Cryptographic Smooth Neighbors

Bruno Sterner, joint work with Giacomo Bruno, Maria Corte-Real Santos, Craig Costello, Jonathan Komada Eriksen, Michael Meyer & Michael Naehrig

Surrey Centre for Cyber Security, University of Surrey, UK

Talk for the GRACE seminar at École Polytechnique
Meet-in-the-Middle
Meet-in-the-Middle
Meat-in-the-Middle
Motivation

Cryptographic sized primes $p$ such that $p \pm 1$ are smooth or contain a large smooth cofactor $\phi$.

The current state-of-the-art in SQISign uses the following 254-bit prime $p$:

$$p = \text{0x348757EADF5C9530B7311A6363F03DB535805FA6E9E48B1FFFFFFFFFFFFFFFF}.$$ 

$\phi$: 

- $p + 1 = 2^{65} \cdot 5^2 \cdot 7 \cdot 11 \cdot 19 \cdot 29^2 \cdot 37^2 \cdot 47 \cdot 197 \cdot 263 \cdot 281 \cdot 461 \cdot 3923 \cdot 62731 \cdot 96362257 \cdot 3924006112952623$,
- $p - 1 = 2 \cdot 3^{65} \cdot 13 \cdot 17 \cdot 43 \cdot 79 \cdot 157 \cdot 239 \cdot 271 \cdot 283 \cdot 307 \cdot 563 \cdot 599 \cdot 607 \cdot 619 \cdot 743 \cdot 827 \cdot 941 \cdot 2357 \cdot 10069.$
Cryptographic sized primes $p$ such that $p \pm 1$ are smooth\(^1\) or contain a large smooth cofactor

\(^1\)A number $n$ is $B$-smooth if all the prime factors of $n$ are at most $B$
Motivation

Cryptographic sized primes $p$ such that $p \pm 1$ are smooth\(^1\) or contain a large smooth cofactor

\[
\begin{align*}
\text{B-SIDH} & \quad \phi : E \to E' \quad \text{SQISign}
\end{align*}
\]

\(^1\)A number $n$ is $B$-smooth if all the prime factors of $n$ are at most $B$
Motivation

Cryptographic sized primes $p$ such that $p \pm 1$ are smooth\(^1\) or contain a large smooth cofactor

\[ B\text{-SIDH} \quad \phi : E \to E' \quad SQISign \]

The current state-of-the-art in SQISign uses the following prime

254-bit prime $p = \text{0x348757EADF5C9530B7311A63633F03DB535805FA6E9E48B1FFFFFFFF}$$\ldots$

\begin{align*}
    p + 1 &= 2^{65} \cdot 5^2 \cdot 7 \cdot 11 \cdot 19 \cdot 29^2 \cdot 37^2 \cdot 47 \cdot 197 \cdot 263 \cdot 281 \cdot 461 \cdot 521 \\
          &\quad \cdot 3923 \cdot 62731 \cdot 96362257 \cdot 3924006112952623, \text{ and}

    p - 1 &= 2 \cdot 3^{65} \cdot 13 \cdot 17 \cdot 43 \cdot 79 \cdot 157 \cdot 239 \cdot 271 \cdot 283 \cdot 307 \cdot 563 \cdot 599 \\
          &\quad \cdot 607 \cdot 619 \cdot 743 \cdot 827 \cdot 941 \cdot 2357 \cdot 10069
\end{align*}

\(^1\)A number $n$ is $B$-smooth if all the prime factors of $n$ are at most $B$
Contributions

We report SQISign friendly parameters at the higher security levels. The idea utilises an algorithm for finding almost all twin-smooth integers.
Contributions

We report SQISign friendly parameters at the higher security levels
Contributions

We report SQISign friendly parameters at the higher security levels.

The idea utilises an algorithm for finding *almost all* twin-smooth integers.
We report SQISign friendly parameters at the higher security levels. The idea utilises an algorithm for finding *almost all* twin-smooth integers.
Finding Twin-Smooth Integers

CHM Algorithm

Parameter Setup for SQISign

Our Method

Practical SQISign Results

Smooth Twins from XGCD over Polynomial Rings
Finding Twin-Smooth Integers
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 instance, the following consecutive integers are both 7-smooth:

$r = 4374 = 2 \cdot 3^7$, and $r+1 = 4375 = 5^4 \cdot 7$.

Remark: For such a smooth twin $(r, r+1)$, if $p = 2r+1$ is a prime then we recover the case that is of interest to isogenies.

Surprisingly, for a fixed $B$ there are finitely many $B$-smooth twins.
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 instance, the following consecutive integers are both 7-smooth:

$$r = 4374 = 2 \cdot 3^7, \text{ and } r + 1 = 4375 = 5^4 \cdot 7$$
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 instance, the following consecutive integers are both 7-smooth:

$$r = 4374 = 2 \cdot 3^7, \text{ and } r + 1 = 4375 = 5^4 \cdot 7$$

*Remark:* For such a smooth twin $(r, r + 1)$, if $p = 2r + 1$ is a prime then we recover the case that is of interest to isogenies.
**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 instance, the following consecutive integers are both 7-smooth:

$$r = 4374 = 2 \cdot 3^7,$$

and

$$r + 1 = 4375 = 5^4 \cdot 7$$

**Remark:** For such a smooth twin $(r, r+1)$, if $p = 2r + 1$ is a prime then we recover the case that is of interest to isogenies.

Surprisingly, for a fixed $B$ there are finitely many $B$-smooth twins.
Current landscape for finding smooth twins
Constructive Methods

Pell equation
CHM
Current landscape for finding smooth twins

Constructive Methods

- Pell equation
- CHM

Probabilistic methods

- Naïve method
- XGCD/CRT
- $p_n(x) = 2x^n - 1$
- Ideal PTE solutions
The following is a complete characterisation of $B$-smooth twins. It was first proved by Størmer (1897) and later improved algorithmically by Lehmer (1964).

For a $B$-smooth twin $(r, r+1)$, let $x = 2r + 1$ so that $x^2 - 1$ is $B$-smooth and write $x^2 - 1 = Dy^2$ where $D, y$ are $B$-smooth and $D$ is squarefree. Then we can see that $(x, y)$ is a solution to the Pell conic $X^2 - DY^2 = 1$.

Solving all $2\pi(B)$ Pell equations (one for each squarefree and $B$-smooth choice of $D$) will find the complete and finite set of $B$-smooth twins. Lehmer ran this algorithm for $B = 41$, Luca and Najman (2011) ran it with $B = 100$ and most recently Costello (2019) ran it with $B = 113$. 


The following is a complete characterisation of $B$-smooth twins

It was first proved by Størmer (1897) and later improved algorithmically by Lehmer (1964).
Solutions to the Pell equation

The following is a complete characterisation of $B$-smooth twins

It was first proved by Størmer (1897) and later improved algorithmically by Lehmer (1964)

For a $B$-smooth twin $(r, r+1)$, let $x = 2r + 1$ so that $x^2 - 1$ is $B$-smooth and write $x^2 - 1 = Dy^2$ where $D, y$ are $B$-smooth and $D$ is squarefree. Then we can see that $(x, y)$ is a solution to the Pell conic

$$X^2 - DY^2 = 1$$
The following is a complete characterisation of $B$-smooth twins

It was first proved by Størmer (1897) and later improved algorithmically by Lehmer (1964)

For a $B$-smooth twin $(r, r + 1)$, let $x = 2r + 1$ so that $x^2 - 1$ is $B$-smooth and write $x^2 - 1 = Dy^2$ where $D, y$ are $B$-smooth and $D$ is squarefree. Then we can see that $(x, y)$ is a solution to the Pell conic

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

Solving all $2^\pi(B)$ Pell equations (one for each squarefree and $B$-smooth choice of $D$) will find the complete and finite set of $B$-smooth twins
The following is a complete characterisation of $B$-smooth twins

It was first proved by Størmer (1897) and later improved algorithmically by Lehmer (1964)

For a $B$-smooth twin $(r, r + 1)$, let $x = 2r + 1$ so that $x^2 - 1$ is $B$-smooth and write $x^2 - 1 = Dy^2$ where $D, y$ are $B$-smooth and $D$ is squarefree. Then we can see that $(x, y)$ is a solution to the Pell conic

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

Solving all $2\pi(B)$ Pell equations (one for each squarefree and $B$-smooth choice of $D$) will find the complete and finite set of $B$-smooth twins

Lehmer ran this algorithm for $B = 41$, Luca and Najman (2011) ran it with $B = 100$ and most recently Costello (2019) ran it with $B = 113$
The most naive approach is to choose a smooth integer $r$ and hope the $r + 1$ is also smooth. A much better approach is to force smooth factors $s | r$ and $t | r + 1$ of size $s \cdot t \approx r$. This means that instead of hoping that an integer of size $r$ is smooth, you hope that two integers of size $\approx \sqrt{r}$ are smooth.

Algorithmically, this can be achieved with either

• Extended Euclidean algorithm (XGCD) by Costello (2019)
• Chinese remainder theorem (CRT) by De Feo et al. (2020)

The smallest smoothness bound of a $\approx 256$-bit twin for which their sum is a prime is $B = 2^{23}$.10
The most naïve approach is to choose a smooth integer $r$ and hope the $r + 1$ is also smooth.
The most naïve approach is to choose a smooth integer $r$ and hope the $r + 1$ is also smooth.

A much better approach is to force smooth factors $s \mid r$ and $t \mid r + 1$ of size $s \cdot t \approx r$. 

Algorithmically, this can be achieved with either
- Extended Euclidean algorithm (XGCD) by Costello (2019)
- Chinese remainder theorem (CRT) by De Feo et al. (2020)
The most naïve approach is to choose a smooth integer $r$ and hope the $r + 1$ is also smooth.

A much better approach is to force smooth factors $s \mid r$ and $t \mid r + 1$ of size $s \cdot t \approx r$.

