3^{97}	5	338	163	130	245	128
3^{67}	7	612	300	129	182	92
3^{53}	11	672	330	140	152	77
3^{43}	13	764	376	138	125	63

Outline of the talk

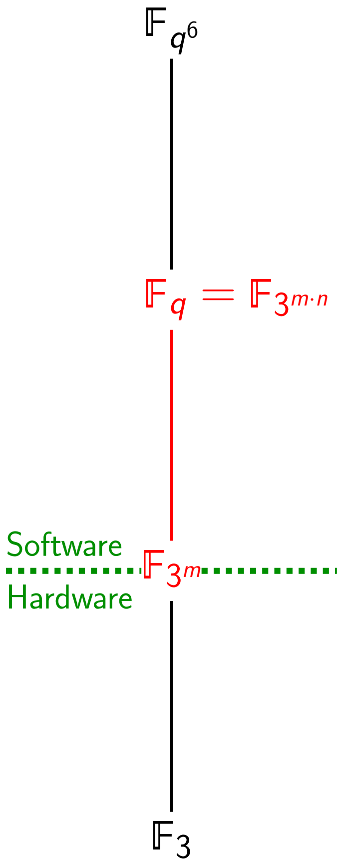
- Context
- Pairings-friendly curves with 128 bits of security
- Implementation and results

Arithmetic of the extension field



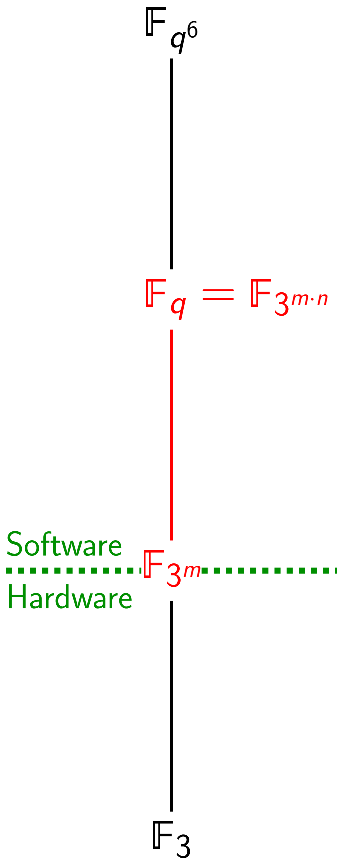
- Polynomial representation: $\mathbb{F}_{3^{m \cdot n}} \cong \mathbb{F}_{3^m}[X]/(f(X))$
  - f irreducible polynomial of degree n
  - addition, cubing (Frobenius automorphism): easy to compute

Arithmetic of the extension field



- Polynomial representation: $\mathbb{F}_{3^{m \cdot n}} \cong \mathbb{F}_{3^m}[X]/(f(X))$
  - f irreducible polynomial of degree n
  - addition, cubing (Frobenius automorphism): easy to compute
- Multiplication
  - bottleneck of pairing computation
  - our test case: multiplication in $\mathbb{F}_{3^{97 \cdot 5}}$
    - ⋆ using "schoolbook" algorithm
    - ⋆ 25 products in $\mathbb{F}_{3^{97}}$
    - ⋆ 24 additions in $\mathbb{F}_{3^{97}}$

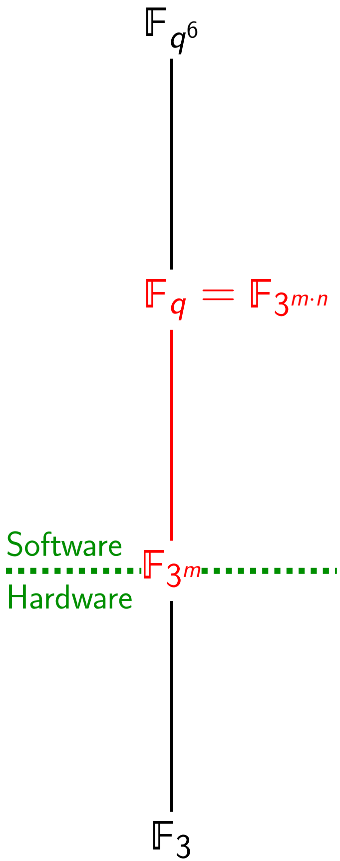
Arithmetic of the extension field



- Polynomial representation: $\mathbb{F}_{3^{m \cdot n}} \cong \mathbb{F}_{3^m}[X]/(f(X))$
  - f irreducible polynomial of degree n
  - addition, cubing (Frobenius automorphism): easy to compute

- Multiplication
  - bottleneck of pairing computation
  - subquadratic multiplication algorithm
    - ⋆ Karatsuba
    - ⋆ Karatsuba with Montgomery's trick
    - ⋆ Montgomery's formulae
    - ⋆ CRT-based algorithms
  - our test case: multiplication in $\mathbb{F}_{3^{97 \cdot 5}}$
    - ⋆ using "schoolbook" algorithm
    - ⋆ 25 products in $\mathbb{F}_{3^{97}}$
    - ⋆ 24 additions in $\mathbb{F}_{3^{97}}$

