Finding Practical Parameters for Isogeny-based Cryptography

Maria Corte-Real Santos, Jonathan Komada Eriksen, Michael Meyer, Francisco Rodríguez-Henríquez

University College London

ENS de Lyon, CNRS

Motivation

We are motivated by *post-quantum cryptography* built from *isogenies between elliptic curves defined over* $\mathbb{F}_{p^2}$.

For the efficient computation of isogenies over $\mathbb{F}_{p^2}$, protocols often have special requirements on p.

Motivation

We are motivated by *post-quantum cryptography* built from *isogenies between elliptic curves defined over* $\mathbb{F}_{p^2}$.

For the efficient computation of isogenies over $\mathbb{F}_{p^2}$, protocols often have special requirements on p.

Cryptographic sized primes p such that $p^2 - 1$ is (sufficiently) smooth.

Motivation

We are motivated by *post-quantum cryptography* built from *isogenies between elliptic curves defined over* $\mathbb{F}_{p^2}$.

For the efficient computation of isogenies over $\mathbb{F}_{p^2}$, protocols often have special requirements on p.

Cryptographic sized primes p such that $p^2 - 1$ is (sufficiently) smooth.

Why?

Factors of $p^2 - 1$ correspond to degrees of isogenies computable over $\mathbb{F}_{p^2}$.

Twin Smooth Integers

Definition:

For an integer B, we say that a pair of consecutive integers $(r, r+1)$ are B-smooth twins if their product $r(r+1)$ is B-smooth.

Twin Smooth Integers

Definition:

For an integer B, we say that a pair of consecutive integers $(r, r+1)$ are B-smooth twins if their product $r(r+1)$ is B-smooth.

For example, the following are 100-smooth twins:

$$166055401586083680 = 2^5 \cdot 3^3 \cdot 5 \cdot 11^3 \cdot 23 \cdot 43 \cdot 59 \cdot 67 \cdot 83 \cdot 89$$
$$166055401586083681 = 7^2 \cdot 17^{10} \cdot 41^2$$

Twin Smooth Integers

Definition:

For an integer B, we say that a pair of consecutive integers $(r, r+1)$ are B-smooth twins if their product $r(r+1)$ is B-smooth.

For example, the following are 100-smooth twins:

$$166055401586083680 = 2^5 \cdot 3^3 \cdot 5 \cdot 11^3 \cdot 23 \cdot 43 \cdot 59 \cdot 67 \cdot 83 \cdot 89$$
$$166055401586083681 = 7^2 \cdot 17^{10} \cdot 41^2$$

Remark:

If $p = 2r + 1$ is prime, then $p^2 - 1 = 4r(r+1)$ is B-smooth!

A brief history of twin smooths in cryptography

B-SIDH: supersingular isogeny Diffie-Hellman using twisted torsion

Costello (ASIACRYPT 2020)

A brief history of twin smooths in cryptography

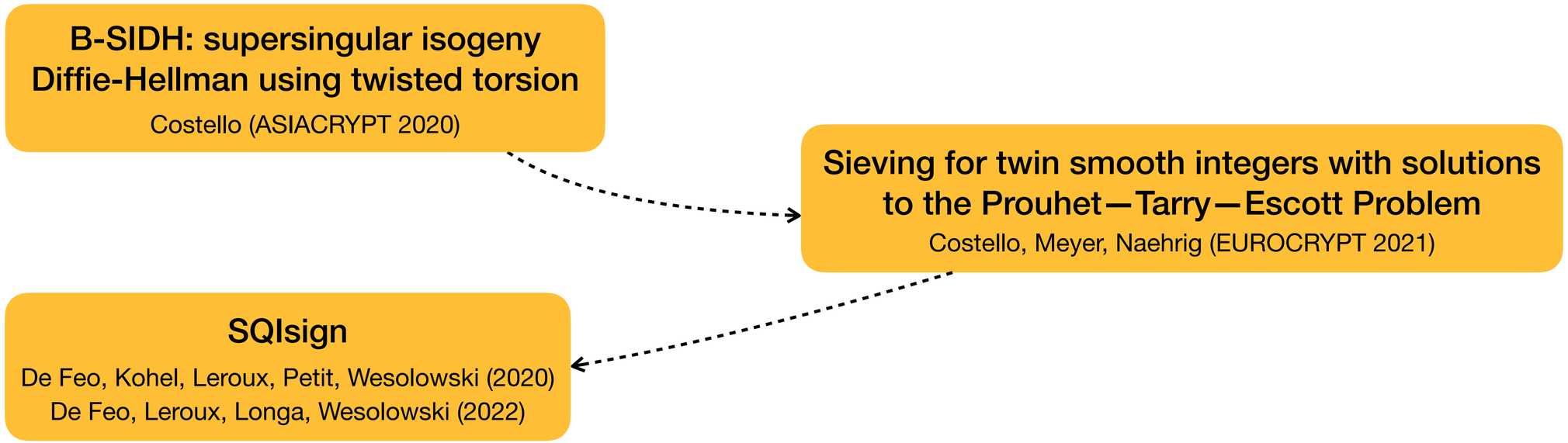
B-SIDH: supersingular isogeny Diffie-Hellman using twisted torsion

Costello (ASIACRYPT 2020)

Sieving for twin smooth integers with solutions to the Prouhet—Tarry—Escott Problem

Costello, Meyer, Naehrig (EUROCRYPT 2021)

A brief history of twin smooths in cryptography



Relaxing the problem

In SQIsign, we don't need $p^2 - 1$ to be *fully* smooth.