This means that instead of hoping that an integer of size $r$ is smooth, you hope that two integers of size $\approx \sqrt{r}$ are smooth.
The most naïve approach is to choose a smooth integer $r$ and hope the $r + 1$ is also smooth.

A much better approach is to force smooth factors $s | r$ and $t | r + 1$ of size $s \cdot t \approx r$.

This means that instead of hoping that an integer of size $r$ is smooth, you hope that two integers of size $\approx \sqrt{r}$ are smooth.

Algorithmically, this can be achieved with either

- Extended Euclidean algorithm (XGCD) by Costello (2019)
- Chinese remainder theorem (CRT) by De Feo et al. (2020)
The most naïve approach is to choose a smooth integer $r$ and hope the $r + 1$ is also smooth.

A much better approach is to force smooth factors $s \mid r$ and $t \mid r + 1$ of size $s \cdot t \approx r$.

This means that instead of hoping that an integer of size $r$ is smooth, you hope that two integers of size $\approx \sqrt{r}$ are smooth.

Algorithmically, this can be achieved with either:

- Extended Euclidean algorithm (XGCD) by Costello (2019)
- Chinese remainder theorem (CRT) by De Feo et al. (2020)

The smallest smoothness bound of a $\approx 256$-bit twin for which their sum is a prime is $B = 2^{23}$.
In Costello's computations with the Pell equations, he noticed that a lot of twins were of the form \((x^2 - 1, x^2)\). To obtain cryptographic sized twins, he generalised this idea to find twins of the form \((x^n - 1, x^n)\) for even \(n\).

The smallest smoothness bound of a \(\approx 256\)-bit twin for which their sum is a prime is \(B = 2^{19}\). We use the notation \(p_n(x) := 2^{x^n} - 1\) to be the result of summing these twins.

More recently, Costello, Meyer and Naehrig (2021) improved this technique by computing twins of the form...
In Costello’s computations with the Pell equations, he noticed that a lot of twins were of the form $(x^2 - 1, x^2)$.
In Costello’s computations with the Pell equations, he noticed that a lot of twins were of the form \((x^2 - 1, x^2)\)

To obtain cryptographic sized twins, he generalised this idea to find twins of the form

\((x^n - 1, x^n)\) for even \(n\)
In Costello’s computations with the Pell equations, he noticed that a lot of twins were of the form \((x^2 - 1, x^2)\)

To obtain cryptographic sized twins, he generalised this idea to find twins of the form

\[(x^n - 1, x^n)\] for even \(n\)

The smallest smoothness bound of a \(\approx 256\)-bit twin for which their sum is a prime is \(B = 2^{19}\)
In Costello’s computations with the Pell equations, he noticed that a lot of twins were of the form \((x^2 - 1, x^2)\)

To obtain cryptographic sized twins, he generalised this idea to find twins of the form

\((x^n - 1, x^n)\) for even \(n\)

The smallest smoothness bound of a \(\approx 256\)-bit twin for which their sum is a prime is \(B = 2^{19}\)

We use the notation \(p_n(x) := 2x^n - 1\) to be the result of summing these twins
In Costello’s computations with the Pell equations, he noticed that a lot of twins were of the form \((x^2 - 1, x^2)\)

To obtain cryptographic sized twins, he generalised this idea to find twins of the form

\((x^n - 1, x^n)\) for even \(n\)

The smallest smoothness bound of a \(\approx 256\)-bit twin for which their sum is a prime is \(B = 2^{19}\)

We use the notation \(p_n(x) := 2x^n - 1\) to be the result of summing these twins

More recently, Costello, Meyer and Naehrig (2021) improved this technique by computing twins of the form
More recently, Costello, Meyer and Naehrig (2021) improved this technique by computing twins of the form

\[(F(x), G(x))\]

where \(F, G\) are polynomials in \(\mathbb{Q}[x]\) that split completely into linear factors.

The decomposition of these polynomials into linear factors increases the smoothness probability.

One can find polynomials of this type from solutions to the ideal Prouhet-Tarry-Escott (PTE) problem.

The smallest smoothness bound of a \(\approx 256\)-bit twin for which their sum is a prime is

\[B = 2^{15}\]
More recently, Costello, Meyer and Naehrig (2021) improved this technique by computing twins of the form

\[(F(x), G(x))\]

where \(F, G\) are polynomials in \(\mathbb{Q}[x]\) that split completely into linear factors.
More recently, Costello, Meyer and Naehrig (2021) improved this technique by computing twins of the form

\[(F(x), G(x))\]

where \(F, G\) are polynomials in \(\mathbb{Q}[x]\) that split completely into linear factors.

The decomposition of these polynomials into linear factors increases the smoothness probability.
More recently, Costello, Meyer and Naehrig (2021) improved this technique by computing twins of the form

\[(F(x), G(x))\]

where \(F, G\) are polynomials in \(\mathbb{Q}[x]\) that split completely into linear factors.

The decomposition of these polynomials into linear factors increases the smoothness probability.

One can find polynomials of this type from solutions to the ideal Prouhet-Tarry-Escott (PTE) problem.
More recently, Costello, Meyer and Naehrig (2021) improved this technique by computing twins of the form

\[(F(x), G(x))\]

where \(F, G\) are polynomials in \(\mathbb{Q}[x]\) that split completely into linear factors.

The decomposition of these polynomials into linear factors increases the smoothness probability.

One can find polynomials of this type from solutions to the ideal Prouhet-Tarry-Escott (PTE) problem.

The smallest smoothness bound of a \(\approx 256\)-bit twin for which their sum is a prime is \(B = 2^{15}\).
CHM Algorithm
An algorithm devised by Conrey, Holmstrom and McLaughlin (2012) that finds almost all $B$-smooth twins.

Start with the initial set of integers $S(0) = \{1, 2, \ldots, B-1\}$—representing the $B$-smooth twins $(1, 2), (2, 3), \ldots, (B-1, B)$.

For each $r, s \in S(0)$ with $r < s$ one computes the following:

$$t = r \cdot s + 1$$

writing $t/t'$ in lowest order terms.

$$S(1) := S(0) \cup \{new \ solutions \ t: t' = t + 1\}$$

Repeat the above but for $S(1)$ instead of $S(0)$. Eventually we must have $S(d+1) = S(d)$ for some $d$ and the algorithm terminates when this happens.
An algorithm devised by Conrey, Holmstrom and McLaughlin (2012) that finds almost all $B$-smooth twins.
CHM algorithm

An algorithm devised by Conrey, Holmstrom and McLaughlin (2012) that finds almost all $B$-smooth twins

Start with the initial set of integers $S^{(0)} = \{1, 2, \cdots, B - 1\}$ – representing the $B$-smooth twins $(1, 2), (2, 3), \cdots, (B - 1, B)$
An algorithm devised by Conrey, Holmstrom and McLaughlin (2012) that finds almost all \( B \)-smooth twins

Start with the initial set of integers \( S^{(0)} = \{1, 2, \cdots, B - 1\} \) — representing the \( B \)-smooth twins \((1, 2), (2, 3), \cdots, (B - 1, B)\)

For each \( r, s \in S^{(0)} \) with \( r < s \) one computes the following

\[
\frac{t}{t'} = \frac{r}{r + 1} \cdot \frac{s + 1}{s}
\]

writing \( t/t' \) in lowest order terms
CHM algorithm

An algorithm devised by Conrey, Holmstrom and McLaughlin (2012) that finds almost all $B$-smooth twins

Start with the initial set of integers $S^{(0)} = \{1, 2, \ldots, B - 1\}$ – representing the $B$-smooth twins $(1, 2), (2, 3), \ldots, (B - 1, B)$

For each $r, s \in S^{(0)}$ with $r < s$ one computes the following

$$\frac{t}{t'} = \frac{r}{r + 1} \cdot \frac{s + 1}{s}$$

writing $t/t'$ in lowest order terms

$$S^{(1)} := S^{(0)} \cup \{\text{new solutions } t : t' = t + 1\}$$
CHM algorithm

An algorithm devised by Conrey, Holmstrom and McLaughlin (2012) that finds *almost all* \( B \)-smooth twins

Start with the initial set of integers \( S^{(0)} = \{1, 2, \cdots, B - 1\} \) – representing the \( B \)-smooth twins \((1, 2), (2, 3), \cdots, (B - 1, B)\)

For each \( r, s \in S^{(0)} \) with \( r < s \) one computes the following

\[
\frac{t}{t'} = \frac{r}{r + 1} \cdot \frac{s + 1}{s}
\]

writing \( t/t' \) in lowest order terms

\[
S^{(1)} := S^{(0)} \cup \{ \text{new solutions } t : t' = t + 1 \}
\]

Repeat the above but for \( S^{(1)} \) instead of \( S^{(0)} \). Eventually we must have \( S^{(d+1)} = S^{(d)} \) for some \( d \) and the algorithm terminates when this happens
We illustrate the algorithm for $B = 5$. The starting set is $S(0) = \{1, 2, 3, 4\}$.