Arithmetic of the extension field



- Polynomial representation: $\mathbb{F}_{3^{m \cdot n}} \cong \mathbb{F}_{3^m}[X]/(f(X))$
  - f irreducible polynomial of degree n
  - addition, cubing (Frobenius automorphism): easy to compute
- Multiplication
  - bottleneck of pairing computation
  - subquadratic multiplication algorithm
    - ⋆ Karatsuba
    - ⋆ Karatsuba with Montgomery's trick
    - ⋆ Montgomery's formulae
    - ⋆ CRT-based algorithms
  - our test case: multiplication in $\mathbb{F}_{3^{97 \cdot 5}}$
    - ⋆ using CRT-based algorithm
    - ⋆ 12 products in $\mathbb{F}_{3^{97}}$
    - ⋆ 53 additions in $\mathbb{F}_{3^{97}}$

Experimental setup

- Full Tate pairing computation over $E(\mathbb{F}_{3^{97 \cdot 5}})$

	$\times$	$+$	$(.)^3$
$\mathbb{F}_{3^{97}}$	37289	253314	21099

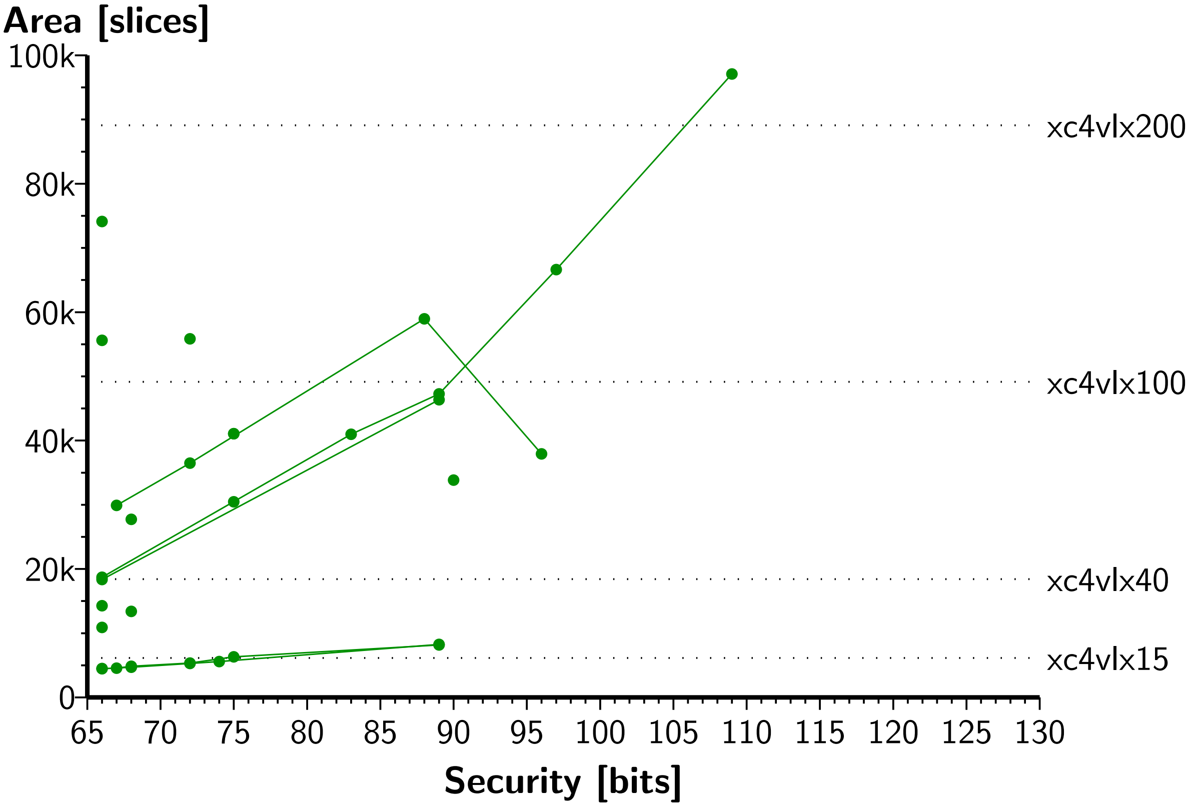
Experimental setup

- Full Tate pairing computation over $E(\mathbb{F}_{3^{97 \cdot 5}})$

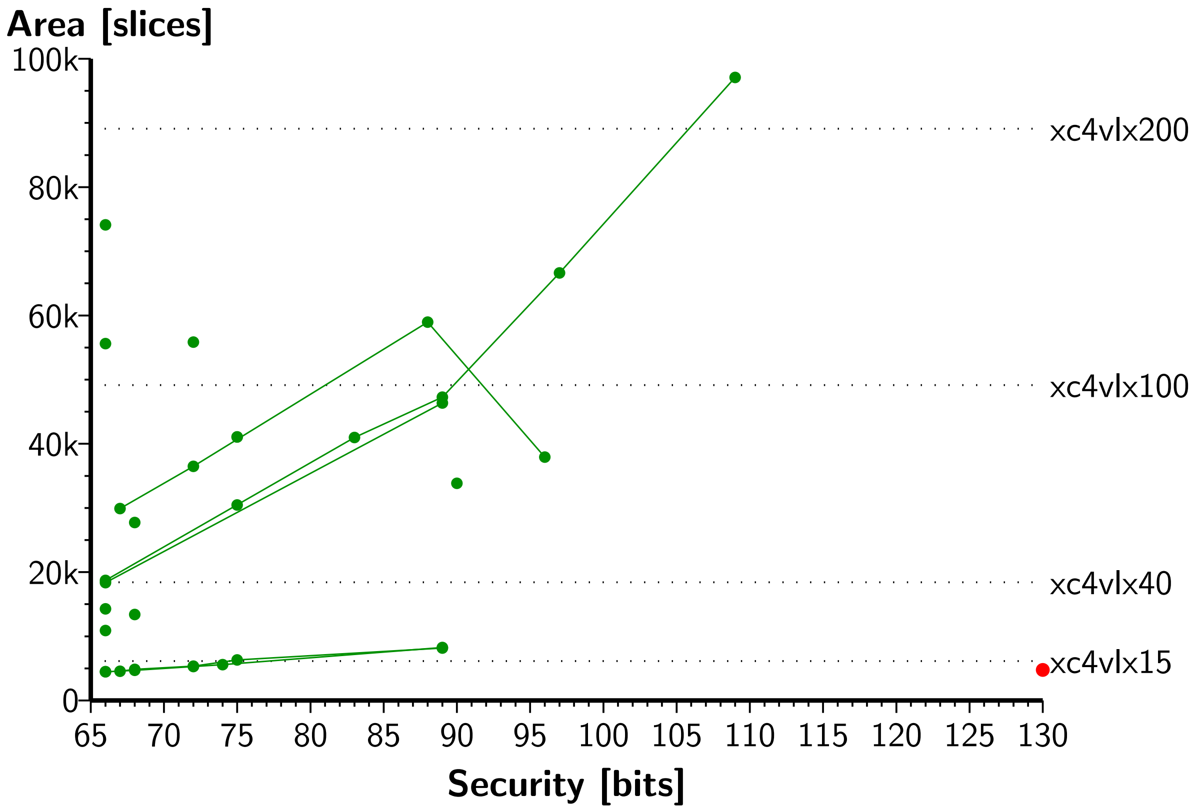
	$\times$	$+$	$(.)^3$
$\mathbb{F}_{3^{97}}$	37289	253314	21099

- Finite field coprocessor
  - Prototyped on Xilinx Virtex-4 LX FPGAs
  - Post-place-and-route timing and area estimations

