AprèsSQI: A Pretty Rad Extension to Signing in SQIsign

Maria Corte-Real Santos, **Jonathan Komada Eriksen**, Michael Meyer, Krijn Reijnders

Goals

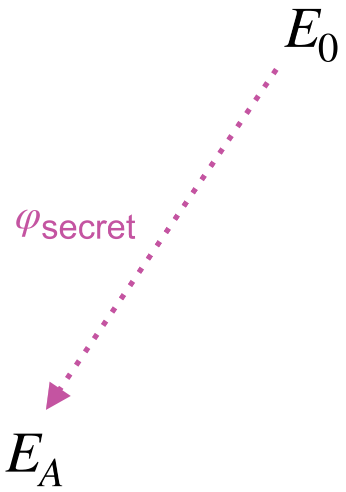
- Original **SQIsign** is ideal for applications where each signature gets verified many times.
    - Tiny public key and signature
    - Relatively fast and easy verification
    - Complex and costly signing

- This work aims to make verification as fast as possible.

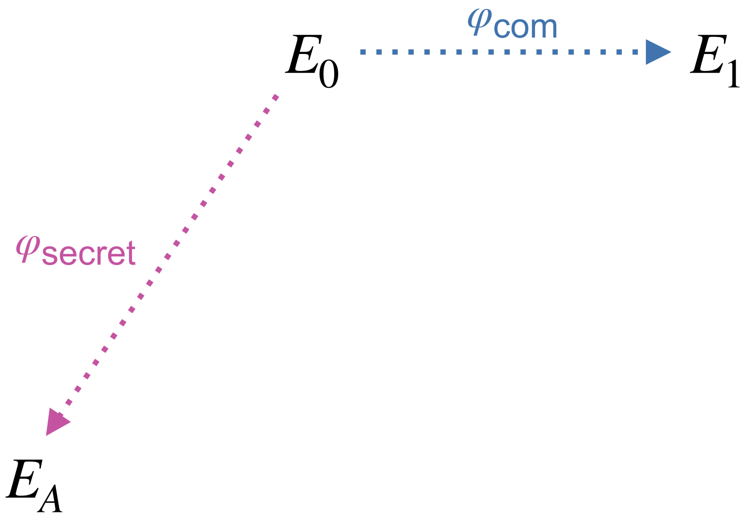
Verifying a

SQISIGN SIGNATURE

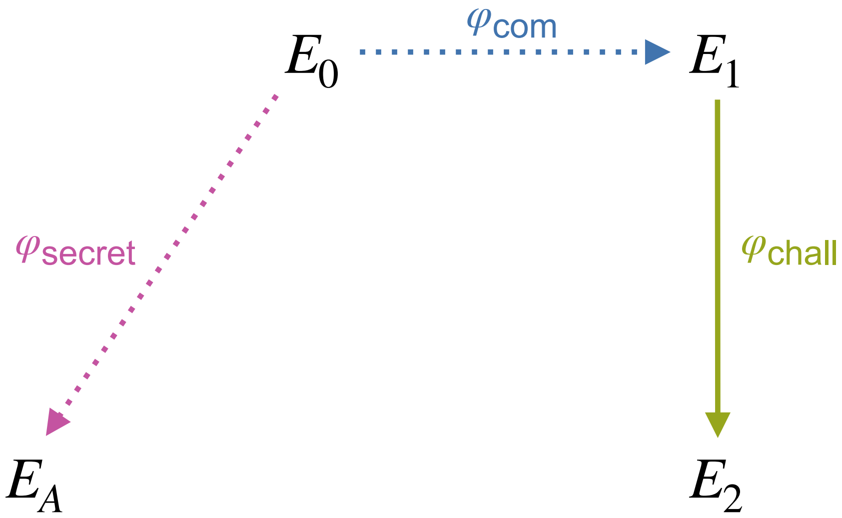
SQIsign Signature - Key Generation



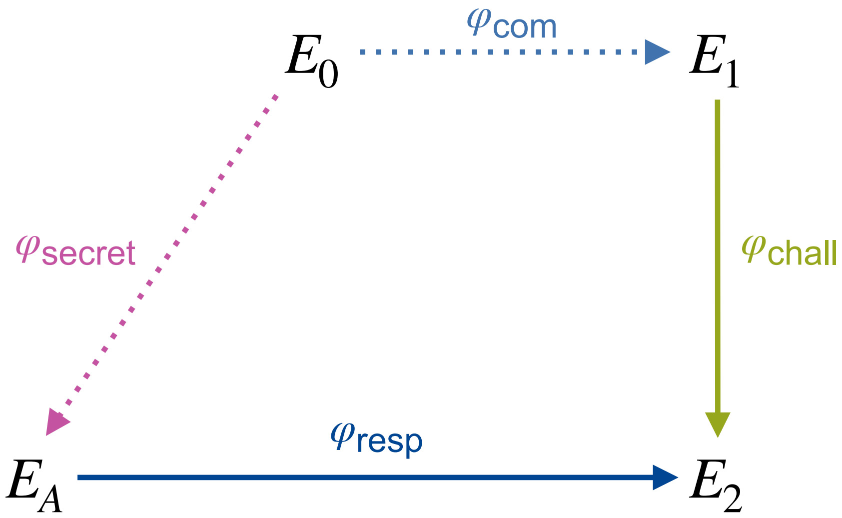
SQIsign Signature - Commitment



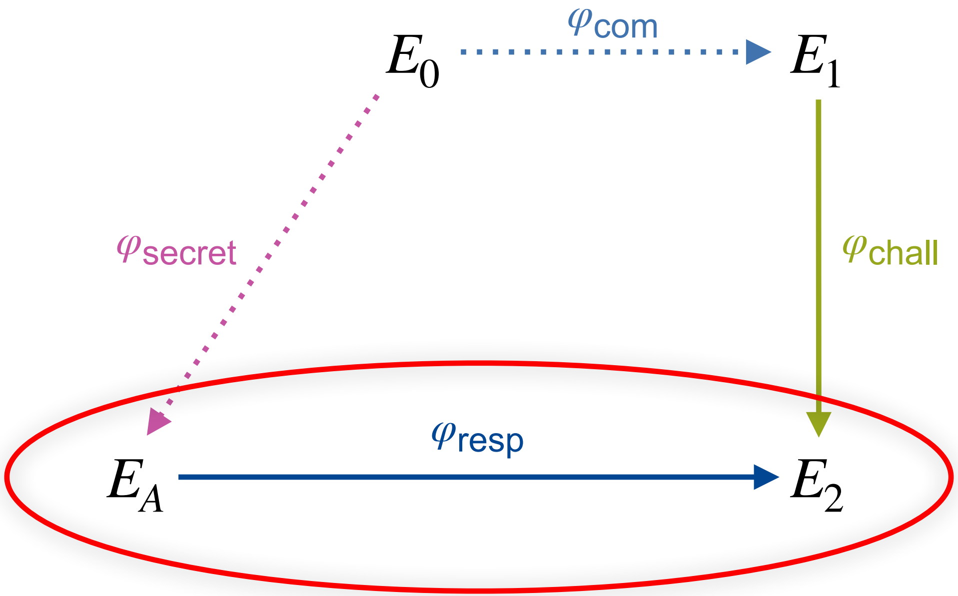
SQIsign Signature - Challenge



SQIsign Signature - Response



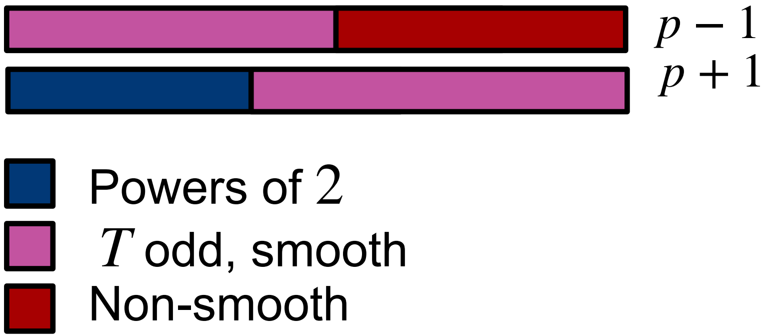
SQIsign Signature - Response



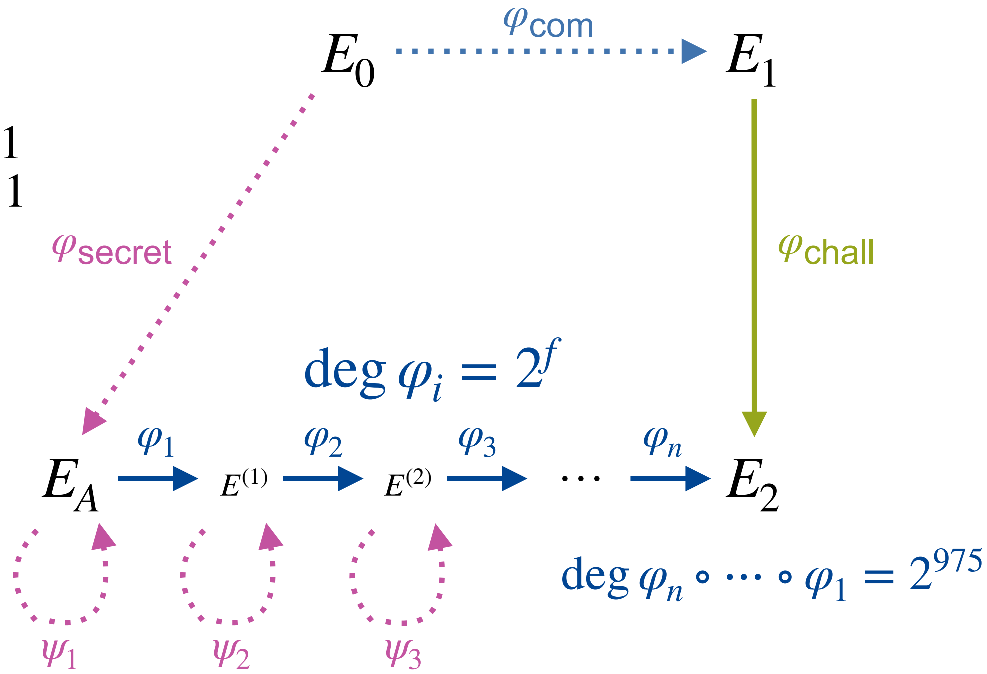
SQIsign Signature - Response

Requirement on prime:

$2^f \mid p+1$ $\qquad$ $T \mid p^2-1$

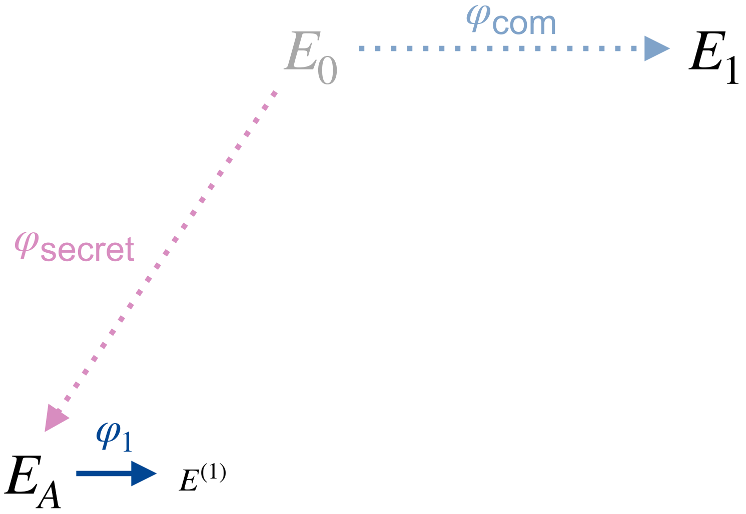


$\deg \psi_i = T^2$



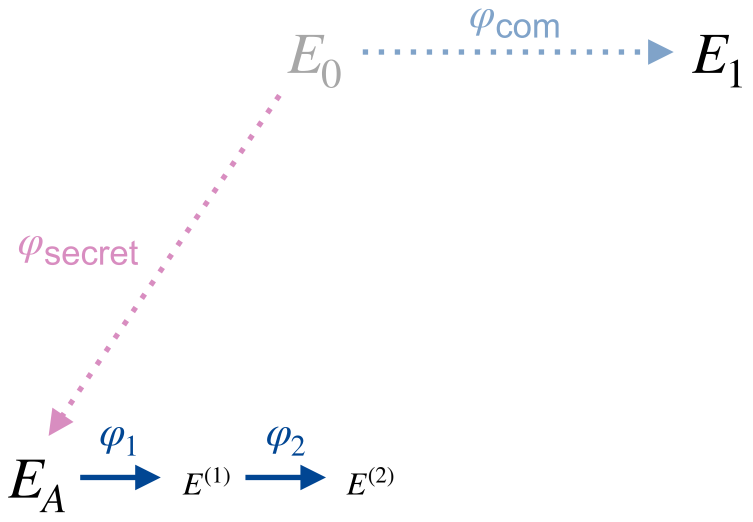
Verifying SQIsign Signature

- In SQIsign: $f = 75$
- $\varphi_{\text{resp}} : K_1, K_2, \ldots, K_{13}$



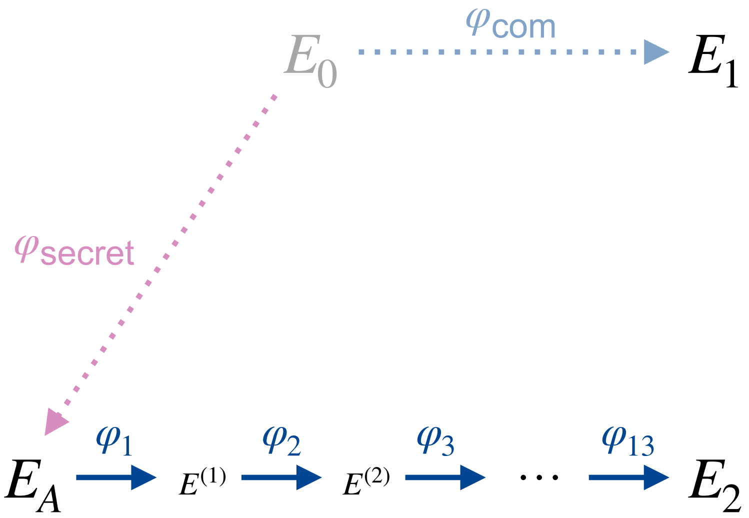
Verifying SQIsign Signature

- In SQIsign: $f = 75$
- $\varphi_{\text{resp}} : K_1, K_2, \ldots, K_{13}$



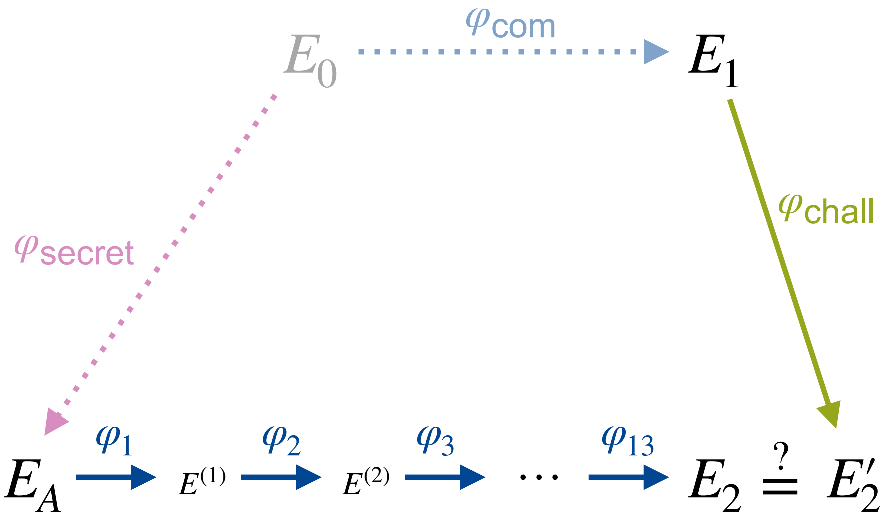
Verifying SQIsign Signature

- In SQIsign: $f = 75$
- $\varphi_{\text{resp}} : K_1, K_2, \ldots, K_{13}$



Verifying SQIsign Signature

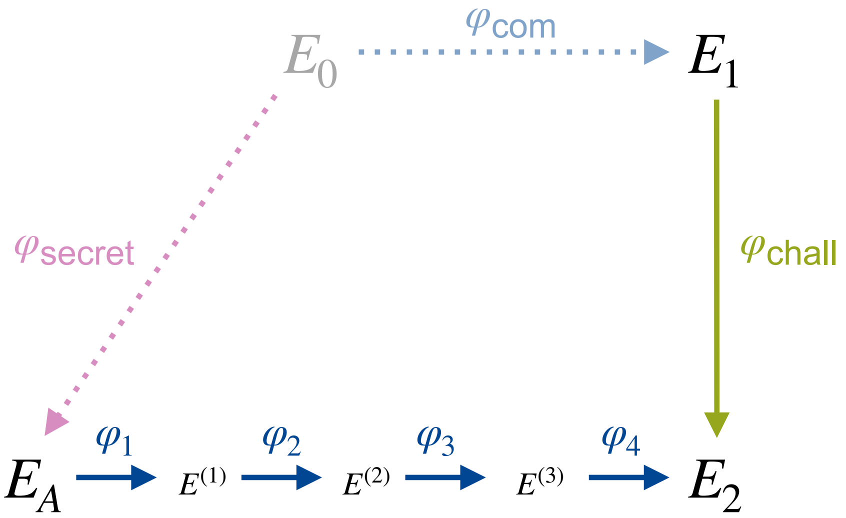
- In SQIsign: $f = 75$
- $\varphi_{\text{resp}} : K_1, K_2, \ldots, K_{13}$
- $K_{\text{chall}} := H(E_1, m)$
- $\varphi_{\text{chall}} : E_1 \to E_2'$