Requirements: $\log(p) \approx 256, 384, 512, \quad p \equiv 3 \bmod 4$

$2^f T \mid p^2 - 1$ is a B-smooth cofactor

with $T \approx p^{5/4+\epsilon}$ odd and f is large.

We also need a large power of 3: $3^{f'} \mid p + 1$ such that $2^f 3^{f'} \geq 2^\lambda$

Other more recent schemes also have this more relaxed requirement (AprèsSQI, POKE, …).

Relaxing the problem

In SQIsign, we don't need $p^2 - 1$ to be *fully* smooth.

Requirements: $\log(p) \approx 256, 384, 512, \quad p \equiv 3 \bmod 4$

$2^f T \mid p^2 - 1$ is a B-smooth cofactor

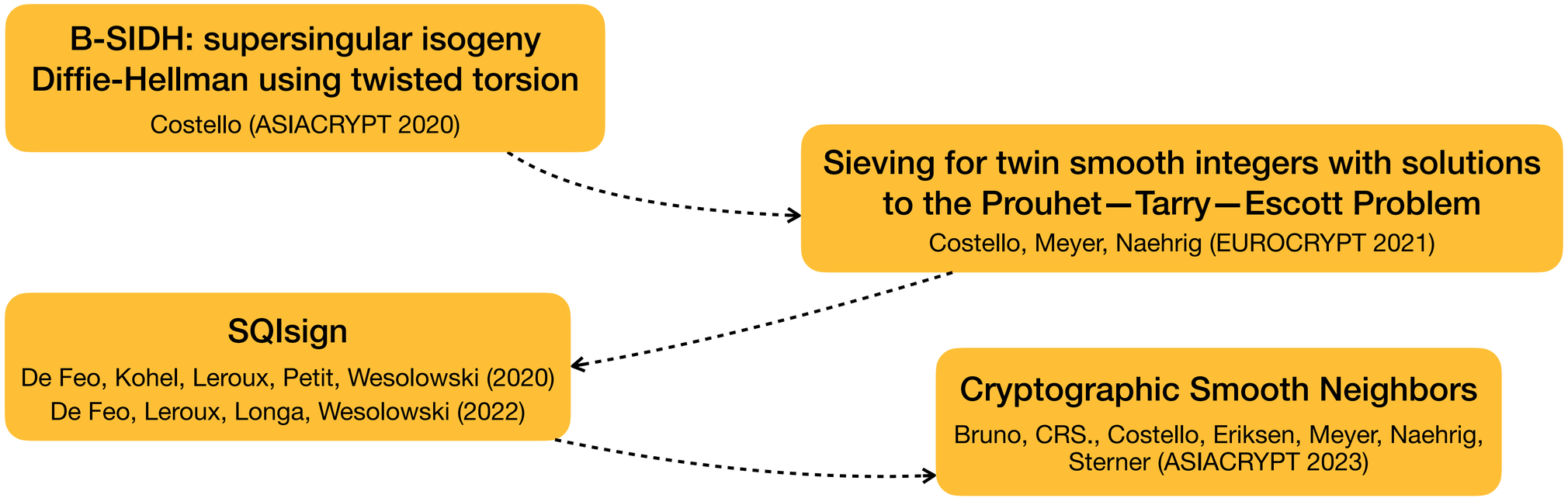
signing

with $T \approx p^{5/4+\epsilon}$ and f is large. ← verification

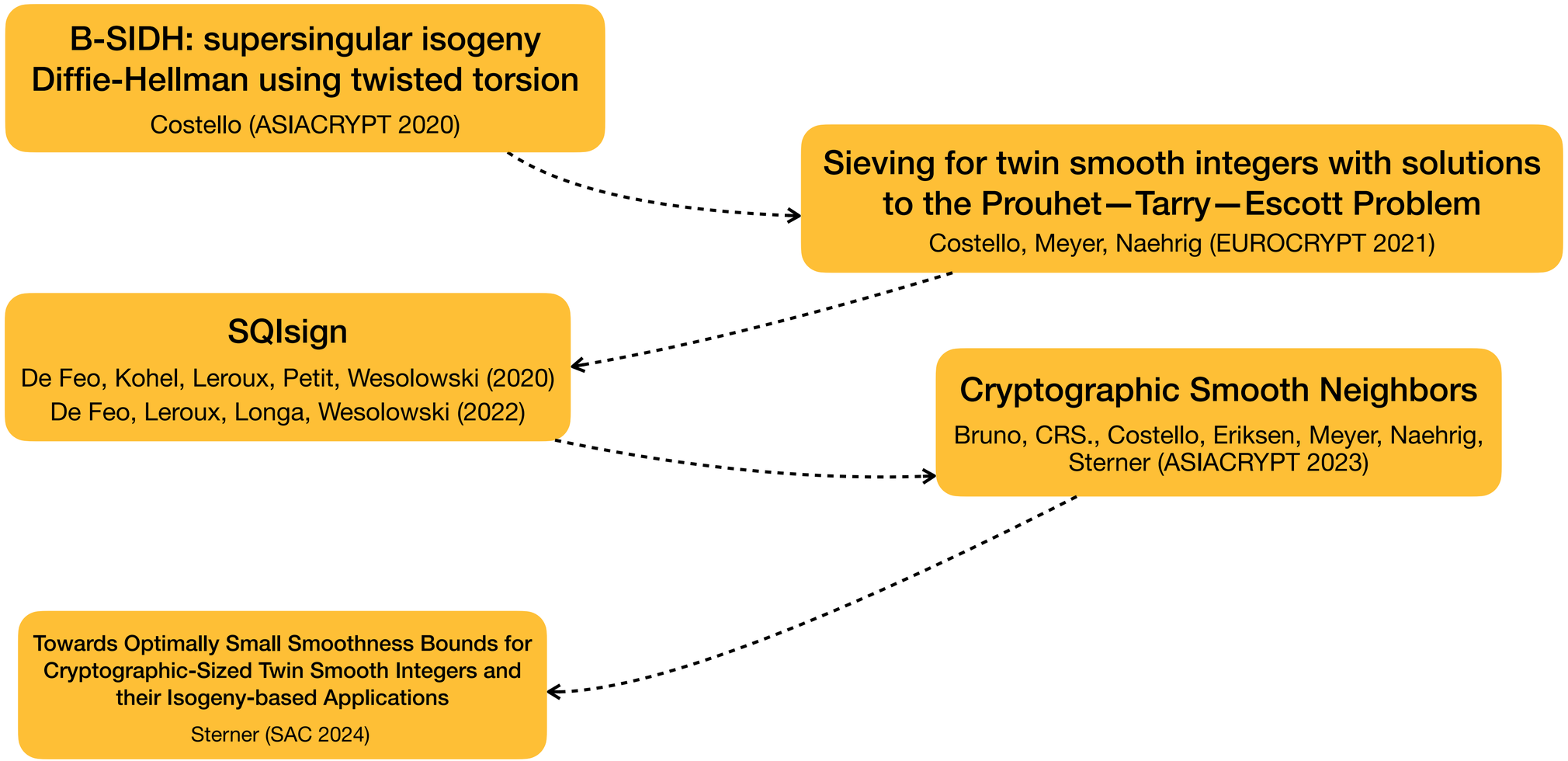
We also need a large power of 3: $3^{f'} \mid p + 1$ such that $2^f 3^{f'} \geq 2^\lambda$ ← challenge

Other more recent schemes also have this more relaxed requirement (AprèsSQI, POKE, …).

A brief history of twin smooths in cryptography



A brief history of twin smooths in cryptography



Landscape for finding twins

Constructive

Pell equations

The CHM algorithm

Probabilistic

XGCD/CRT

Cyclotomic factors

Sieving + PTE Solutions

XGCD over $\mathbb{Q}[x]$

Landscape for finding twins

Constructive

Pell equations

The CHM algorithm

Probabilistic

XGCD/CRT

Cyclotomic factors

Sieving + PTE Solutions

XGCD over $\mathbb{Q}[x]$

A deeper look at constructive methods

Pell equations

- $(r, r+1)$ a B-smooth twin
- $x := 2r + 1$, so that $x^2 - 1$ is B-smooth
- Let $x^2 - 1 = Dy^2$ where D, y are B-smooth and D is square free
- Then (x, y) is a solution to the Pell conic