Going through all pairs $r, s \in S(0)$ with $r < s$, we see when the computation yields a new twin smooth pair $(t, t + 1)$:

- $1 + 1 \cdot 2 + 1 = 3$
- $1 + 1 \cdot 3 + 1 = 2$
- $1 + 1 \cdot 4 + 1 = 5$
- $2 + 1 \cdot 3 + 1 = 8$
- $2 + 1 \cdot 4 + 1 = 5$
- $3 + 1 \cdot 4 + 1 = 15$

Hence, we add 5, 8, and 15 to get the next set as $S(1) = \{1, 2, 3, 4, 5, 8, 15\}$.
We illustrate the algorithm for $B = 5$. The starting set is

$$S^{(0)} = \{1, 2, 3, 4\}.$$
We illustrate the algorithm for $B = 5$. The starting set is

$$S^{(0)} = \{1, 2, 3, 4\}.$$ 

Going through all pairs $r, s \in S^{(0)}$ with $r < s$, we see when the computation yields a new twin smooth pair $(t, t + 1)$
We illustrate the algorithm for $B = 5$. The starting set is

$$S^{(0)} = \{1, 2, 3, 4\}.$$ 

Going through all pairs $r, s \in S^{(0)}$ with $r < s$, we see when the computation yields a new twin smooth pair $(t, t + 1)$

$$\frac{1}{1+1} \cdot \frac{2+1}{2} = \frac{3}{4}, \quad \frac{1}{1+1} \cdot \frac{3+1}{3} = \frac{2}{3}, \quad \frac{1}{1+1} \cdot \frac{4+1}{4} = \frac{5}{8},$$

$$\frac{2}{2+1} \cdot \frac{3+1}{3} = \frac{8}{9}, \quad \frac{2}{2+1} \cdot \frac{4+1}{4} = \frac{5}{6}, \quad \text{and} \quad \frac{3}{3+1} \cdot \frac{4+1}{4} = \frac{15}{16}.$$
We illustrate the algorithm for $B = 5$. The starting set is

$$S^{(0)} = \{1, 2, 3, 4\}.$$

Going through all pairs $r, s \in S^{(0)}$ with $r < s$, we see when the computation yields a new twin smooth pair $(t, t + 1)$

$$\frac{1}{1+1} \cdot \frac{2 + 1}{2} = \frac{3}{4}, \quad \frac{1}{1+1} \cdot \frac{3 + 1}{3} = \frac{2}{3}, \quad \frac{1}{1+1} \cdot \frac{4 + 1}{4} = \frac{5}{8},$$

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

Hence, we add $5, 8$ and $15$ to get the next set as

$$S^{(1)} = \{1, 2, 3, 4, 5, 8, 15\}$$
Hence, we add 5, 8 and 15 to get the next set as

\[ S^{(1)} = \{1, 2, 3, 4, 5, 8, 15\} \]
Hence, we add 5, 8 and 15 to get the next set as

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

In the second iteration, only two new twins are found

$$\frac{3}{3+1} \cdot \frac{5+1}{5} = \frac{9}{10}, \quad \text{and} \quad \frac{4}{4+1} \cdot \frac{5+1}{5} = \frac{24}{25}$$
Hence, we add 5, 8 and 15 to get the next set as

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

In the second iteration, only two new twins are found

\[ \frac{3}{3+1} \cdot \frac{5+1}{5} = \frac{9}{10}, \quad \text{and} \quad \frac{4}{4+1} \cdot \frac{5+1}{5} = \frac{24}{25} \]

Hence, we add 9 and 24 to get the next set as

\[ S^{(2)} = \{1, 2, 3, 4, 5, 8, 9, 15, 24\} \]
Hence, we add 5, 8 and 15 to get the next set as

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

In the second iteration, only two new twins are found

\[ \frac{3}{3 + 1} \cdot \frac{5 + 1}{5} = \frac{9}{10}, \quad \text{and} \quad \frac{4}{4 + 1} \cdot \frac{5 + 1}{5} = \frac{24}{25} \]

Hence, we add 9 and 24 to get the next set as

\[ S^{(2)} = \{1, 2, 3, 4, 5, 8, 9, 15, 24\} \]

In the third CHM iterations we add 80 and get

\[ S^{(3)} = \{1, 2, 3, 4, 5, 8, 9, 15, 24, 80\} \]
In the third CHM iterations we add 80

\[ S^{(3)} = \{1, 2, 3, 4, 5, 8, 9, 15, 24, 80\} \]
In the third CHM iterations we add 80

\[ S^{(3)} = \{1, 2, 3, 4, 5, 8, 9, 15, 24, 80\} \]

The fourth iteration does not produce any new numbers, i.e. we have \( S^{(4)} = S^{(3)} \)
In the third CHM iterations we add 80

\[ S^{(3)} = \{1, 2, 3, 4, 5, 8, 9, 15, 24, 80\} \]

The fourth iteration does not produce any new numbers, i.e. we have \[ S^{(4)} = S^{(3)} \]

Doing the corresponding computations with the Pell equations verifies that this is indeed the full set of 5-smooth twins
In the third CHM iterations we add 80

\[ S^{(3)} = \{1, 2, 3, 4, 5, 8, 9, 15, 24, 80\} \]

The fourth iteration does not produce any new numbers, i.e. we have \( S^{(4)} = S^{(3)} \)

Doing the corresponding computations with the Pell equations verifies that this is indeed the full set of 5-smooth twins

In general this method does not guarantee to produce all \( B \)-smooth twins
In the third CHM iterations we add 80

\[ S^{(3)} = \{1, 2, 3, 4, 5, 8, 9, 15, 24, 80\} \]

The fourth iteration does not produce any new numbers, i.e. we have
\[ S^{(4)} = S^{(3)} \]

Doing the corresponding computations with the Pell equations verifies that this is indeed the full set of 5-smooth twins

In general this method does not guarantee to produce all \( B \)-smooth twins

Applying the CHM algorithm with \( B = 7 \), we get \( S^{(5)} = S^{(4)} \)
In the third CHM iterations we add 80

\[ S^{(3)} = \{1, 2, 3, 4, 5, 8, 9, 15, 24, 80\} \]

The fourth iteration does not produce any new numbers, i.e. we have \( S^{(4)} = S^{(3)} \)

Doing the corresponding computations with the Pell equations verifies that this is indeed the full set of 5-smooth twins

In general this method does not guarantee to produce all \( B \)-smooth twins

Applying the CHM algorithm with \( B = 7 \), we get \( S^{(5)} = S^{(4)} \)

\[ S^{(4)} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 14, 15, 20, 24, 27, 35, 48, 49, 63, 80, 125, 224, 2400\} \]
Applying the CHM algorithm with $B = 7$, we get $S^{(5)} = S^{(4)}$ where

$$S^{(4)} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 14, 15, 20, 24, 27, 35, 48, 49, 63, 80, 125, 224, 2400\}$$
Applying the CHM algorithm with $B = 7$, we get $S^{(5)} = S^{(4)}$ where

$$S^{(4)} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 14, 15, 20, 24, 27, 35, 48, 49, 63, 80, 125, 224, 2400\}$$

This is not the full set of 7-smooth twins and you miss the largest 7-smooth twin: $(4374, 4375)$
Applying the CHM algorithm with $B = 7$, we get $S^{(5)} = S^{(4)}$ where

$$S^{(4)} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 14, 15, 20, 24, 27, 35, 48, 49, 63, 80, 125, 224, 2400\}$$

This is not the full set of 7-smooth twins and you miss the largest 7-smooth twin: $(4374, 4375)$

However, running this with $B = 11$ will find this missing twin.
Applying the CHM algorithm with $B = 7$, we get $S^{(5)} = S^{(4)}$ where

$$S^{(4)} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 14, 15, 20, 24, 27, 35, 48, 49, 63, 80, 125, 224, 2400\}$$

This is not the full set of 7-smooth twins and you miss the largest 7-smooth twin: $(4374, 4375)$

However, running this with $B = 11$ will find this missing twin

When $11 \leq B < 41$ the algorithm finds all $B$-smooth twins but with $B \geq 41$ the algorithm will (at least conjecturally) find almost all twins
CHM in action

Applying the CHM algorithm with $B = 7$, we get $S^{(5)} = S^{(4)}$ where

$$S^{(4)} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 14, 15, 20, 24, 27, 35, 48, 49, 63, 80, 125, 224, 2400\}$$

This is not the full set of 7-smooth twins and you miss the largest 7-smooth twin: $(4374, 4375)$

However, running this with $B = 11$ will find this missing twin

When $11 \leq B < 41$ the algorithm finds all $B$-smooth twins but with $B \geq 41$ the algorithm will (at least conjecturally) find almost all twins

The original authors ran CHM with $B = 100$ and found all 100-smooth twins with the exception of 37 solutions. They subsequently ran it with $B = 200$ which took 2 weeks for them to compute
Our experiments

We heavily optimised the CHM algorithm and are able to run it with $B = 200$ much faster. Subsequently we ran it fully for $B = 547$ – the largest twin found was the following 122-bit twin:

$$r = 5\cdot7\cdot13^2\cdot17^2\cdot19\cdot29\cdot41\cdot109\cdot163\cdot173\cdot239\cdot241^2\cdot271\cdot283\cdot499\cdot509,$$

and $r+1 = 2\cdot3^2\cdot31^2\cdot43^2\cdot47^2\cdot83^2\cdot103^2\cdot311^2\cdot479^2\cdot523^2$.

An additional 2,649 twins were found that are 200-smooth through this computation. We also introduced some other optimisations that made it possible for us to run larger values of $B$.

The computation only took us a mere 7 minutes to run on a laptop.
We heavily optimised the CHM algorithm and are able to run it with $B = 200$ much faster\(^2\)!

\(^2\)The computation only took us a mere 7 minutes to run on a laptop
Our experiments

We heavily optimised the CHM algorithm and are able to run it with $B = 200$ much faster\(^2\)!