Advantage of bigger f

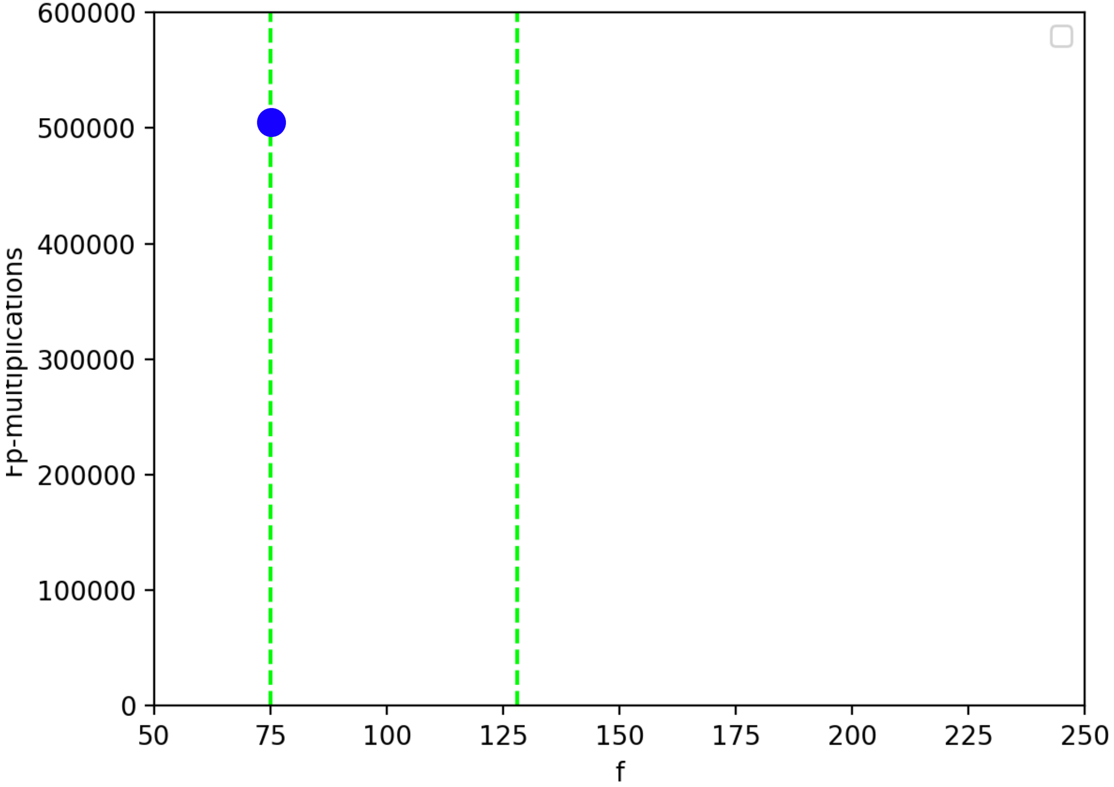
- In SQIsign: $f = 75$
- $\varphi_{\text{resp}} : K_1, K_2, \ldots, K_{13}$

- E.g. $f = 250$ gives 4 points.