$$X^2 - DY^2 = 1$$

A deeper look at constructive methods

Pell equations

- $(r, r+1)$ a B-smooth twin
- $x := 2r + 1$, so that $x^2 - 1$ is B-smooth
- Let $x^2 - 1 = Dy^2$ where D, y are B-smooth and D is square free
- Then (x, y) is a solution to the Pell conic

$$X^2 - DY^2 = 1$$

So, solving all $2^{\pi(B)}$ Pell equations will find the complete set of B-smooth twins.

A deeper look at constructive methods

Pell equations

- $(r, r+1)$ a B-smooth twin
- $x := 2r + 1$, so that $x^2 - 1$ is B-smooth
- Let $x^2 - 1 = Dy^2$ where D, y are B-smooth and D is square free
- Then (x, y) is a solution to the Pell conic

$$X^2 - DY^2 = 1$$

So, solving all $2^{\pi(B)}$ Pell equations will find the complete set of B-smooth twins.

Example: with $B = 113$ the largest twins $(r, r+1)$ found by solving all 2^{30} Pell equations have

$$r = 193161583770739238340000 \approx 2^{75}$$

A deeper look at constructive methods

Conrey, Holmstrom and McLaughlin (2012) algorithm finds almost all B-smooth twins:

- Let $S^{(0)} = \{1,2,\ldots B-1\}$ representing B-smooth twins $(1,2), (2,3), \ldots, (B-1,B)$
- For each $r, s \in S^{(0)}$ with $r < s$ compute

$$\frac{t}{t'} = \frac{r}{r+1} \cdot \frac{s+1}{s} \text{ with } \gcd(t, t') = 1$$

- Let $S^{(1)} = S^{(0)} \cup \{\text{new solutions } t \mid t' = t+1\}$
- Repeat with $S^{(1)}$ instead of $S^{(0)}$
- Terminate when $S^{(d+1)} = S^{(d)}$ for some d

A deeper look at constructive methods

Conrey, Holmstrom and McLaughlin (2012) algorithm finds almost all B-smooth twins:

Example: Let's illustrate for $B = 5$. Starting set is $S^{(0)} = \{1,2,3,4\}$.