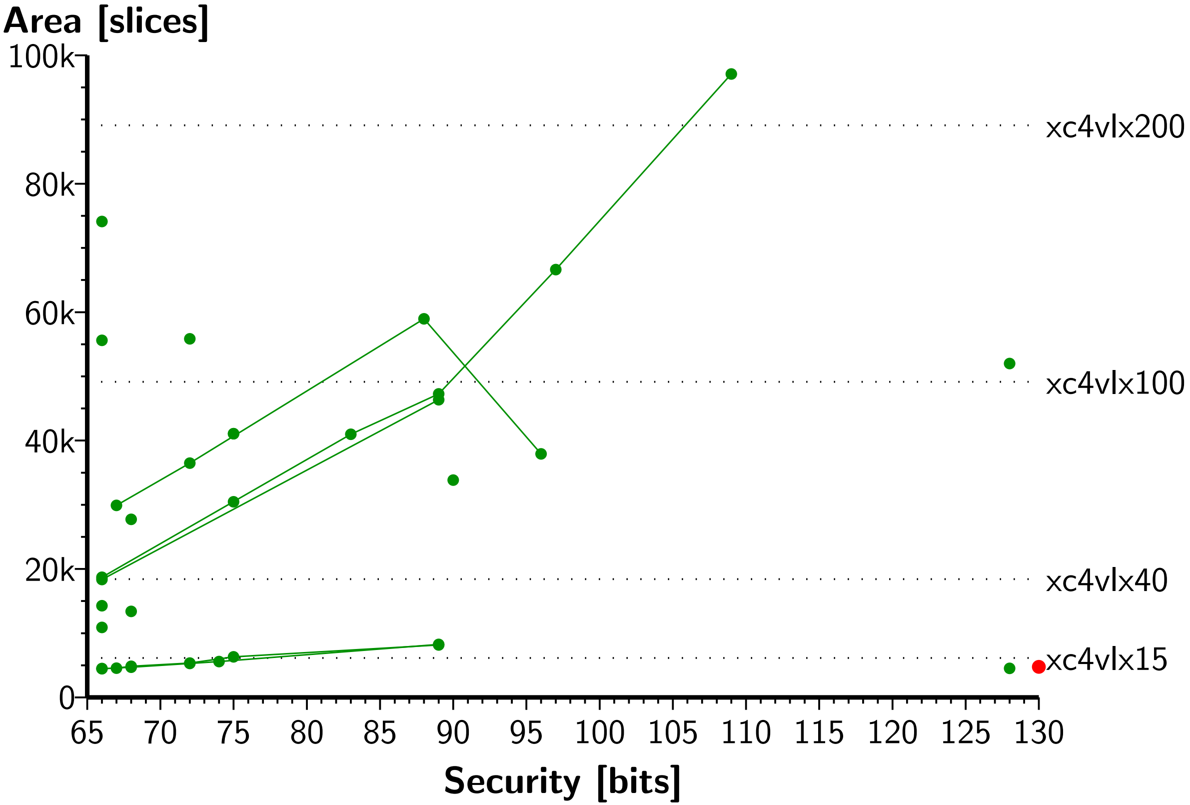
Area


Area [slices]
100k
80k
60k
40k
20k
0
65
70
75
80
85
90
95
100
105
110
115
120
125
130
Security [bits]
xc4vlx200
xc4vlx100
xc4vlx40
xc4vlx15

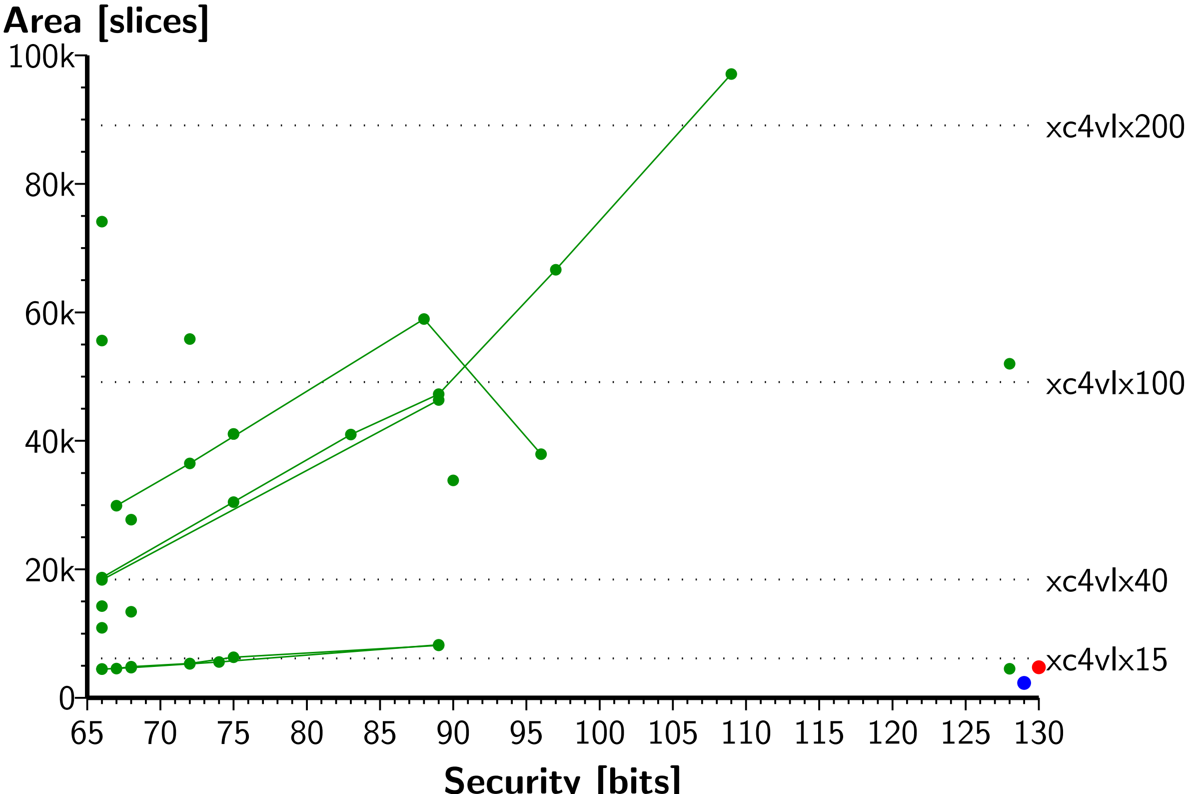
Area


Area [slices]
100k
80k
60k
40k
20k
0
xc4vlx200
xc4vlx100
xc4vlx40
xc4vlx15
65
70
75
80
85
90
95
100
105
110
115
120
125
130
Security [bits]