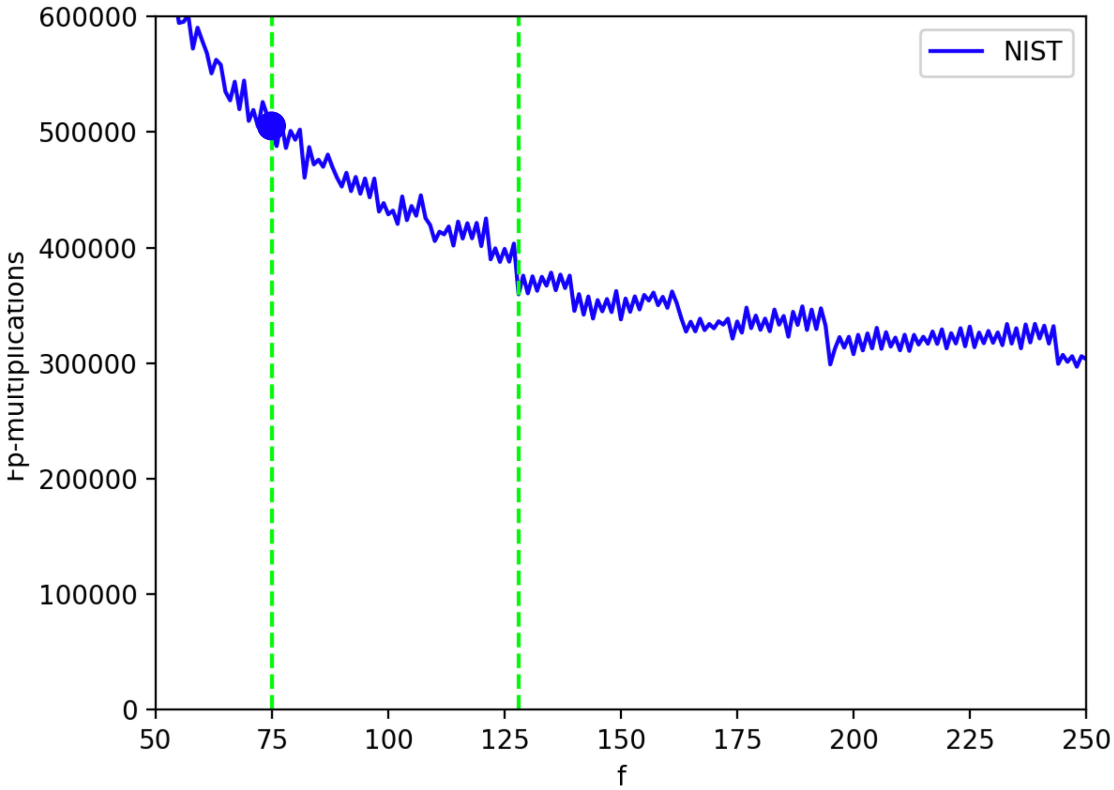



BENCHMARKS

Effect of larger f



Effect of larger f



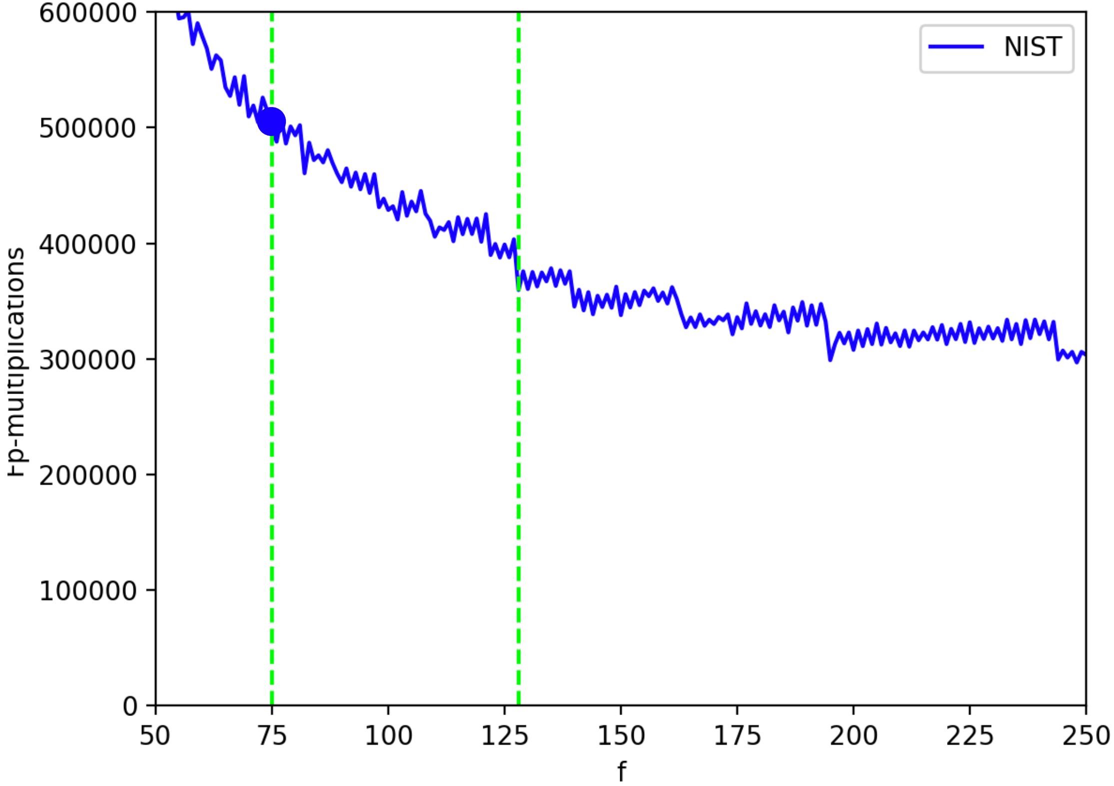
Other Optimisations

- Several low-level optimisations.
  - Faster basis generation.
  - Faster kernel point computation.

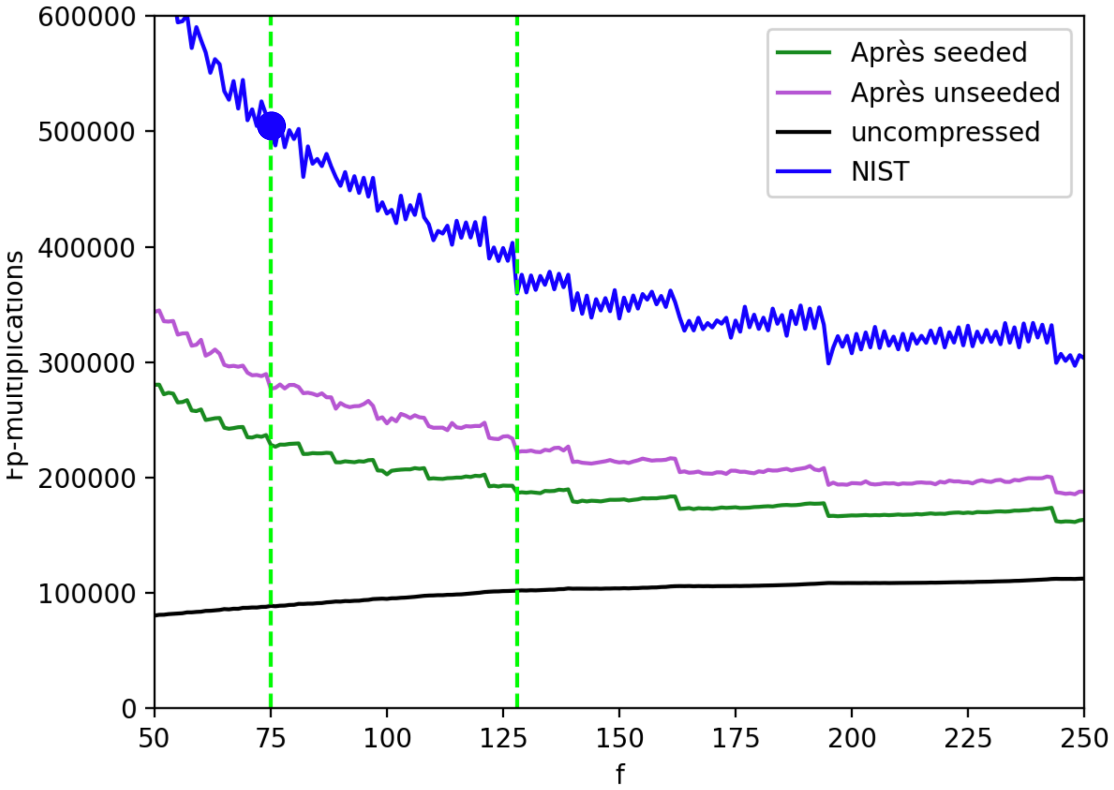
Other Optimisations

- Several low-level optimisations.
  - Faster basis generation.
  - Faster kernel point computation.
- Size-speed tradeoffs.
  - Seeds for basis generation.
  - Uncompressed signatures.
    - Compressed NIST-signatures: 177 B
    - Uncompressed NIST : 896 B
    - Uncompressed : 322 B