We have that

$$\frac{2}{2+1} \cdot \frac{3+1}{3} = \frac{8}{9}, \frac{2}{2+1} \cdot \frac{4+1}{4} = \frac{5}{6}, \frac{3}{3+1} \cdot \frac{4+1}{4} = \frac{15}{16}$$

So we add $5,8,$ and 15 to get

$$S^{(1)} = \{1,2,3,4,5,8,15\}.$$

After second and third CHM iterations we get

$$S^{(2)} = \{1,2,3,4,5,8,15,24\} \text{ and } S^{(3)} = \{1,2,3,4,5,8,15,24,80\}$$

We get $S^{(3)} = S^{(4)}$ so we terminate. Indeed this is the full set of 5-smooth integers.

A deeper look at constructive methods

Conrey, Holmstrom and McLaughlin (2012) algorithm finds almost all B-smooth twins:

- Luca and Najman (2011) computed the full set of 13,374 twin smooths for $B = 100$ in 15 days by solving 2^{25} Pell equations.
- Running the CHM algorithm we find 13,333 of these twins in 20 minutes.

A deeper look at constructive methods

Conrey, Holmstrom and McLaughlin (2012) algorithm finds almost all B-smooth twins:

- Luca and Najman (2011) computed the full set of 13,374 twin smooths for $B = 100$ in 15 days by solving 2^{25} Pell equations.
- Running the CHM algorithm we find 13,333 of these twins in 20 minutes.
- The largest run for the Pell equation approach was for $B = 113$.

A deeper look at constructive methods

Conrey, Holmstrom and McLaughlin (2012) algorithm finds almost all B-smooth twins:

- Luca and Najman (2011) computed the full set of 13,374 twin smooths for $B = 100$ in 15 days by solving 2^{25} Pell equations.
- Running the CHM algorithm we find 13,333 of these twins in 20 minutes.
- The largest run for the Pell equation approach was for $B = 113$.
- CHM ran their algorithm for $B = 200$ in about 2 weeks to find 346,192 twin smooths, the largest of which were the 79-bit integers:

$$589864439608716991201560 = 2^3 \cdot 3^3 \cdot 5 \cdot 7^2 \cdot 11^2 \cdot 17 \cdot 31 \cdot 59^2 \cdot 83 \cdot 139^2 \cdot 173 \cdot 181$$

$$589864439608716991201561 = 13^2 \cdot 113^2 \cdot 127^2 \cdot 137^2 \cdot 151^2 \cdot 199^2$$

A deeper look at constructive methods

Conrey, Holmstrom and McLaughlin (2012) algorithm

[BCC+23] optimised the CHM algorithm and ran it for $B = 547$ to get 82,026,426 twin pairs, the largest of which are the 122-bit pair

$$r = 5^4 \cdot 7 \cdot 13^2 \cdot 17^2 \cdot 19 \cdot 29 \cdot 41 \cdot 109 \cdot 163 \cdot 173 \cdot 239 \cdot 241^2 \cdot 271 \cdot 283 \cdot 499 \cdot 509$$

$$r + 1 = 2^8 \cdot 3^2 \cdot 31^2 \cdot 43^2 \cdot 47^2 \cdot 83^2 \cdot 103^2 \cdot 311^2 \cdot 479^2 \cdot 523^2$$

A deeper look at probabilistic methods

XGCD method

To generate a pair $(r, r+1)$:

1. Choose two B-smooth numbers $a, b \approx 2^{\lambda}$ with $\gcd(a, b) = 1$.
2. Run XGCD algorithm to find integers s, t with $|s| < |b/2|$, $|t| < |a/2|$ such that

$$as + bt = 1.$$

3. If s, t have large enough smooth factors, we set $(r, r+1) = (|as|, |bt|)$ where $r \approx 2^{2\lambda}$.

A deeper look at probabilistic methods

XGCD method

To generate a pair $(r, r+1)$:

1. Choose two B-smooth numbers $a, b \approx 2^\lambda$ with $\gcd(a, b) = 1$.
2. Run XGCD algorithm to find integers s, t with $|s| < |b/2|$, $|t| < |a/2|$ such that

$$as + bt = 1.$$

3. If s, t have large enough smooth factors, we set $(r, r+1) = (|as|, |bt|)$ where $r \approx 2^{2\lambda}$.

Why does this work? The product of two numbers $s \cdot t$ of 2 numbers of size $\approx 2^\lambda$ is much more likely to be smooth than a random integer of size $\approx 2^{2\lambda}$.

Dickman—de Bruijn function

A deeper look at probabilistic methods

Cyclotomic factors

Costello (2019) noticed that a lot of twins found with Pell equations were of the form $(x^2 - 1, x^2)$.