Area

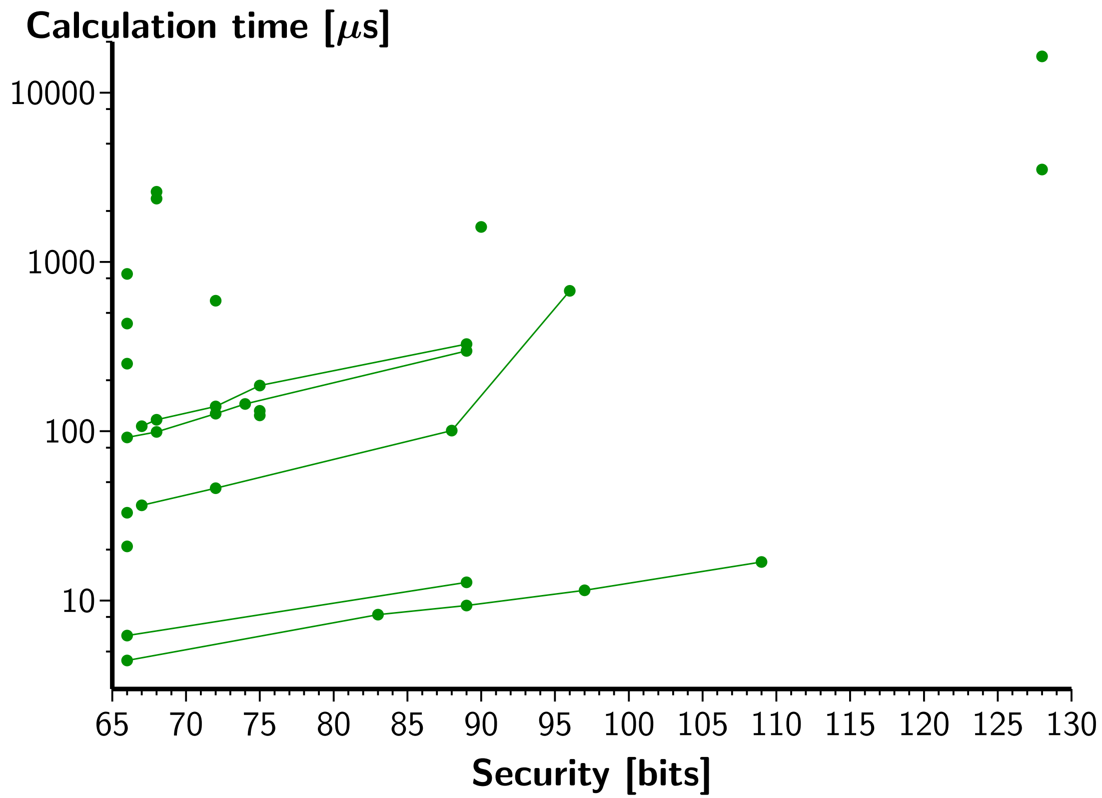

Area [slices]
100k
80k
60k
40k
20k
0
65 70 75 80 85 90 95 100 105 110 115 120 125 130
Security [bits]
xc4vlx200
xc4vlx100
xc4vlx40
xc4vlx15