Effect of larger f

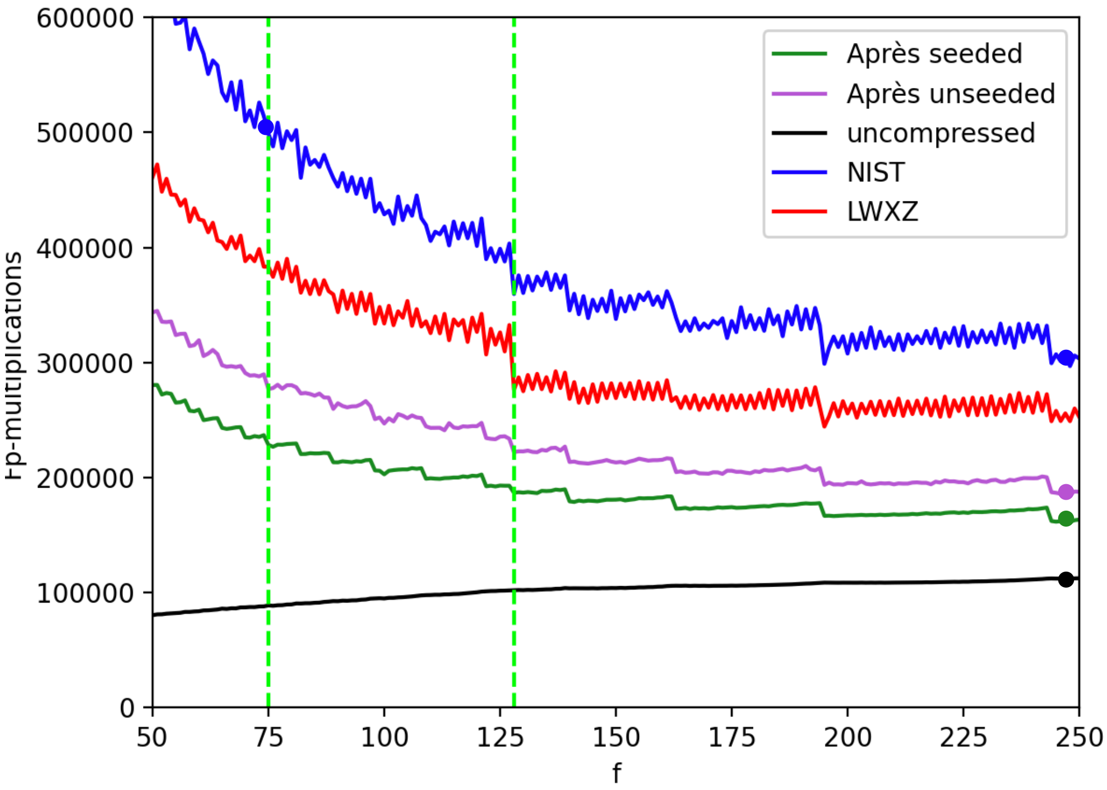


All Results

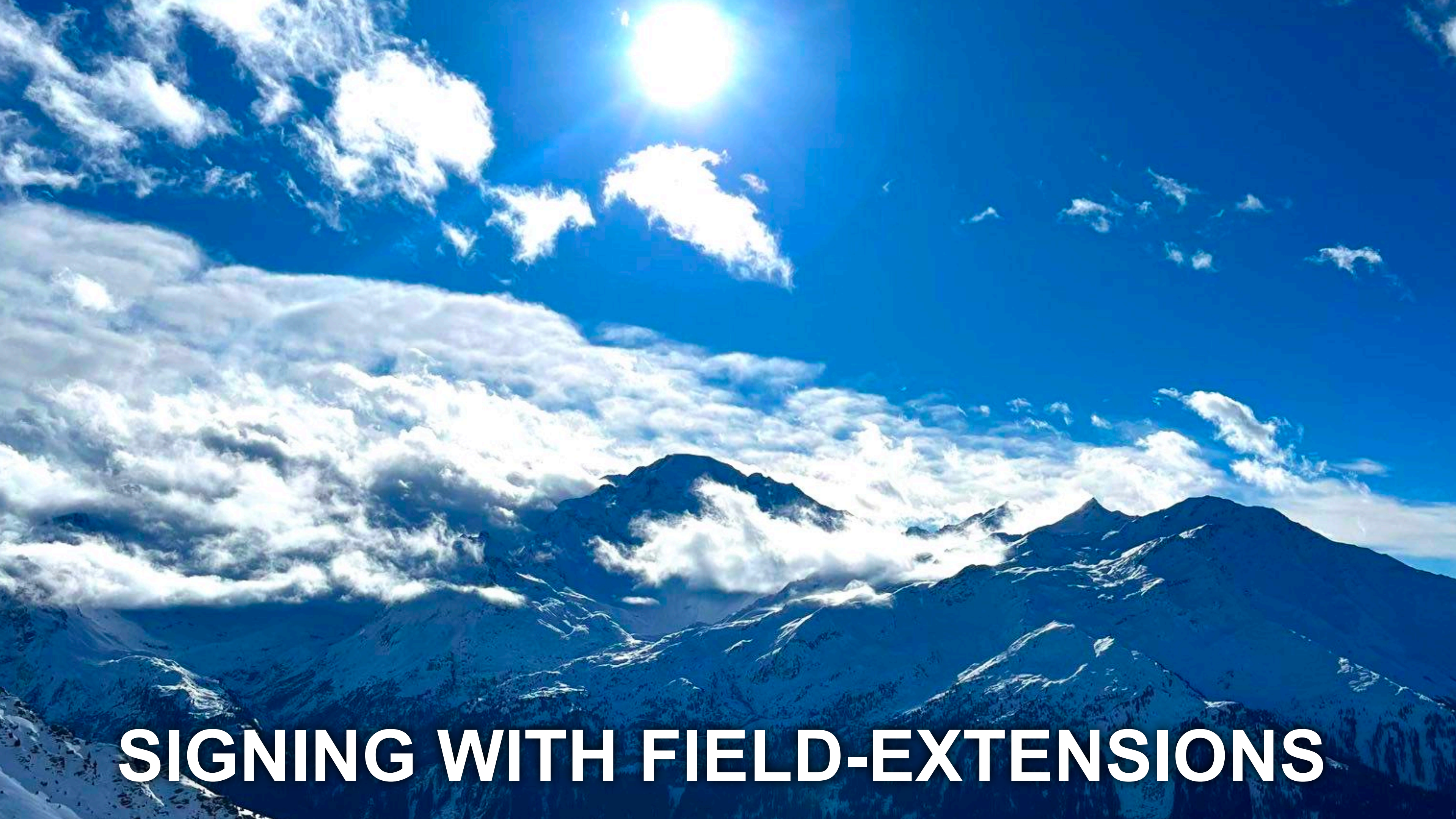


Après seeded
Après unseeded
uncompressed
NIST
Fp-multiplications
600000
500000
400000
300000
200000
100000
0
50
75
100
125
150
175
200
225
250
f

All Results



- Increasing f:
  - 1.68x faster
- Optimised:
  - 2.65x faster
- Seeded (+10 B)
  - 3.04x faster
- Uncomp. (2x B)
  - 4.40x faster