⟶ Let's consider twins of the form $(x^n - 1, x^n)$ for even n.

Why does this work? The polynomial $x^n - 1$ splits into cyclotomic factors.

A deeper look at probabilistic methods

Cyclotomic factors

Costello (2019) noticed that a lot of twins found with Pell equations were of the form $(x^2 - 1, x^2)$.

⟶ Let's consider twins of the form $(x^n - 1, x^n)$ for even n.

Why does this work? The polynomial $x^n - 1$ splits into cyclotomic factors.

Example: $x^4 - 1 = (x - 1)(x + 1)(x^2 + 1)$

So we just need to find an integer ℓ such that

$\ell - 1, \ell + 1, \ell^2 + 1$, are all smooth

A deeper look at probabilistic methods

Cyclotomic factors

Costello (2019) noticed that a lot of twins found with Pell equations were of the form $(x^2 - 1, x^2)$.

⟶ Let's consider twins of the form $(x^n - 1, x^n)$ for even n.

Why does this work? The polynomial $x^n - 1$ splits into cyclotomic factors.

Example: $x^4 - 1 = (x - 1)(x + 1)(x^2 + 1)$

So we just need to find an integer ℓ such that

$\ell - 1, \ell + 1, \ell^2 + 1$, are all smooth

The larger the degree of the factors, the larger the smoothness bound

⟶ **PTE solutions** [CMN21]

The boosting method

These methods don’t scale well to higher security levels. In [BCC+23] we introduce a method to use cyclotomic factors to *boost* twin smooths.

The boosting method

These methods don't scale well to higher security levels. In [BCC+23] we introduce a method to use cyclotomic factors to *boost* twin smooths.

Example: Let $(r, r+1)$ be b-bit smooth twins.

Letting $x = r + 1$, then $x - 1 = r$ is smooth too. Setting $p = 2x^n - 1$ we have

$$p^2 - 1 = 4x^n(x^n - 1) \text{ and } x - 1 \mid x^n - 1 \text{ for even } n$$

$\longrightarrow$ $((n+1)b + 2)$-bits of guaranteed smoothness

The boosting method

These methods don't scale well to higher security levels. In [BCC+23] we introduce a method to use cyclotomic factors to *boost* twin smooths.

Example: Let $(r, r+1)$ be b-bit smooth twins.

Letting $x = r + 1$, then $x - 1 = r$ is smooth too. Setting $p = 2x^n - 1$ we have

$$p^2 - 1 = 4x^n(x^n - 1) \text{ and } x - 1 \mid x^n - 1 \text{ for even } n$$

$\longrightarrow$ $((n+1)b + 2)$-bits of guaranteed smoothness

Take $b = \dfrac{\log(p) - 1}{n}$ and we have $\dfrac{n+1}{n}(\log(p) - 1) + 2$ bits of guaranteed smoothness.

The boosting method

These methods don't scale well to higher security levels. In [BCC+23] we introduce a method to use cyclotomic factors to *boost* twin smooths.

Example: Let $(r, r+1)$ be b-bit smooth twins.

Letting $x = r + 1$, then $x - 1 = r$ is smooth too. Setting $p = 2x^n - 1$ we have