Subsequently we ran it fully for $B = 547$ – the largest twin found was the following 122-bit twin:

\[
\begin{align*}
    r &= 5^4 \cdot 7 \cdot 13^2 \cdot 17^2 \cdot 19 \cdot 29 \cdot 41 \cdot 109 \cdot 163 \cdot 173 \cdot 239 \cdot 241^2 \\
        &\quad \cdot 271 \cdot 283 \cdot 499 \cdot 509, \text{ and} \\
    r + 1 &= 2^8 \cdot 3^2 \cdot 31^2 \cdot 43^2 \cdot 47^2 \cdot 83^2 \cdot 103^2 \cdot 311^2 \cdot 479^2 \cdot 523^2.
\end{align*}
\]

\(^2\)The computation only took us a mere 7 minutes to run on a laptop.
Our experiments

We heavily optimised the CHM algorithm and are able to run it with $B = 200$ much faster!\footnote{The computation only took us a mere 7 minutes to run on a laptop}

Subsequently we ran it fully for $B = 547$ – the largest twin found was the following 122-bit twin

\[
r = 5^4 \cdot 7 \cdot 13^2 \cdot 17^2 \cdot 19 \cdot 29 \cdot 41 \cdot 109 \cdot 163 \cdot 173 \cdot 239 \cdot 241^2 \\
\cdot 271 \cdot 283 \cdot 499 \cdot 509, \text{ and } \\
r + 1 = 2^8 \cdot 3^2 \cdot 31^2 \cdot 43^2 \cdot 47^2 \cdot 83^2 \cdot 103^2 \cdot 311^2 \cdot 479^2 \cdot 523^2.
\]

An additional 2,649 twins were found that are 200-smooth through this computation
Our experiments

We heavily optimised the CHM algorithm and are able to run it with $B = 200$ much faster! 

Subsequently we ran it fully for $B = 547$ – the largest twin found was the following 122-bit twin

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

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

An additional 2,649 twins were found that are 200-smooth through this computation

We also introduced some other optimisations that made it possible for us to run larger values of $B$

$^2$The computation only took us a mere 7 minutes to run on a laptop
## Optimisations

<table>
<thead>
<tr>
<th>Variant Parameter</th>
<th>Runtime Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHM</td>
<td>4705s</td>
</tr>
<tr>
<td>$k = 2$</td>
<td>2300s</td>
</tr>
<tr>
<td>$k = 1.5$</td>
<td>226s</td>
</tr>
<tr>
<td>$k = 1.05$</td>
<td>27s</td>
</tr>
<tr>
<td>$R = 10000$</td>
<td>82s</td>
</tr>
<tr>
<td>$R = 5000$</td>
<td>35s</td>
</tr>
<tr>
<td>$R = 1000$</td>
<td>16s</td>
</tr>
</tbody>
</table>

### Table 1:
Performance results for different variants of our CHM implementation for smoothness bound $B = 300$. Speedup factors refer to the full CHM variant.

- **global-$k$**: Fix some $1 < k < 2$ and only check $(r, s)$ with $r < s < k \cdot r$.
- **constant-range** $R$: Fix a range $R$ and only check $(r, s)$ for the $R$ successors $s$ of $r$ in each iteration.
Optimisations

<table>
<thead>
<tr>
<th>Variant</th>
<th>Parameter</th>
<th>Runtime</th>
<th>Speedup</th>
<th>#twins</th>
<th>#twins from largest 100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full CHM</td>
<td>-</td>
<td>4705s</td>
<td>1</td>
<td>2300724</td>
<td>100</td>
</tr>
<tr>
<td>global-k</td>
<td>$k = 2.0$</td>
<td>364s</td>
<td>13</td>
<td>2289000</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>$k = 1.5$</td>
<td>226s</td>
<td>21</td>
<td>2282741</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>$k = 1.05$</td>
<td>27s</td>
<td>174</td>
<td>2206656</td>
<td>65</td>
</tr>
<tr>
<td>constant-range</td>
<td>$R = 10000$</td>
<td>82s</td>
<td>57</td>
<td>2273197</td>
<td>93</td>
</tr>
<tr>
<td></td>
<td>$R = 5000$</td>
<td>35s</td>
<td>134</td>
<td>2247121</td>
<td>87</td>
</tr>
<tr>
<td></td>
<td>$R = 1000$</td>
<td>16s</td>
<td>294</td>
<td>2074530</td>
<td>75</td>
</tr>
</tbody>
</table>

**Table 1:** Performance results for different variants of our CHM implementation for smoothness bound $B = 300$. Speedup factors refer to the full CHM variant.
## Optimisations

<table>
<thead>
<tr>
<th>Variant</th>
<th>Parameter</th>
<th>Runtime</th>
<th>Speedup</th>
<th>#twins</th>
<th>#twins from largest 100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full CHM</td>
<td>-</td>
<td>4705s</td>
<td>1</td>
<td>2300724</td>
<td>100</td>
</tr>
<tr>
<td>global-(k)</td>
<td>(k = 2.0)</td>
<td>364s</td>
<td>13</td>
<td>2289000</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td>(k = 1.5)</td>
<td>226s</td>
<td>21</td>
<td>2282741</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>(k = 1.05)</td>
<td>27s</td>
<td>174</td>
<td>2206656</td>
<td>65</td>
</tr>
<tr>
<td>constant-range</td>
<td>(R = 10000)</td>
<td>82s</td>
<td>57</td>
<td>2273197</td>
<td>93</td>
</tr>
<tr>
<td></td>
<td>(R = 5000)</td>
<td>35s</td>
<td>134</td>
<td>2247121</td>
<td>87</td>
</tr>
<tr>
<td></td>
<td>(R = 1000)</td>
<td>16s</td>
<td>294</td>
<td>2074530</td>
<td>75</td>
</tr>
</tbody>
</table>

**Table 1:** Performance results for different variants of our CHM implementation for smoothness bound \(B = 300\). Speedup factors refer to the full CHM variant.

**global-\(k\):** Fix some \(1 < k \leq 2\) and only check \((r, s)\) with \(r < s < k \cdot r\)

**constant-range:** Fix a range \(R\) and only check \((r, s)\) for the \(R\) successors \(s\) of \(r\) in each iteration
Our experiments

We ran these optimisations for larger $B$. CHM was run with $B = 1300$ using the constant-range optimisation with a range $R = 5000$, specifically targeting twins ($r, r+1$) with $r > 2^{115}$ - the largest twin found was the following 145-bit twins

\[ r = 2^{5} \cdot 5 \cdot 7 \cdot 11 \cdot 2^{2} \cdot 13 \cdot 23 \cdot 53 \cdot 71 \cdot 109 \cdot 127 \cdot 131 \cdot 193 \cdot 251 \cdot 283 \cdot 307 \cdot 359 \cdot 367 \],

and $r+1 = 2^{29} \cdot 29 \cdot 31 \cdot 43 \cdot 59 \cdot 61 \cdot 73 \cdot 79 \cdot 89 \cdot 167 \cdot 401 \cdot 419 \cdot 653 \cdot 1277$. Other experiments were done with $B = 2^{11}$. Unfortunately, choosing $B$ large enough and running this to give you cryptographic sized twins is infeasible due to time and memory limitations.
Our experiments

We ran these optimisations for larger $B$

Unfortunately, choosing $B$ large enough and running this to give you cryptographic sized twins is infeasible due to time and memory limitations.
Our experiments

We ran these optimisations for larger $B$.

CHM was run with $B = 1300$ using the constant-range optimisation with a range $R = 5000$, specifically targeting twins $(r, r + 1)$ with $r > 2^{115}$. The largest twin found was the following 145-bit twins:

$$r = 2^5 \cdot 5 \cdot 7 \cdot 11^2 \cdot 13 \cdot 23 \cdot 53 \cdot 71 \cdot 109 \cdot 127 \cdot 131 \cdot 193 \cdot 251 \cdot 283 \cdot 307 \cdot 359 \cdot 367 \cdot 461 \cdot 613 \cdot 653 \cdot 1277,$$

and

$$r + 1 = 3^2 \cdot 29^2 \cdot 31^2 \cdot 43^2 \cdot 59^2 \cdot 61^2 \cdot 73^2 \cdot 79^2 \cdot 89^2 \cdot 167^2 \cdot 401^2 \cdot 419^2.$$
Our experiments

We ran these optimisations for larger $B$

CHM was run with $B = 1300$ using the constant-range optimisation with a range $R = 5000$, specifically targeting twins $(r, r+1)$ with $r > 2^{115}$ - the largest twin found was the following 145-bit twins

$$r = 2^5 \cdot 5 \cdot 7 \cdot 11^2 \cdot 13 \cdot 23 \cdot 53 \cdot 71 \cdot 109 \cdot 127 \cdot 131 \cdot 193 \cdot 251$$

$$\cdot 283 \cdot 307 \cdot 359 \cdot 367 \cdot 461 \cdot 613 \cdot 653 \cdot 1277,$$

and

$$r + 1 = 3^2 \cdot 29^2 \cdot 31^2 \cdot 43^2 \cdot 59^2 \cdot 61^2 \cdot 73^2 \cdot 79^2 \cdot 89^2 \cdot 167^2 \cdot 401^2 \cdot 419^2.$$

Other experiments were done with $B = 2^{11}$
Our experiments

We ran these optimisations for larger $B$

CHM was run with $B = 1300$ using the constant-range optimisation with a range $R = 5000$, specifically targeting twins $(r, r+1)$ with $r > 2^{115}$ - the largest twin found was the following 145-bit twins

$$r = 2^5 \cdot 5 \cdot 7 \cdot 11^2 \cdot 13 \cdot 23 \cdot 53 \cdot 71 \cdot 109 \cdot 127 \cdot 131 \cdot 193 \cdot 251$$

$$\cdot 283 \cdot 307 \cdot 359 \cdot 367 \cdot 461 \cdot 613 \cdot 653 \cdot 1277$$

and

$$r + 1 = 3^2 \cdot 29^2 \cdot 31^2 \cdot 43^2 \cdot 59^2 \cdot 61^2 \cdot 73^2 \cdot 79^2 \cdot 89^2 \cdot 167^2 \cdot 401^2 \cdot 419^2.$$

Other experiments were done with $B = 2^{11}$

Unfortunately, choosing $B$ large enough and running this to give you cryptographic sized twins is infeasible due to time and memory limitations.
Parameter Setup for SQISign
SQISign requirements

- **Setup**
  - Cryptographic prime $p$ (of $\approx 256, 384, 512$-bits), such that $p^2 - 1 = 2^f \cdot T \cdot R$, where $f$ is a "relatively" large exponent, $T$ is an odd smooth cofactor of size $\approx p^{5/4 + \epsilon}$ and $R$ can have rough factors.