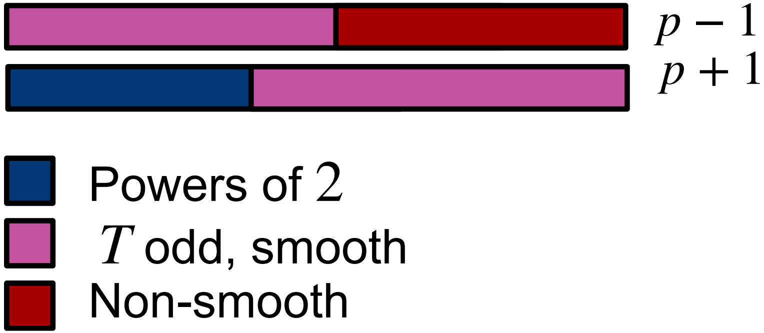


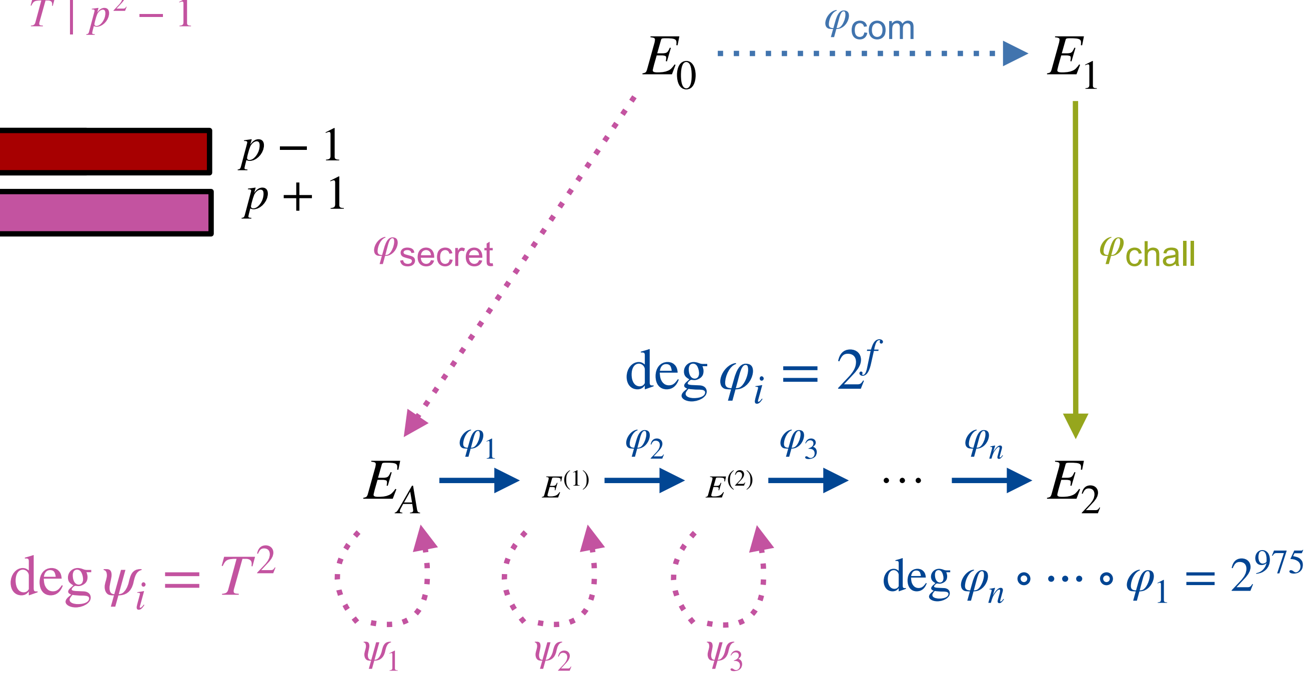
SIGNING WITH FIELD-EXTENSIONS

Recall - Response

Requirement on prime:

$2^f \mid p+1$ $\quad T \mid p^2 - 1$

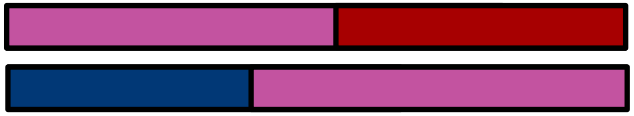




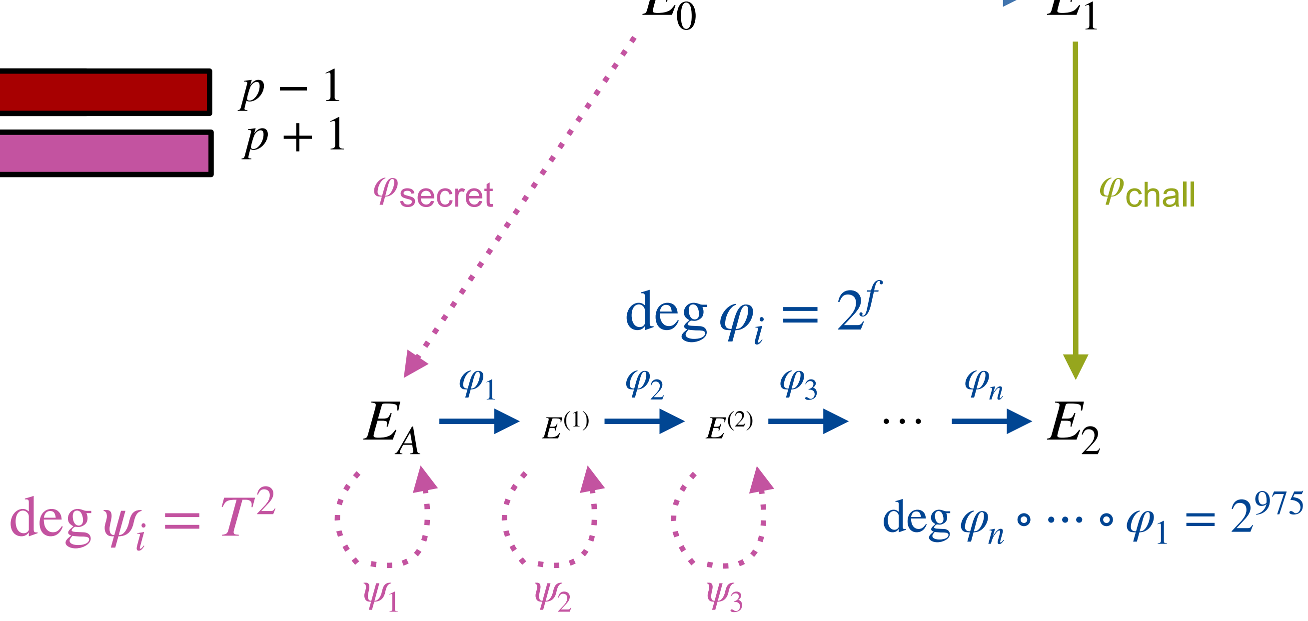
Recall - Response

Requirement on prime:

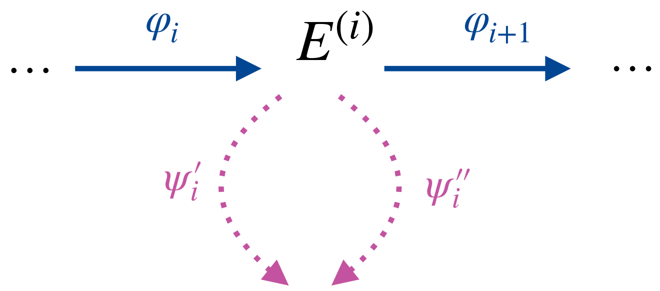
$2^f \mid p+1$ ~~$T \mid p+1$~~



- Powers of 2
- T odd, smooth
- Non-smooth



Computing the T-isogenies



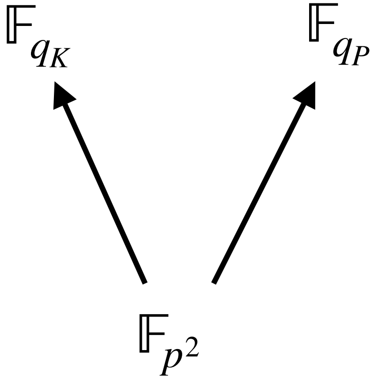
$$\deg \psi_i' = T = \prod \ell_i^{r_i}$$

Supersingular magic ⇒ All isogenies defined over $\mathbb{F}_{p^2}$

⇒ We can work with one prime power at a time

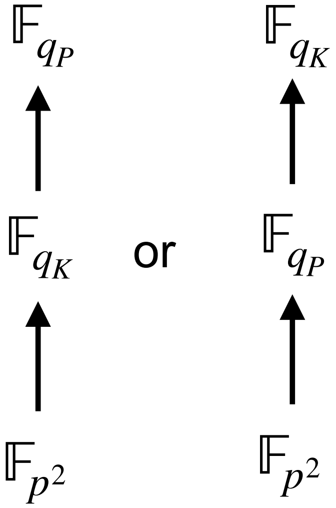
Computing the T-isogenies: Four options

- Compute $\varphi(P)$, φ generated by K
- Let K be defined over $\mathbb{F}_{q_K}$
- Let P be defined over $\mathbb{F}_{q_P}$
- $\deg \varphi = n$
- Case 1: Different field extensions
  - Rational polynomial (Kohel)



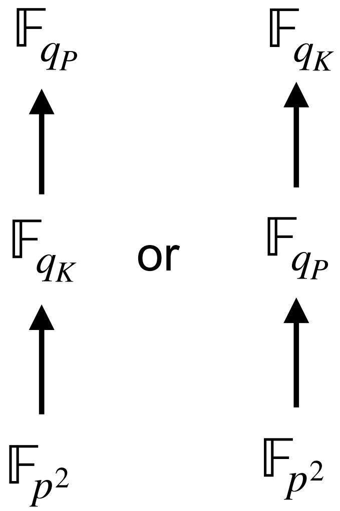
Computing the T-isogenies: Four options

- Compute $\varphi(P)$, φ generated by K
- Let K be defined over $\mathbb{F}_{q_K}$
- Let P be defined over $\mathbb{F}_{q_P}$
- $\deg \varphi = n$
- Case 2: $n < 100$ and containment
  - Velu



Computing the T-isogenies: Four options

- Compute $\varphi(P)$, φ generated by K
- Let K be defined over $\mathbb{F}_{q_K}$
- Let P be defined over $\mathbb{F}_{q_P}$
- $\deg \varphi = n$
- Case 3: $n > 100$ and containment
  - Square-root Velu



Computing the T-isogenies: Four options

- Compute $\varphi(P)$, φ generated by K
- Let K be defined over $\mathbb{F}_{q_K}$
- Let P be defined over $\mathbb{F}_{q_P}$
- $\deg \varphi = n$
- Case 4: Containment and n small
    - Computing norms (F&F)

Example prime

- 7-block verification

$p_7 = 2^{145} \cdot 3^9 \cdot 59^3 \cdot 311^3 \cdot 317^3 \cdot 503^3 - 1.$

T is 997-smooth

- 4-block verification

$p_4 = 2^{242} \cdot 3 \cdot 67 - 1$

T is 2293-smooth

$E(\mathbb{F}_{p^{2k}})$	Torsion group
$k = 1$	$E[3^7], E[53^2], E[59^3], E[61], E[79], E[283], E[311^3]$ $E[317^3], E[349], E[503^2], E[859], E[997]$
$k = 3$	$E[13], E[109], E[223], E[331]$
$k = 4$	$E[17]$
$k = 5$	$E[11], E[31], E[71], E[241], E[271]$
$k = 6$	$E[157]$
$k = 7$	$E[7^2], E[29], E[43], E[239]$
$k = 8$	$E[113]$
$k = 9$	$E[19^2]$
$k = 10$	$E[5^4], E[41]$
$k = 11$	$E[23], E[67]$
$k = 12$	$E[193]$
$k = 13$	$E[131]$
$k = 15$	$E[181]$
$k = 18$	$E[37], E[73]$
$k = 23$	$E[47]$

Sage-Math Implementation

Proof-of-concept signing implemented with Velu, SqrtVelu and Kohel's formulae:

Table 1: Comparison between estimated cost of signing for three different primes.

p	largest $\ell \mid T$	largest $\mathbb{F}_{p^{2k}}$	$\textsc{SigningCost}_p(T)$	Adj. Cost	Timing
p_{1973}	1973	$k = 1$	8371.7	1956.5	11m, 32s
p_7	997	$k = 23$	4137.9	-	9m, 20s
p_4	2293	$k = 53$	9632.7	-	15m, 52s


THANK YOU!
Mountains:
🇳🇴, 🇨🇭 , 🇨🇭 , 🇳🇴, 🇳🇴, 🇨🇭