Area

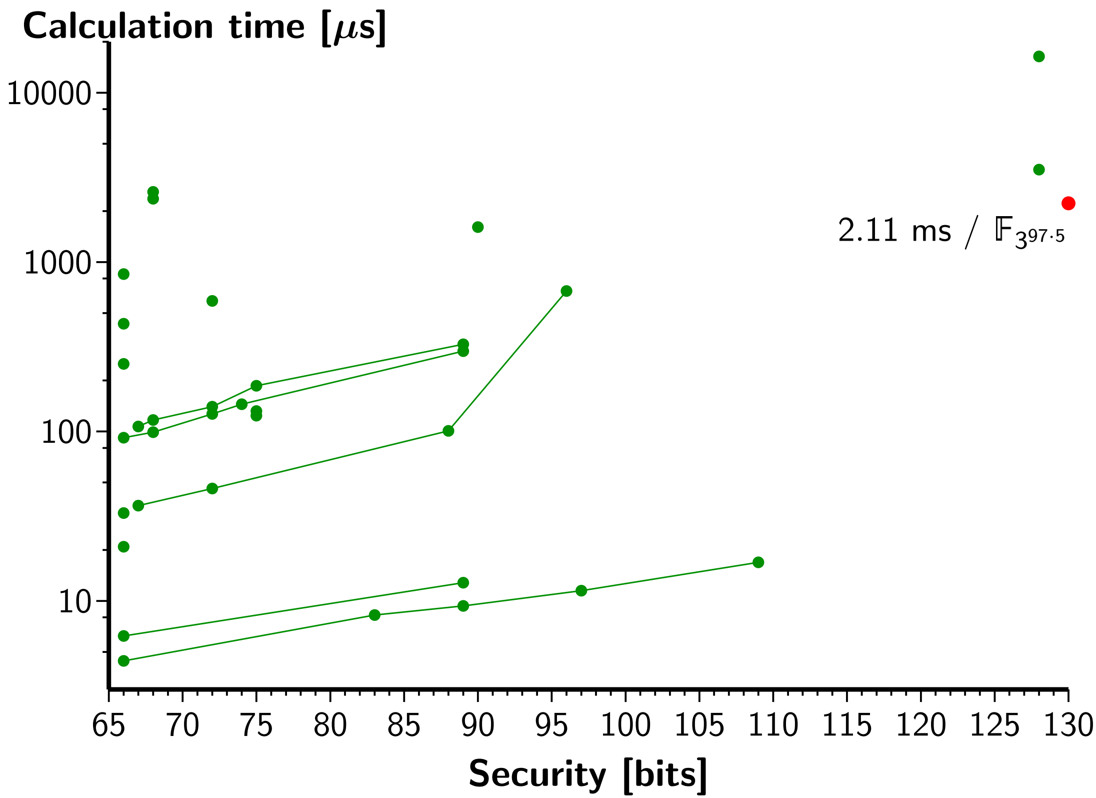


Area [slices]
100k
80k
60k
40k
20k
0
xc4vlx200
xc4vlx100
xc4vlx40
xc4vlx15
65 70 75 80 85 90 95 100 105 110 115 120 125 130
Security [bits]