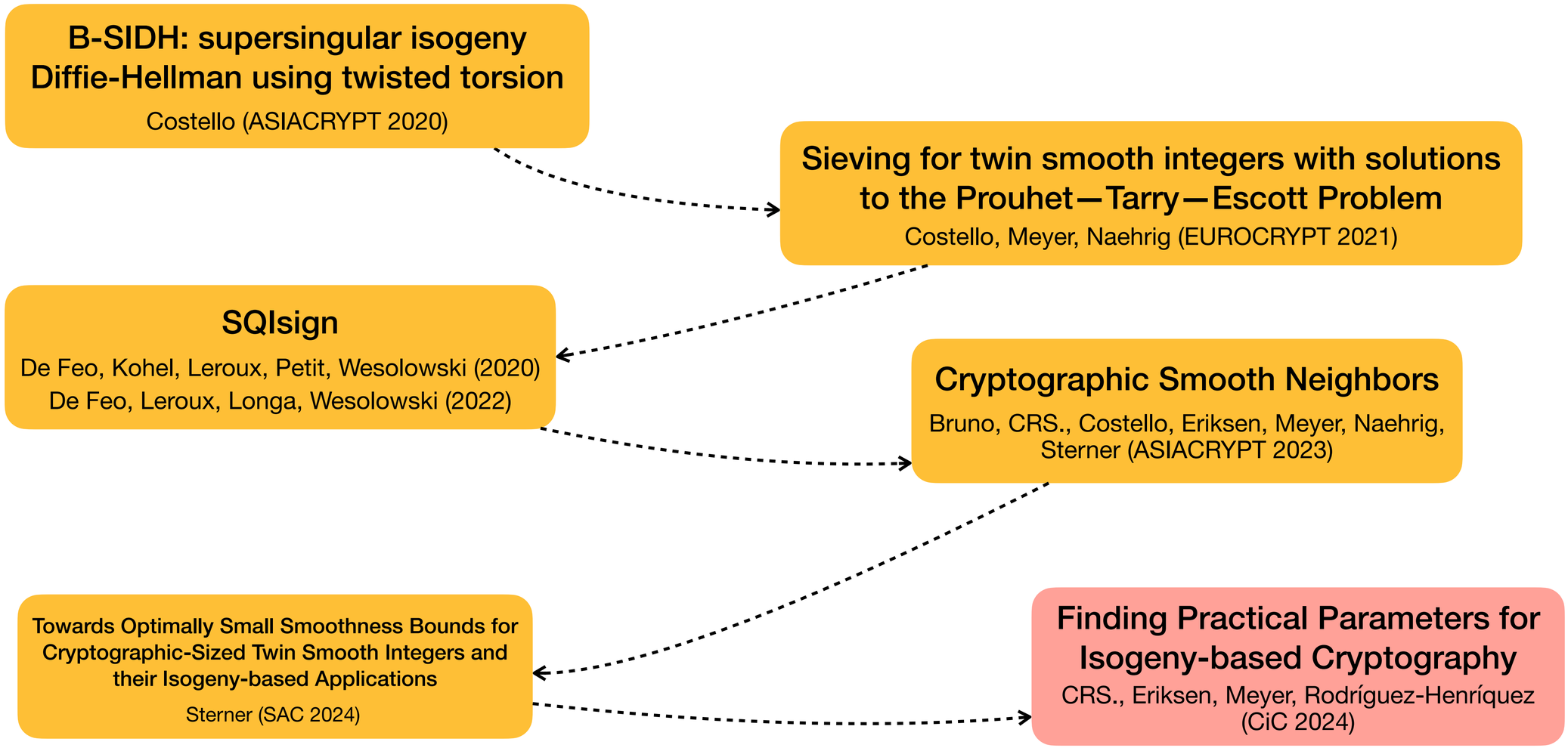
$$p^2 - 1 = 4x^n(x^n - 1) \text{ and } x - 1 \mid x^n - 1 \text{ for even } n$$

$\longrightarrow$ $((n+1)b + 2)$-bits of guaranteed smoothness

Take $b = \dfrac{\log(p) - 1}{n}$ and we have $\dfrac{n+1}{n}(\log(p) - 1) + 2$ bits of guaranteed smoothness.

In [BCC+23], CHM twins were used as inputs (for optimal smoothness bounds). In this work, we use different inputs.

A brief history of twin smooths in cryptography



Combining methods

XGCD

Sieving

Boosting

Then we score primes against a cost metric.

Parameters for SQIsign: NIST-I

Previous work $p = p_{3923}$ with $\log(p) = 253.7$ and [DLLW23]

$$f = 65$$

$$T = 3^{65} \cdot 5^2 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 29^2 \cdot 37^2 \cdot 43 \cdot 47 \cdot 79 \cdot 157 \cdot 197 \cdot 239 \cdot 263 \cdot 271 \cdot 281 \cdot 283 \cdot 307 \cdot 461 \cdot 521 \cdot 563 \cdot 599 \cdot 607 \cdot 619 \cdot 743 \cdot 827 \cdot 941 \cdot 2357 \cdot 3923$$

Cost: 2.16

Parameters for SQIsign: NIST-I

Previous work $p = p_{3923}$ with $\log(p) = 253.7$ and [DLLW23]

$$f = 65$$

$$T = 3^{65} \cdot 5^2 \cdot 7 \cdot 11 \cdot 13 \cdot 17 \cdot 19 \cdot 29^2 \cdot 37^2 \cdot 43 \cdot 47 \cdot 79 \cdot 157 \cdot 197 \cdot 239 \cdot 263 \cdot 271 \cdot 281 \cdot 283 \cdot 307 \cdot 461 \cdot 521 \cdot 563 \cdot 599 \cdot 607 \cdot 619 \cdot 743 \cdot 827 \cdot 941 \cdot 2357 \cdot 3923$$

Cost: 2.16

This work $p = p^I_{1973}$ with $\log(p) = 253.7$ and

$$f = 75$$

$$T = 3^{36} \cdot 7^4 \cdot 11 \cdot 13 \cdot 23^2 \cdot 37 \cdot 59^2 \cdot 89 \cdot 97 \cdot 101^2 \cdot 107 \cdot 109^2 \cdot 131 \cdot 137 \cdot 197^2 \cdot 223 \cdot 239 \cdot 383 \cdot 389 \cdot 491^2 \cdot 499 \cdot 607 \cdot 743^2 \cdot 1033 \cdot 1049 \cdot 1193 \cdot 1913^2 \cdot 1973$$

Cost: 2.01 + larger power-of-2 for verification

Parameters for SQIsign: NIST-III

Previous work $p = p_{10243}$ with $\log(p) = 381.2$ and [BCC+23]

$$f = 79$$

$$T = 3^6 \cdot 5^2 \cdot 7 \cdot 11 \cdot 17 \cdot 19 \cdot 23^{12} \cdot 47 \cdot 71 \cdot 79 \cdot 107^6 \cdot 109 \cdot 127^6 \cdot 149 \cdot 229 \cdot 269 \cdot 283 \cdot 307^6 \cdot 349 \cdot 401^6 \cdot 449 \cdot 463 \cdot 547^6 \cdot 1019 \cdot 1033 \cdot 1657 \cdot 2179 \cdot 2293 \cdot 4099 \cdot 5119 \cdot 10243$$