*Remark:* The necessity of the power of two can in theory be replaced by a powersmooth integer $L$.

If $B$ is the smoothness bound of $T$, the quantity $\sqrt{B}/f$ is a rough cost metric for the signing algorithm in SQISign.
## SQISign requirements

### Setup

Cryptographic prime $p$ (of $\approx 256, 384, 512$-bits), such that

$$p^2 - 1 = 2^f \cdot T \cdot R,$$

where $f$ is a “relatively” large exponent, $T$ is an odd smooth cofactor of size $\approx p^{5/4+\epsilon}$ and $R$ can have rough factors.
**SQISign requirements**

<table>
<thead>
<tr>
<th>Setup</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cryptographic prime $p$ (of $\approx 256, 384, 512$-bits), such that</td>
</tr>
<tr>
<td>$p^2 - 1 = 2^f \cdot T \cdot R,$</td>
</tr>
<tr>
<td>where $f$ is a “relatively” large exponent, $T$ is an odd smooth cofactor of size $\approx p^{5/4+\epsilon}$ and $R$ can have rough factors</td>
</tr>
</tbody>
</table>

*Remark:* The necessity of the power of two can in theory be replaced by a powersmooth integer $L$
## SQISign requirements

### Setup

Cryptographic prime $p$ (of $\approx 256, 384, 512$-bits), such that

$$p^2 - 1 = 2^f \cdot T \cdot R,$$

where $f$ is a “relatively” large exponent, $T$ is an odd smooth cofactor of size $\approx p^{5/4+\epsilon}$ and $R$ can have rough factors.

**Remark:** The necessity of the power of two can in theory be replaced by a powersmooth integer $L$.

If $B$ is the smoothness bound of $T$, the quantity $\sqrt{B}/f$ is a rough cost metric for the signing algorithm in SQISign.
De Feo, Kohel, Leroux, Petit and Wesolowski (2020, 2022) explored the XGCD/CRT method to find SQISign friendly parameters. They forced $p \pm 1 \equiv 0 \pmod{2^\alpha}$, $p \pm 1 \equiv 0 \pmod{3^\beta}$, $p \pm 1 \equiv 0 \pmod{q}$ for small primes $q$, $p \pm 1 \equiv 0 \pmod{q'}$ for other small primes $q'$ and used CRT to find $p$. With this technique, they found SQISign friendly primes whose smooth cofactor $T$ is $2^{12}$-smooth.
De Feo$^2$, Kohel, Leroux$^2$, Petit and Wesolowski$^2$ (2020, 2022) explored the XGCD/CRT method to find SQISign friendly parameters.
De Feo\textsuperscript{2}, Kohel, Leroux\textsuperscript{2}, Petit and Wesolowski\textsuperscript{2} (2020, 2022) explored the XGCD/CRT method to find SQISign friendly parameters. They forced

\[
p \pm 1 = 0 \mod 2^\alpha,
\]
\[
p \mp 1 = 0 \mod 3^\beta,
\]
\[
p \pm 1 = 0 \mod q \quad \text{for small primes } q,
\]
\[
p \mp 1 = 0 \mod q' \quad \text{for other small primes } q'
\]

and used CRT to find \( p \).
De Feo\textsuperscript{2}, Kohel, Leroux\textsuperscript{2}, Petit and Wesolowski\textsuperscript{2} (2020,2022) explored the XGCD/CRT method to find SQISign friendly parameters. They forced

\[ p \pm 1 = 0 \mod 2^\alpha, \]
\[ p \mp 1 = 0 \mod 3^\beta, \]
\[ p \pm 1 = 0 \mod q \text{ for small primes } q, \]
\[ p \mp 1 = 0 \mod q' \text{ for other small primes } q' \]

and used CRT to find \( p \).

With this technique, they found SQISign friendly primes whose smooth cofactor \( T \) is \( 2^{12} \)-smooth.
Exhaustive search using $p_4(x)$
Exhaustive search using $p_4(x)$

De Feo, Leroux and Wesolowski used the polynomial $p_4(x) = 2x^4 - 1$ to attempt to find SQISign friendly primes.
De Feo, Leroux and Wesolowski used the polynomial $p_4(x) = 2x^4 - 1$ to attempt to find SQISign friendly primes. Note that in this setting we have

$$p_4(x) - 1 = 2(x - 1)(x + 1)(x^2 + 1)$$
De Feo, Leroux and Wesolowski used the polynomial $p_4(x) = 2x^4 - 1$ to attempt to find SQISign friendly primes. Note that in this setting we have

$$p_4(x) - 1 = 2(x - 1)(x + 1)(x^2 + 1)$$

Their idea is the following:

1. Replace $x \mapsto 2^{15} \cdot x$ in the polynomial $p_4$
2. Sieve the interval $x \in [2^{47}, 2^{49}]$ to identify $2^{11}$-smooth integers
3. Compute the $2^{11}$-smooth odd cofactor, $T$, of
   $$x^4(2^{15}x - 1)(2^{15}x + 1)(2^{30}x^2 + 1)$$
4. Record it if $T > p^{5/4+\epsilon}$ and the evaluation $p$ is prime

---

$^3$This guarantees at least a factor of $2^{61}$ in $p + 1$ after evaluation
Exhaustive search using $p_4(x)$

De Feo, Leroux and Wesolowski used the polynomial $p_4(x) = 2x^4 - 1$ to attempt to find SQISign friendly primes. Note that in this setting we have

$$p_4(x) - 1 = 2(x - 1)(x + 1)(x^2 + 1)$$

Their idea is the following:

1. Replace $x \mapsto 2^{15} \cdot x$ in the polynomial $p_4$
2. Sieve the interval $x \in [2^{47}, 2^{49}]$ to identify $2^{11}$-smooth integers
3. Compute the $2^{11}$-smooth odd cofactor, $T$, of

$$x^4(2^{15}x - 1)(2^{15}x + 1)(2^{30}x^2 + 1)$$

4. Record it if $T > p^{5/4+\epsilon}$ and the evaluation $p$ is prime

They found 15 primes of this type

$^3$This guarantees at least a factor of $2^{61}$ in $p + 1$ after evaluation
Comparison of their primes

254-bit prime $p$

\[ p + 1 = 2^65 \cdot 5^2 \cdot 7 \cdot 11 \cdot 19 \cdot 29 \cdot 37^2 \cdot 47 \cdot 197 \cdot 263 \cdot 281 \cdot 461 \cdot 521 \cdot 3923 \cdot R, \]

and

\[ p - 1 = 2^6 \cdot 3^{65} \cdot 13 \cdot 17 \cdot 43 \cdot 79 \cdot 157 \cdot 239 \cdot 271 \cdot 283 \cdot 307 \cdot 563 \cdot 599 \cdot 607 \cdot 619 \cdot 743 \cdot 827 \cdot 941 \cdot 2357 \cdot 10069. \]

In practice, the first of these performs slightly better despite having a larger signing cost metric - owing in large part to the large power of 3 but also the amount of small smoothness.
Comparison of their primes

<table>
<thead>
<tr>
<th>254-bit prime $p$:</th>
<th>256-bit prime $p = p_4(r) = 2r^4 - 1$ with $r = 2^{15} \cdot 411099446409699$:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p + 1 = 2^{65} \cdot 5^2 \cdot 7 \cdot 11 \cdot 19 \cdot 29^2 \cdot 37^2$</td>
<td>$p + 1 = 2^{61} \cdot 3^4 \cdot 31^4 \cdot 127^4 \cdot 307^4$</td>
</tr>
<tr>
<td>$\cdot 47 \cdot 197 \cdot 263 \cdot 281 \cdot 461$</td>
<td>$\cdot 353^4 \cdot 509^4 \cdot 631^4$</td>
</tr>
<tr>
<td>$\cdot 521 \cdot 3923 \cdot R$, and</td>
<td></td>
</tr>
<tr>
<td>$p - 1 = 2 \cdot 3^{65} \cdot 13 \cdot 17 \cdot 43 \cdot 79 \cdot 157$</td>
<td>$p - 1 = 2 \cdot 5^2 \cdot 13 \cdot 17 \cdot 29 \cdot 37 \cdot 41$</td>
</tr>
<tr>
<td>$\cdot 239 \cdot 271 \cdot 283 \cdot 307 \cdot 563$</td>
<td>$\cdot 103 \cdot 109 \cdot 149 \cdot 191 \cdot 269$</td>
</tr>
<tr>
<td>$\cdot 599 \cdot 607 \cdot 619 \cdot 743 \cdot 827$</td>
<td>$\cdot 313 \cdot 367 \cdot 379 \cdot 503 \cdot 587$</td>
</tr>
<tr>
<td>$\cdot 941 \cdot 2357 \cdot 10069$</td>
<td>$\cdot 683 \cdot 1217 \cdot 1487 \cdot R$</td>
</tr>
</tbody>
</table>
## Comparison of their primes