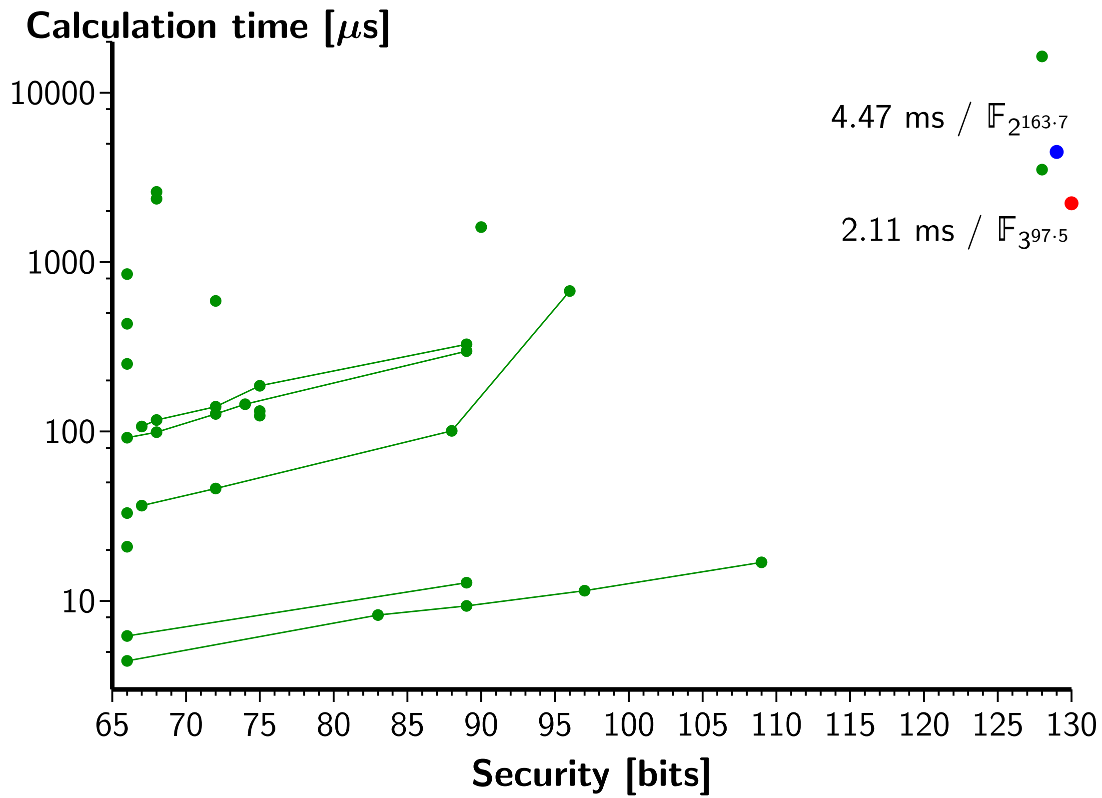
Calculation time



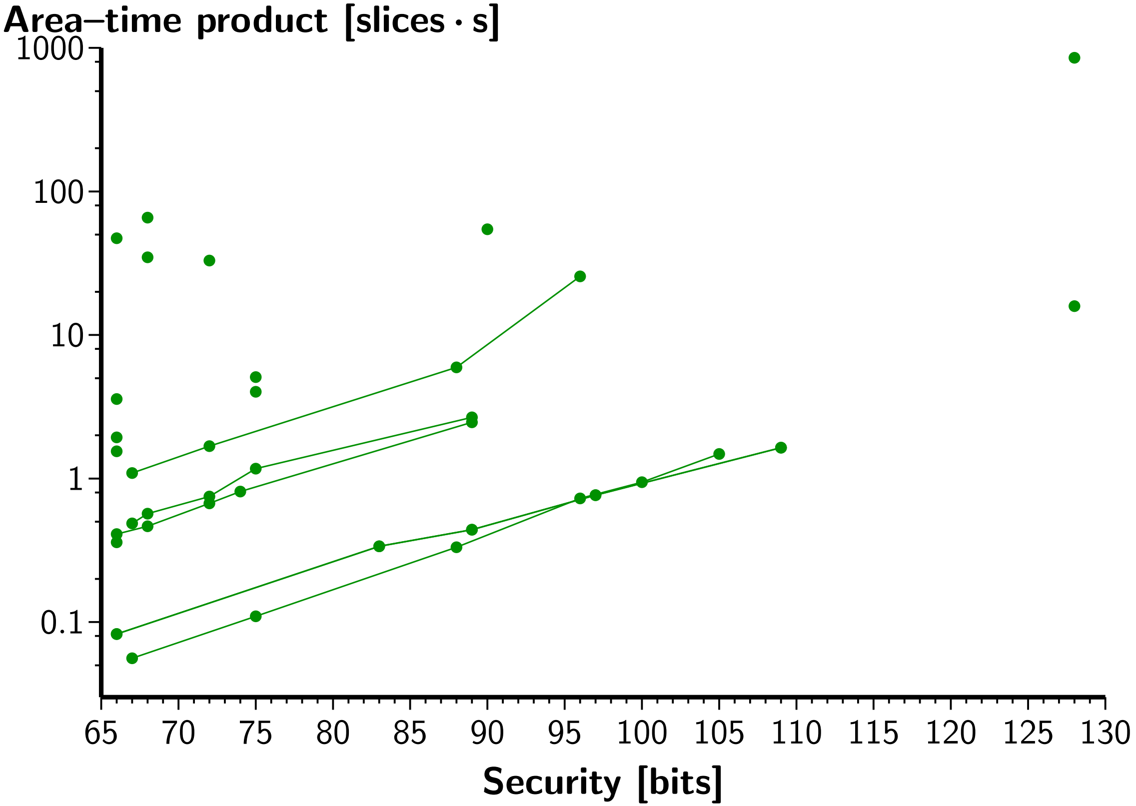
Calculation time



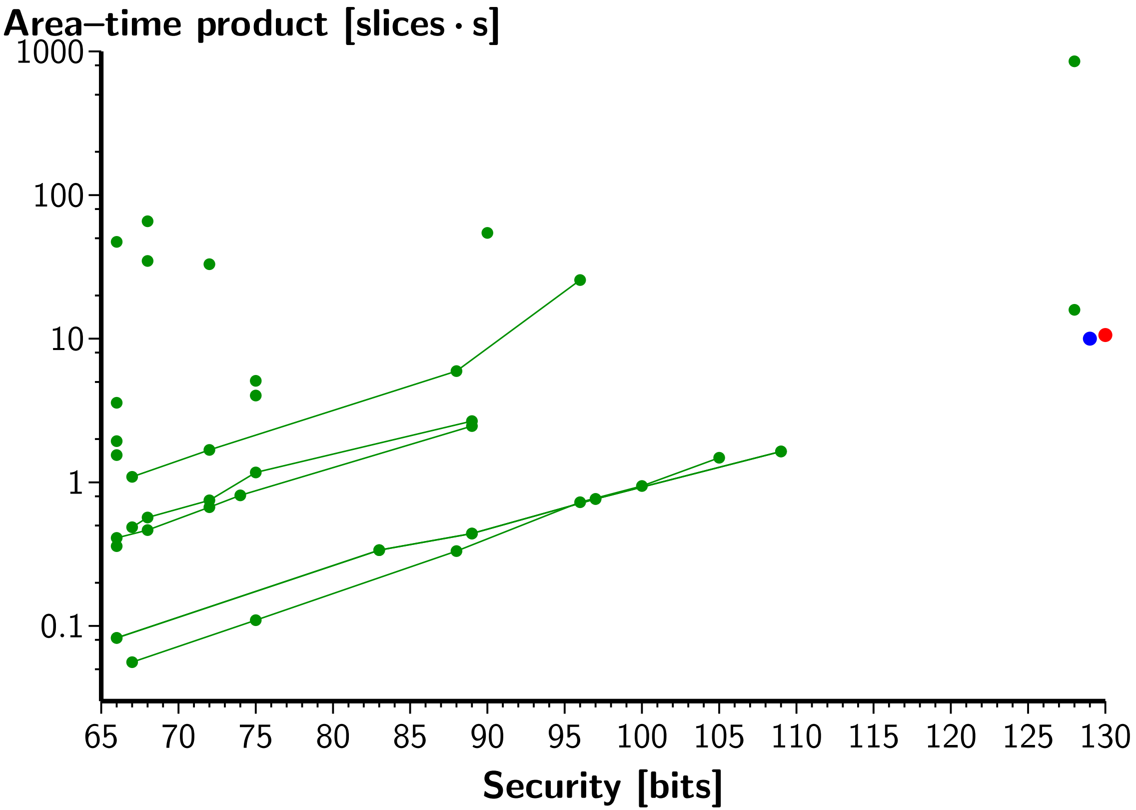
Calculation time



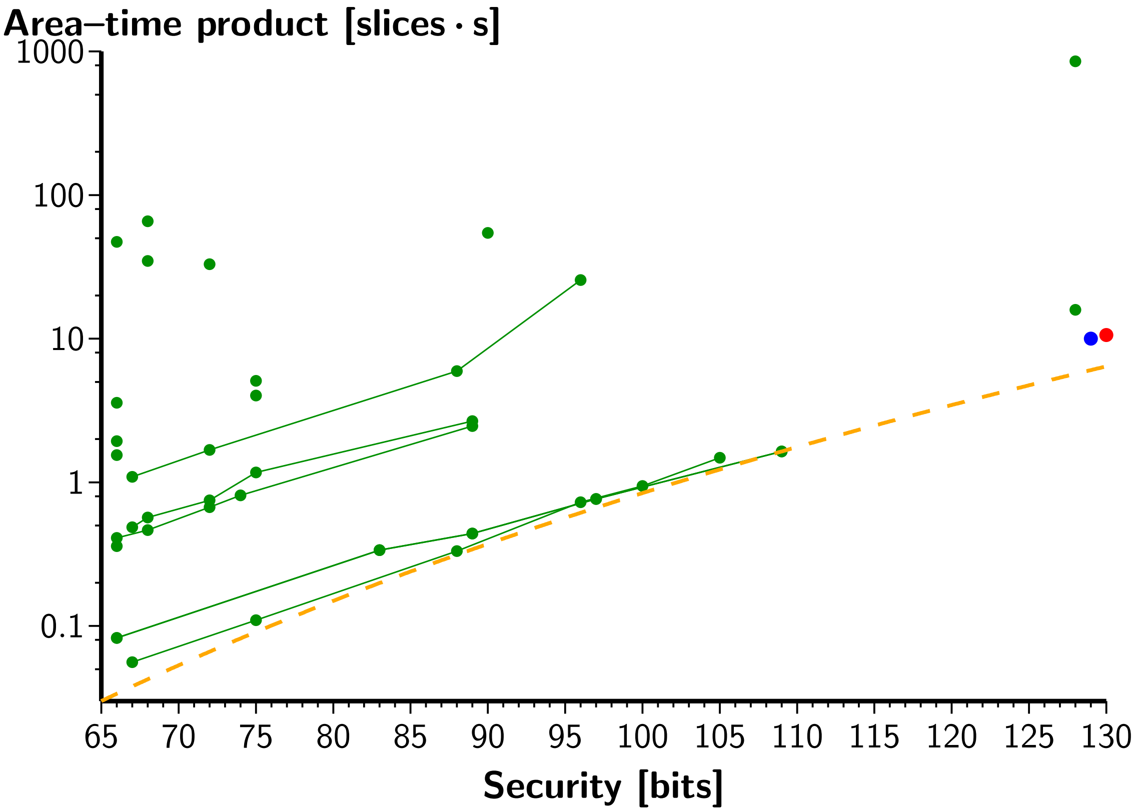
Area–Time product



Area–Time product



Area–Time product



Comparison with ASIC and software

	Supersingular curves	BN-curves
FPGA	2.11 ms (This Work)	52 ms (Ghosh *et al.*, 2010)
ASIC	–	2.91 ms (Fan *et al.*, 2009)
Software (2.4 GHz Intel Core2)	7.59 ms (Beuchat *et al.*, 2009)	0.92 ms (Aranha *et al.*, 2010)

Conclusion

- Compact, yet reasonably fast, accelerator for pairings with 128 bits of security
  - supersingular elliptic curve
  - low characteristic
  - take advantage of the sub-optimal k to implement efficient field arithmetic

Conclusion

- Compact, yet reasonably fast, accelerator for pairings with 128 bits of security
  - supersingular elliptic curve
  - low characteristic
  - take advantage of the sub-optimal k to implement efficient field arithmetic
- Implement this pairing on more curves:
  - better understanding of the software/hardware frontier
  - hopefully improve performance
  - try higher security level
  - study genus-2 supersingular curves

Thank you for your attention
Questions?