Cost: 6.08

Parameters for SQIsign: NIST-III

Previous work $p = p_{10243}$ with $\log(p) = 381.2$ and [BCC+23]

$$f = 79$$

$$T = 3^6 \cdot 5^2 \cdot 7 \cdot 11 \cdot 17 \cdot 19 \cdot 23^{12} \cdot 47 \cdot 71 \cdot 79 \cdot 107^6 \cdot 109 \cdot 127^6 \cdot 149 \cdot 229 \cdot 269 \cdot 283 \cdot 307^6 \cdot 349 \cdot 401^6 \cdot 449 \cdot 463 \cdot 547^6 \cdot 1019 \cdot 1033 \cdot 1657 \cdot 2179 \cdot 2293 \cdot 4099 \cdot 5119 \cdot 10243$$

Cost: 6.08

This work $p = p^{III}_{47441}$ with $\log(p) = 377.9$ and

$$f = 97$$

$$T = 3^{68} \cdot 5 \cdot 7^{12} \cdot 11^4 \cdot 13 \cdot 47^4 \cdot 89 \cdot 113 \cdot 157^4 \cdot 173 \cdot 233 \cdot 239 \cdot 241 \cdot 443 \cdot 509^4 \cdot 569 \cdot 761^4 \cdot 1229 \cdot 2393 \cdot 3371 \cdot 4517 \cdot 5147 \cdot 5693 \cdot 5813 \cdot 9397 \cdot 26777 \cdot 39679 \cdot 47441$$

Cost: 5.67 + larger power-of-2 for verification

Parameters for SQIsign: NIST-V

Previous work $p = p_{150151}$ with $\log(p) = 507.8$ and [BCC+23]

$$f = 85$$

$$T = 3^2 \cdot 5 \cdot 7 \cdot 11^2 \cdot 17^{12} \cdot 23^2 \cdot 29 \cdot 37^6 \cdot 59^6 \cdot 97^6 \cdot 127 \cdot 163 \cdot 173 \cdot 191 \cdot 193 \cdot 211 \cdot 233^6 \cdot 277 \cdot 311^{12} \cdot 911^6 \cdot 347 \cdot 617 \cdot 661 \cdot 761 \cdot 1039 \cdot 1297^6 \cdot 4637 \cdot 5821 \cdot 15649 \cdot 19139 \cdot 143443 \cdot 150151$$

Cost: 17.05

Parameters for SQIsign: NIST-V

Previous work $p = p_{150151}$ with $\log(p) = 507.8$ and [BCC+23]

$$f = 85$$

$$T = 3^2 \cdot 5 \cdot 7 \cdot 11^2 \cdot 17^{12} \cdot 23^2 \cdot 29 \cdot 37^6 \cdot 59^6 \cdot 97^6 \cdot 127 \cdot 163 \cdot 173 \cdot 191 \cdot 193 \cdot 211 \cdot 233^6 \cdot 277 \cdot 311^{12} \cdot 911^6 \cdot 347 \cdot 617 \cdot 661 \cdot 761 \cdot 1039 \cdot 1297^6 \cdot 4637 \cdot 5821 \cdot 15649 \cdot 19139 \cdot 143443 \cdot 150151$$

Cost: 17.05

This work $p = p^V_{318233}$ with $\log(p) = 501.2$ and

$$f = 145$$

$$T = 3^{72} \cdot 5 \cdot 7 \cdot 13^6 \cdot 17 \cdot 37 \cdot 41^6 \cdot 53 \cdot 67^6 \cdot 73 \cdot 103^6 \cdot 127 \cdot 151 \cdot 461^6 \cdot 643 \cdot 733 \cdot 739 \cdot 827^6 \cdot 1009 \cdot 2539 \cdot 4153 \cdot 5059 \cdot 7127 \cdot 10597 \cdot 13591 \cdot 14923 \cdot 15541 \cdot 15991 \cdot 18583 \cdot 23227 \cdot 48187 \cdot 63247 \cdot 65521 \cdot 318233$$

Cost: 12.40 + larger power-of-2 for verification

Finding parameters for POKE (PKE scheme)

Old Requirements: [BM25]

Uses prime $p = 2^a 3^b c - 1$.
For security, we need $2^a \approx 2^\lambda, 3^b \approx 2^{2\lambda}$ $\longrightarrow$ $p \approx 2^{3\lambda}$

Finding parameters for POKE (PKE scheme)

Old Requirements: [BM25]

Uses prime $p = 2^a 3^b c - 1$.
For security, we need $2^a \approx 2^\lambda, 3^b \approx 2^{2\lambda}$ $\longrightarrow$ $p \approx 2^{3\lambda}$

New requirements (this work):

Uses prime p such that $2^a T \mid p^2 - 1$ $\longrightarrow$ $p \approx 2^{2\lambda}$

Finding parameters for POKE (PKE scheme)

Old Requirements: [BM25]

Uses prime $p = 2^a 3^b c - 1$.
For security, we need $2^a \approx 2^\lambda, 3^b \approx 2^{2\lambda}$ $\longrightarrow$ $p \approx 2^{3\lambda}$

New requirements (this work):

Uses prime p such that $2^a T \mid p^2 - 1$ $\longrightarrow$ $p \approx 2^{2\lambda}$

Hope: efficiency benefits of using a smaller prime outweighs the performance penalty incurred by employing smooth T-torsion compared to 3^b-torsion.