<table>
<thead>
<tr>
<th>254-bit prime $p$:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p + 1 = 2^{65} \cdot 5^2 \cdot 7 \cdot 11 \cdot 29^2 \cdot 37^2$</td>
</tr>
<tr>
<td>$\cdot 47 \cdot 197 \cdot 263 \cdot 281 \cdot 461$</td>
</tr>
<tr>
<td>$\cdot 521 \cdot 3923 \cdot R$, and</td>
</tr>
<tr>
<td>$p - 1 = 2 \cdot 3^{65} \cdot 13 \cdot 17 \cdot 43 \cdot 79 \cdot 157$</td>
</tr>
<tr>
<td>$\cdot 239 \cdot 271 \cdot 283 \cdot 307 \cdot 563$</td>
</tr>
<tr>
<td>$\cdot 599 \cdot 607 \cdot 619 \cdot 743 \cdot 827$</td>
</tr>
<tr>
<td>$\cdot 941 \cdot 2357 \cdot 10069$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>256-bit prime $p = p_4(r) = 2r^4 - 1$ with $r = 2^{15} \cdot 411099446409699$:</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p + 1 = 2^{61} \cdot 3^4 \cdot 31^4 \cdot 127^4 \cdot 307^4$</td>
</tr>
<tr>
<td>$\cdot 353^4 \cdot 509^4 \cdot 631^4$</td>
</tr>
<tr>
<td>$p - 1 = 2 \cdot 5^2 \cdot 13 \cdot 17 \cdot 29 \cdot 37 \cdot 41$</td>
</tr>
<tr>
<td>$\cdot 103 \cdot 109 \cdot 149 \cdot 191 \cdot 269$</td>
</tr>
<tr>
<td>$\cdot 313 \cdot 367 \cdot 379 \cdot 503 \cdot 587$</td>
</tr>
<tr>
<td>$\cdot 683 \cdot 1217 \cdot 1487 \cdot R$</td>
</tr>
</tbody>
</table>

In practice, the first of these performs slightly better despite having a larger signing cost metric - owing in large part to the large power of 3 but also the amount of small smoothness\(^4\) is also larger

\(^4\)A cofactor that is, say, 100-smooth
Other primes in the literature

In the context of other isogeny-based applications, larger primes have been found for which $p \pm 1$ is smooth. As part of the parameter search for Sélétar by De Feo et al. (2021), they found the following SQISign friendly parameter:

- A 389-bit prime $p = p_{12}(r) = 2^{r_{12}} - 1$ with $r = 5114946480$:
  - $p + 1 = 2^{49} \cdot 3^{12} \cdot 5^{12} \cdot 7^{12} \cdot 73^{12} \cdot 179^{12} \cdot 233^{12}$,
  - and $p - 1 = 2 \cdot 13 \cdot 97 \cdot 379 \cdot 661 \cdot 853 \cdot 1693 \cdot 2767 \cdot 3121 \cdot 4297 \cdot 8623 \cdot 8629 \cdot 17929 \cdot 21937 \cdot 31327 \cdot R$.

This could be used in theory but we find better more applicable primes.
Other primes in the literature

In the context of other isogeny-based applications, larger primes have been found for which \( p \pm 1 \) is smooth.
In the context of other isogeny-based applications, larger primes have been found for which \( p \pm 1 \) is smooth.

As part of the parameter search for Séta by De Feo et al. (2021), they found the following SQISign friendly parameter:

389-bit prime \( p = p_{12}(r) = 2r^{12} - 1 \) with \( r = 5114946480 \):

\[
\begin{align*}
p + 1 &= 2^{49} \cdot 3^{12} \cdot 5^{12} \cdot 7^{12} \cdot 73^{12} \cdot 179^{12} \cdot 233^{12}, \\
p - 1 &= 2 \cdot 13 \cdot 97 \cdot 379 \cdot 661 \cdot 853 \cdot 1693 \cdot 2767 \cdot 3121 \cdot 4297 \cdot 8623 \\
&\quad \cdot 8629 \cdot 17929 \cdot 21937 \cdot 31327 \cdot R
\end{align*}
\]
Other primes in the literature

In the context of other isogeny-based applications, larger primes have been found for which $p \pm 1$ is smooth.

As part of the parameter search for Séta by De Feo et al. (2021), they found the following SQISign friendly parameter:

A 389-bit prime $p = p_{12}(r) = 2r^{12} - 1$ with $r = 5114946480$:

\[
p + 1 = 2^{49} \cdot 3^{12} \cdot 5^{12} \cdot 7^{12} \cdot 73^{12} \cdot 179^{12} \cdot 233^{12}, \text{ and}
\]

\[
p - 1 = 2 \cdot 13 \cdot 97 \cdot 379 \cdot 661 \cdot 853 \cdot 1693 \cdot 2767 \cdot 3121 \cdot 4297 \cdot 8623 \\
\cdot 8629 \cdot 17929 \cdot 21937 \cdot 31327 \cdot R
\]

This could be used in theory but we find better more applicable primes.
Our Method
General framework

For the polynomials $p_n(x) = 2x^n - 1$, we have

$4x^n(x-1) | p_{2n}(x) - 1$ for all $n$, and

$4x^n(x-1)(x+1) | p_{2n}(x) - 1$ when $n$ is even.

So, for a smooth twin $(r, r \pm 1)$ found using the CHM machinery, we compute the evaluation $p = p_n(r) = 2r^n - 1$.

The amount of guaranteed smoothness in $p_{2n} - 1$ is $\approx p^{1+1/n}$.

Depending on the choice of $n$ and the power of two in $p_{2n} - 1$, this might not necessarily give us something that is suitable for SQISign. If so, then compute the other smooth factors of $p_{2n} - 1$ and check to see if the combined cofactor is larger than $p^{5/4 + \epsilon}$.26
General framework

For the polynomials $p_n(x) = 2x^n - 1$, we have

$$4x^n(x - 1) \mid p^2_n(x) - 1 \quad \text{for all } n, \text{ and}$$

$$4x^n(x - 1)(x + 1) \mid p^2_n(x) - 1 \quad \text{when } n \text{ is even}$$
For the polynomials $p_n(x) = 2x^n - 1$, we have

$$4x^n(x - 1) \mid p_n^2(x) - 1 \quad \text{for all } n, \text{ and}$$

$$4x^n(x - 1)(x + 1) \mid p_n^2(x) - 1 \quad \text{when } n \text{ is even}$$

So, for a smooth twin $(r, r \pm 1)$ found using the CHM machinery, we compute the evaluation

$$p = p_n(r) = 2r^n - 1$$
For the polynomials $p_n(x) = 2x^n - 1$, we have

\[ 4x^n(x - 1) \mid p_n^2(x) - 1 \quad \text{for all } n, \text{ and} \]
\[ 4x^n(x - 1)(x + 1) \mid p_n^2(x) - 1 \quad \text{when } n \text{ is even} \]

So, for a smooth twin $(r, r \pm 1)$ found using the CHM machinery, we compute the evaluation

\[ p = p_n(r) = 2r^n - 1 \]

The amount of \textit{guaranteed smoothness} in $p^2 - 1$ is $\approx p^{1+1/n}$
For the polynomials $p_n(x) = 2x^n - 1$, we have

$$4x^n(x - 1) \mid p_n^2(x) - 1 \quad \text{for all } n, \text{ and}$$

$$4x^n(x - 1)(x + 1) \mid p_n^2(x) - 1 \quad \text{when } n \text{ is even}$$

So, for a smooth twin $(r, r \pm 1)$ found using the CHM machinery, we compute the evaluation

$$p = p_n(r) = 2r^n - 1$$

The amount of guaranteed smoothness in $p^2 - 1$ is $\approx p^{1+1/n}$

Depending on the choice of $n$ and the power of two in $p^2 - 1$, this might not necessarily give us something that is suitable for SQISign
For the polynomials $p_n(x) = 2x^n - 1$, we have

$$4x^n(x - 1) \mid p^2_n(x) - 1 \quad \text{for all } n,$$

and

$$4x^n(x - 1)(x + 1) \mid p^2_n(x) - 1 \quad \text{when } n \text{ is even}.$$

So, for a smooth twin $(r, r \pm 1)$ found using the CHM machinery, we compute the evaluation

$$p = p_n(r) = 2r^n - 1.$$

The amount of guaranteed smoothness in $p^2 - 1$ is $\approx p^{1+1/n}$.

Depending on the choice of $n$ and the power of two in $p^2 - 1$, this might not necessarily give us something that is suitable for SQISign.

If so then compute the other smooth factors of $p^2 - 1$ and check to see if the combined cofactor is larger than $p^{5/4+\epsilon}$. 
Choosing $n$

Let $T' = 2^f T$. Current implementation of SQISign has $f \approx \lfloor \log_2 \left( \frac{p_1}{4} \right) \rfloor$ which translates to $T' \approx \frac{p_3}{2} + \epsilon$.

The smaller $n$ is, we get more guaranteed smoothness from the twin. This comes at a cost of finding larger twins $n = 2^r$: For a smooth twin $(r, r \pm 1)$, let $p = 2^{r^2} - 1$. Here we have $p - 1 = 2(r - 1)(r + 1)$.

The amount of guaranteed smoothness from the twin alone is $\frac{p_3}{2}$. If the power of two in $p^2 - 1$ is less than $\lfloor \log_2 \left( \frac{p_1}{4} \right) \rfloor$ then we have enough smoothness for a SQISign parameter.
Let $T' = 2^f T$. Current implementation of SQISign has $f \approx \lfloor \log_2(p^{1/4}) \rfloor$ which translates to $T' \approx p^{3/2 + \epsilon}$. The smaller $n$ is, we get more guaranteed smoothness from the twin. This comes at a cost of finding larger twins $n = 2^r$: For a smooth twin $(r, r \pm 1)$, let $p = 2r^2 - 1$. Here we have $p - 1 = 2(r - 1)(r + 1)$. The amount of guaranteed smoothness from the twin alone is $p^{3/2}$. If the power of two in $p^2 - 1$ is less than $\lfloor \log_2(p^{1/4}) \rfloor$ then we have enough smoothness for a SQISign parameter.
Choosing \( n \)

Let \( T' = 2^f T \). Current implementation of SQISign has \( f \approx \lfloor \log_2(p^{1/4}) \rfloor \)
which translates to \( T' \approx p^{3/2+\epsilon} \)

The smaller \( n \) is, we get more guaranteed smoothness from the twin. This comes at a cost of finding larger twins.
Choosing $n$

Let $T' = 2^f T$. Current implementation of SQISign has $f \approx \lfloor \log_2(p^{1/4}) \rfloor$ which translates to $T' \approx p^{3/2+\epsilon}$

The smaller $n$ is, we get more guaranteed smoothness from the twin. This comes at a cost of finding larger twins

$n = 2$: For a smooth twin $(r, r \pm 1)$, let $p = 2r^2 - 1$. Here we have

$$p - 1 = 2(r - 1)(r + 1)$$
Choosing \( n \)

Let \( T' = 2^f T \). Current implementation of SQISign has \( f \approx \left\lfloor \log_2(p^{1/4}) \right\rfloor \) which translates to \( T' \approx p^{3/2+\epsilon} \)

The smaller \( n \) is, we get more guaranteed smoothness from the twin. This comes at a cost of finding larger twins

\( n = 2 \): For a smooth twin \((r, r \pm 1)\), let \( p = 2r^2 - 1 \). Here we have

\[
p - 1 = 2(r - 1)(r + 1)
\]

The amount of guaranteed smoothness from the twin alone is \( p^{3/2} \)
Choosing $n$

Let $T' = 2^f T$. Current implementation of SQISign has $f \approx \lfloor \log_2(p^{1/4}) \rfloor$ which translates to $T' \approx p^{3/2+\epsilon}$

The smaller $n$ is, we get more guaranteed smoothness from the twin. This comes at a cost of finding larger twins

$n = 2$: For a smooth twin $(r, r \pm 1)$, let $p = 2r^2 - 1$. Here we have

$$p - 1 = 2(r - 1)(r + 1)$$

The amount of guaranteed smoothness from the twin alone is $p^{3/2}$

If the power of two in $p^2 - 1$ is less than $\lfloor \log_2(p^{1/4}) \rfloor$ then we have enough smoothness for a SQISign parameter
Choosing $n$

Let $T' = 2^f T$. Current implementation of SQISign has $f \approx \lfloor \log_2(p^{1/4}) \rfloor$ which translates to $T' \approx p^{3/2+\epsilon}$.
Choosing $n$

Let $T' = 2^f T$. Current implementation of SQISign has $f \approx \lfloor \log_2(p^{1/4}) \rfloor$ which translates to $T' \approx p^{3/2 + \epsilon}$

$n = 3$: For a smooth twin $(r, r - 1)$, let $p = 2r^3 - 1$. Here we have

$p - 1 = 2(r - 1)(r^2 + r + 1)$
Choosing \( n \)

Let \( T' = 2^f T \). Current implementation of SQISign has \( f \approx \lfloor \log_2(p^{1/4}) \rfloor \)
which translates to \( T' \approx p^{3/2+\epsilon} \)

\( n = 3 \): For a smooth twin \((r, r - 1)\), let \( p = 2r^3 - 1 \). Here we have

\[
p - 1 = 2(r - 1)(r^2 + r + 1)
\]

The amount of guaranteed smoothness from the twin alone is \( p^{4/3} \)}
Choosing \( n \)

Let \( T' = 2^f T \). Current implementation of SQISign has \( f \approx \lfloor \log_2(p^{1/4}) \rfloor \) which translates to \( T' \approx p^{3/2+\epsilon} \)

\( n = 3 \): For a smooth twin \((r, r - 1)\), let \( p = 2r^3 - 1 \). Here we have

\[
p - 1 = 2(r - 1)(r^2 + r + 1)
\]

The amount of guaranteed smoothness from the twin alone is \( p^{4/3} \)

If the power of two in \( p^2 - 1 \) is less than \( \lfloor \log_2(p^{1/20}) \rfloor \) then we have enough smoothness for a SQISign parameter
Choosing $n$

Let $T' = 2^f T$. Current implementation of SQISign has $f \approx \lfloor \log_2(p^{1/4}) \rfloor$ which translates to $T' \approx p^{3/2+\epsilon}$

$n = 3$: For a smooth twin $(r, r - 1)$, let $p = 2r^3 - 1$. Here we have

$$p - 1 = 2(r - 1)(r^2 + r + 1)$$

The amount of guaranteed smoothness from the twin alone is $p^{4/3}$

If the power of two in $p^2 - 1$ is less than $\lfloor \log_2(p^{1/20}) \rfloor$ then we have enough smoothness for a SQISign parameter

Obtaining a smooth factor of size $\approx p^{3/2}$ would require that us to hope that there is a smooth factor of size $\approx p^{1/6}$ in the factor $r^2 + r + 1$
Choosing \( n \)

Let \( T' = 2^f T \). Current implementation of SQISign has \( f \approx \lceil \log_2(p^{1/4}) \rceil \) which translates to \( T' \approx p^{3/2+\epsilon} \)

**n = 3:** For a smooth twin \((r, r - 1)\), let \( p = 2r^3 - 1 \). Here we have

\[
p - 1 = 2(r - 1)(r^2 + r + 1)
\]

The amount of guaranteed smoothness from the twin alone is \( p^{4/3} \)

If the power of two in \( p^2 - 1 \) is less than \( \lceil \log_2(p^{1/20}) \rceil \) then we have enough smoothness for a SQISign parameter

Obtaining a smooth factor of size \( \approx p^{3/2} \) would require that us to hope that there is a smooth factor of size \( \approx p^{1/6} \) in the factor \( r^2 + r + 1 \)

One can estimate the probability of this happening using a result by Banks and Shaparlinski (2006)
Choosing $n$

Let $T' = 2^f T$. Current implementation of SQISign has $f \approx \lfloor \log_2(p^{1/4}) \rfloor$ which translates to $T' \approx p^{3/2+\epsilon}$.
Choosing \( n \)

Let \( T' = 2^f T \). Current implementation of SQISign has \( f \approx \lfloor \log_2(p^{1/4}) \rfloor \) which translates to \( T' \approx p^{3/2+\epsilon} \)

\( n = 4 \): For a smooth twin \((r, r \pm 1)\), let \( p = 2r^4 - 1 \). Here we have

\[ p - 1 = 2(r - 1)(r + 1)(r^2 + 1) \]
Choosing $n$

Let $T' = 2^f T$. Current implementation of SQISign has $f \approx \lfloor \log_2(p^{1/4}) \rfloor$ which translates to $T' \approx p^{3/2+\epsilon}$

$n = 4$: For a smooth twin $(r, r \pm 1)$, let $p = 2r^4 - 1$. Here we have

$$p - 1 = 2(r - 1)(r + 1)(r^2 + 1)$$

The amount of guaranteed smoothness from the twin alone is $p^{5/4}$
Choosing $n$

Let $T' = 2^f T$. Current implementation of SQISign has $f \approx \lfloor \log_2(p^{1/4}) \rfloor$ which translates to $T' \approx p^{3/2+\epsilon}$

**n = 4**: For a smooth twin $(r, r \pm 1)$, let $p = 2r^4 - 1$. Here we have

$$p - 1 = 2(r - 1)(r + 1)(r^2 + 1)$$

The amount of guaranteed smoothness from the twin alone is $p^{5/4}$

No matter what the power of two is, we have no choice but to check for smooth factors of $(r \mp 1)(r^2 + 1)$
Let $T' = 2^f T$. Current implementation of SQISign has $f \approx \lceil \log_2 (p^{1/4}) \rceil$ which translates to $T' \approx p^{3/2+\epsilon}$

$n = 4$: For a smooth twin $(r, r \pm 1)$, let $p = 2r^4 - 1$. Here we have

$$p - 1 = 2(r - 1)(r + 1)(r^2 + 1)$$

The amount of guaranteed smoothness from the twin alone is $p^{5/4}$.

No matter what the power of two is, we have no choice but to check for smooth factors of $(r \mp 1)(r^2 + 1)$.

Estimating the probability of this is a little non-trivial to do since we are given some “factoring structure”
Choosing $n$

Let $T' = 2^f T$. Current implementation of SQISign has $f \approx \lceil \log_2(p^{1/4}) \rceil$ which translates to $T' \approx p^{3/2+\epsilon}$

$n = 4$: For a smooth twin $(r, r \pm 1)$, let $p = 2r^4 - 1$. Here we have

$$p - 1 = 2(r - 1)(r + 1)(r^2 + 1)$$

The amount of guaranteed smoothness from the twin alone is $p^{5/4}$

No matter what the power of two is, we have no choice but to check for smooth factors of $(r \mp 1)(r^2 + 1)$

Estimating the probability of this is a little non-trivial to do since we are given some “factoring structure”

We give a worst case probability
Choosing $n$

Let $T' = 2^f T$. Current implementation of SQISign has $f \approx \lfloor \log_2(p^{1/4}) \rfloor$ which translates to $T' \approx p^{3/2 + \epsilon}$. 
Choosing $n$

Let $T' = 2^f T$. Current implementation of SQISign has $f \approx \lfloor \log_2(p^{1/4}) \rfloor$ which translates to $T' \approx p^{3/2 + \epsilon}$

The larger $n$ is, the more smoothness we require from the other factor(s)
Choosing $n$

Let $T' = 2^f T$. Current implementation of SQISign has $f \approx \lfloor \log_2(p^{1/4}) \rfloor$ which translates to $T' \approx p^{3/2+\epsilon}$

The larger $n$ is, the more smoothness we require from the other factor(s)

$n = 6$: For a smooth twin $(r, r \pm 1)$, let $p = 2r^6 - 1$. Here we have

$$p - 1 = 2(r - 1)(r + 1)(r^2 - r + 1)(r^2 + r + 1)$$
Let $T' = 2^f T$. Current implementation of SQISign has $f \approx \lfloor \log_2(p^{1/4}) \rfloor$ which translates to $T' \approx p^{3/2 + \epsilon}$

The larger $n$ is, the more smoothness we require from the other factor(s)

$n = 6$: For a smooth twin $(r, r \pm 1)$, let $p = 2r^6 - 1$. Here we have

$$p - 1 = 2(r - 1)(r + 1)(r^2 - r + 1)(r^2 + r + 1)$$

The amount of guaranteed smoothness from the twin alone is $p^{7/6}$
Choosing $n$

Let $T' = 2^f T$. Current implementation of SQISign has $f \approx \lfloor \log_2(p^{1/4}) \rfloor$ which translates to $T' \approx p^{3/2+\epsilon}$

The larger $n$ is, the more smoothness we require from the other factor(s)

$n = 6$: For a smooth twin $(r, r \pm 1)$, let $p = 2r^6 - 1$. Here we have

$$p - 1 = 2(r - 1)(r + 1)(r^2 - r + 1)(r^2 + r + 1)$$

The amount of guaranteed smoothness from the twin alone is $p^{7/6}$

Here we exploit the multiple factors, $(r \mp 1)(r^2 - r + 1)(r^2 + r + 1)$, to give a better chance of finding enough smoothness for SQISign parameters
Table 2: Assuming that \((r, r \pm 1)\) are twin smooth integers and \(p\) has \(\log p\) bits, calculates the probability of having a \(B\)-smooth divisor \(T' | p^2 - 1\) of size \(\approx p^{3/2}\).
Practical SQISign Results
We used \( n = 2, 3, 4 \) to find a collection of 256-bit SQISign friendly primes.