Finding parameters for POKE (PKE scheme)

Old Requirements: [BM25]

Uses prime $p = 2^a 3^b c - 1$.
For security, we need $2^a \approx 2^\lambda, 3^b \approx 2^{2\lambda}$ $\longrightarrow$ $p \approx 2^{3\lambda}$

New requirements (this work):

Uses prime p such that $2^a T \mid p^2 - 1$ $\longrightarrow$ $p \approx 2^{2\lambda}$

We can run the same searches as SQIsign, except with different parameters.

Parameters for POKE: NIST-I

This work $p = p_{1879}$ with $\log(p) = 251.3$ and $2^f T \mid (p^2 - 1)/2$ with

$$f = 129$$

$$T = 3^{10} \cdot 5 \cdot 7 \cdot 11^2 \cdot 13^2 \cdot 17 \cdot 19 \cdot 43^2 \cdot 47^2 \cdot 73^2 \cdot 79 \cdot 139^2 \cdot 233^2 \cdot 263 \cdot 317^2 \cdot 383^2 \cdot 401^2 \cdot 443^2 \cdot 599 \cdot 643 \cdot 1231 \cdot 1301 \cdot 1549 \cdot 1879$$

Concluding thoughts

- Many isogeny-based schemes naturally require primes with a *special shape.*
- New protocols show that primes of this special shape are *still relevant* despite new HD-techniques.
- We present two methods for finding such primes that produce the best parameters in the literature.

eprint 2024/1150

Extra Slides

XGCD + Boost

First outlined in [DLLW23] (for $n = 4$) and extended in this work.

High-level idea:

Do the boosting method as described in [BCC+23], but with twin smooths $(r, r \pm 1)$ of size $\approx 2^{2\lambda/n}$ generated using the XGCD approach.

XGCD + Boost

First outlined in [DLLW23] (for $n = 4$) and extended in this work.

High-level idea:

Do the boosting method as described in [BCC+23], but with twin smooths $(r, r \pm 1)$ of size $\approx 2^{2\lambda/n}$ generated using the XGCD approach.

Fixing n, f_n, f'_n , smoothness bound B and parameter M to control the search space.

XGCD. Set $a = 2^{f_n} 3^{f'_n} \approx 2^{\lambda/n}$ $\quad b = \ell_s^{k_i} \prod \ell_j$

where $\ell_j \notin \{2,3\}$ are M B-smooth primes sampled at random, and ℓ_s is a small prime with k_i chosen so $b \approx 2^{\lambda/n}$. Compute $(r, r \pm 1) = (|sa|, |tb|)$ with the XGCD method.

XGCD + Boost

First outlined in [DLLW23] (for $n = 4$) and extended in this work.

High-level idea:

Do the boosting method as described in [BCC+23], but with twin smooths $(r, r \pm 1)$ of size $\approx 2^{2\lambda/n}$ generated using the XGCD approach.

Fixing n, f_n, f'_n , smoothness bound B and parameter M to control the search space.

XGCD. Set $a = 2^{f_n} 3^{f'_n} \approx 2^{\lambda/n}$ $\quad b = \ell_s^{k_i} \prod \ell_j$

where $\ell_j \notin \{2,3\}$ are M B-smooth primes sampled at random, and ℓ_s is a small prime with k_i chosen so $b \approx 2^{\lambda/n}$. Compute $(r, r \pm 1) = (|sa|, |tb|)$ with the XGCD method.

Boosting. For each pair $(r_i, r_i \pm 1)$ compute $p_i = 2(r_i)^n - 1$. For all p_i, determine if $p_i^2 - 1$ has a sufficiently large B-smooth factor.

Sieve + Boost

High-level idea:

Search for primes of the form $p = 2(2^{f_n}3^{f'_n}x)^n - 1$ for a smooth number x and f_n, f'_n depending on the prime requirements.

Sieve + Boost

High-level idea:

Search for primes of the form $p = 2(2^{f_n}3^{f'_n}x)^n - 1$ for a smooth number x and f_n, f'_n depending on the prime requirements.

Fixing n, f_n, f'_n and smoothness bound B

Sieving. Find all B-smooth numbers x in some suitable range $[L, R]$.

Sieve + Boost

High-level idea:

Search for primes of the form $p = 2(2^{f_n}3^{f'_n}x)^n - 1$ for a smooth number x and f_n, f'_n depending on the prime requirements.

Fixing n, f_n, f'_n and smoothness bound B

Sieving. Find all B-smooth numbers x in some suitable range $[L, R]$.

Boosting. For all smooth $x_i \in [L, R]$, compute $p_i = 2(2^{f_n}3^{f'_n}x_i)^n - 1$. For all p_i, determine if $p_i^2 - 1$ has a sufficiently large B-smooth factor.