243-bit prime \( p = 2^{r_2} - 1 \) with \( r_2 = 2091023014142971802357816084152713216 \):

\[
p + 1 = 2^{49} \cdot 3^{4} \cdot 7^2 \cdot 11^2 \cdot 31^2 \cdot 41^2 \cdot 47^2 \cdot 67^2 \cdot 151^2 \cdot 173^2 \cdot 193^2 \cdot 223^2 \cdot 307^2 \cdot 317^2 \cdot 463^2 \cdot 887^2,
\]

and

\[
p - 1 = 2 \cdot 5 \cdot 13 \cdot 19 \cdot 29 \cdot 53 \cdot 61 \cdot 113 \cdot 211 \cdot 311 \cdot 337 \cdot 479 \cdot 599 \cdot 691 \cdot 739 \cdot 773 \cdot 811 \cdot 1277 \cdot 9910061678402709963781118882240347.
\]

255-bit prime \( p = 2^{r_3} - 1 \) with \( r_3 = 26606682403634464748953600 \):

\[
p + 1 = 2^{40} \cdot 5^6 \cdot 11^3 \cdot 47^3 \cdot 67^6 \cdot 101^3 \cdot 113^3 \cdot 137^3 \cdot 277^3 \cdot 307^3 \cdot 421^3,
\]

and

\[
p - 1 = 2 \cdot 3^2 \cdot 19^3 \cdot 37 \cdot 59 \cdot 61 \cdot 97 \cdot 181^2 \cdot 197 \cdot 223 \cdot 271 \cdot 281 \cdot 311 \cdot 397 \cdot 547 \cdot R^3.
\]
We used $n = 2, 3, 4$ to find a collection of 256-bit SQISign friendly primes.
NIST-I parameters

We used \( n = 2, 3, 4 \) to find a collection of 256-bit SQISign friendly primes.

243-bit prime \( p = 2r^2 - 1 \) with \( r=2091023014142971802357816084152713216 \):

\[
p + 1 = 2^{49} \cdot 3^4 \cdot 7^2 \cdot 11^2 \cdot 31^2 \cdot 41^2 \cdot 47^2 \cdot 67^2 \cdot 151^2 \cdot 173^2 \cdot 193^2 \cdot 223^2 \\
\quad \cdot 307^2 \cdot 317^2 \cdot 463^2 \cdot 887^2, \text{ and}
\]

\[
p - 1 = 2 \cdot 5 \cdot 13^2 \cdot 19 \cdot 29 \cdot 53 \cdot 61 \cdot 113 \cdot 211 \cdot 311 \cdot 337 \cdot 479 \cdot 599 \cdot 691 \\
\quad \cdot 739 \cdot 773 \cdot 811 \cdot 1277 \cdot 9910061678402709963781118882240347
\]
NIST-I parameters

We used \( n = 2, 3, 4 \) to find a collection of 256-bit SQISign friendly primes

**243-bit prime** \( p = 2r^2 - 1 \) with \( r = 2091023014142971802357816084152713216 \):

\[
p + 1 = 2^{49} \cdot 3^4 \cdot 7^2 \cdot 11^2 \cdot 31^2 \cdot 41^2 \cdot 47^2 \cdot 67^2 \cdot 151^2 \cdot 173^2 \cdot 193^2 \cdot 223^2 \\
\cdot 307^2 \cdot 317^2 \cdot 463^2 \cdot 887^2, \text{ and} \\
p - 1 = 2 \cdot 5 \cdot 13^2 \cdot 19 \cdot 29 \cdot 53 \cdot 61 \cdot 113 \cdot 211 \cdot 311 \cdot 337 \cdot 479 \cdot 599 \cdot 691 \\
\cdot 739 \cdot 773 \cdot 811 \cdot 1277 \cdot 9910061678402709963781118882240347
\]

**255-bit prime** \( p = 2r^3 - 1 \) with \( r = 26606682403634464748953600 \):

\[
p + 1 = 2^{40} \cdot 5^6 \cdot 11^3 \cdot 47^3 \cdot 67^6 \cdot 101^3 \cdot 113^3 \cdot 137^3 \cdot 277^3 \cdot 307^3 \cdot 421^3, \text{ and} \\
p - 1 = 2 \cdot 3^2 \cdot 19^3 \cdot 37 \cdot 59 \cdot 61 \cdot 97 \cdot 181^2 \cdot 197 \cdot 223 \cdot 271 \cdot 281 \cdot 311 \\
\cdot 397 \cdot 547 \cdot R
\]
We used $n = 2, 3, 4$ to find a collection of 256-bit SQISign friendly primes.
We used $n = 2, 3, 4$ to find a collection of 256-bit SQISign friendly primes

253-bit prime $p = 2r^4 - 1$ with $r = 8077251317941145600$:

$$p + 1 = 2^{49} \cdot 5^8 \cdot 13^4 \cdot 41^4 \cdot 71^4 \cdot 113^4 \cdot 181^4 \cdot 223^4 \cdot 457^4,$$

and

$$p - 1 = 2 \cdot 3^2 \cdot 7^5 \cdot 17 \cdot 31 \cdot 53 \cdot 61 \cdot 73 \cdot 83 \cdot 127 \cdot 149 \cdot 233 \cdot 293 \cdot 313 \cdot 347 \cdot 397 \cdot 467 \cdot 479 \cdot R$$
**NIST-I parameters**

We used $n = 2, 3, 4$ to find a collection of 256-bit SQISign friendly primes

253-bit prime $p = 2^r - 1$ with $r = 8077251317941145600$:

\[
p + 1 = 2^{49} \cdot 5^8 \cdot 13^4 \cdot 41^4 \cdot 71^4 \cdot 113^4 \cdot 181^4 \cdot 223^4 \cdot 457^4, \quad \text{and} \\
p - 1 = 2 \cdot 3^2 \cdot 7^5 \cdot 17 \cdot 31 \cdot 53 \cdot 61 \cdot 73 \cdot 83 \cdot 127 \cdot 149 \cdot 233 \cdot 293 \cdot 313 \cdot 347 \cdot 397 \cdot 467 \cdot 479 \cdot R
\]

**Remarks:**

- This prime is out of scope for De Feo, Leroux and Wesolowski to find since they “maximised” the power of two in $p + 1$
- No conclusions should be made about how these primes compare to the state-of-the-art without an implementation
We used $n = 3, 4, 6$ to find a collection of 384-bit SQISign friendly primes.

A 375-bit prime $p = 2^r - 1$ with $r = 12326212283367463507272925184$:

$$p + 1 = 2^{77} \cdot 11^4 \cdot 29^4 \cdot 59^4 \cdot 67^4 \cdot 149^4 \cdot 331^4 \cdot 443^4 \cdot 593^4 \cdot 1091^4 \cdot 1319^4,$$

and

$$p - 1 = 2 \cdot 3 \cdot 5 \cdot 13 \cdot 17 \cdot 31 \cdot 37 \cdot 53 \cdot 83 \cdot 109 \cdot 131 \cdot 241 \cdot 269 \cdot 277 \cdot 283 \cdot 353 \cdot 419 \cdot 499 \cdot 661 \cdot 877 \cdot 1877 \cdot 3709 \cdot 9613 \cdot 44017 \cdot 55967 \cdot R.$$
NIST-III parameters

We used $n = 3, 4, 6$ to find a collection of 384-bit SQISign friendly primes.
NIST-III parameters

We used $n = 3, 4, 6$ to find a collection of 384-bit SQISign friendly primes

375-bit prime $p = 2^r - 1$ with $r = 12326212283367463507272925184$:

$$p + 1 = 2^{77} \cdot 11^4 \cdot 29^4 \cdot 59^4 \cdot 67^4 \cdot 149^4 \cdot 331^4 \cdot 443^4 \cdot 593^4 \cdot 1091^4$$

$$\cdot 1319^4$$

and

$$p - 1 = 2 \cdot 3 \cdot 5 \cdot 13 \cdot 17 \cdot 31 \cdot 37 \cdot 53 \cdot 83 \cdot 109 \cdot 131 \cdot 241 \cdot 269 \cdot 277 \cdot 283$$

$$\cdot 353 \cdot 419 \cdot 499 \cdot 661 \cdot 877 \cdot 1877 \cdot 3709 \cdot 9613 \cdot 44017 \cdot 55967 \cdot R$$
NIST-III parameters

We used \( n = 3, 4, 6 \) to find a collection of 384-bit SQISign friendly primes

375-bit prime \( p = 2r^4 - 1 \) with \( r = 12326212283367463507272925184 \):

\[
p + 1 = 2^{77} \cdot 11^4 \cdot 29^4 \cdot 59^4 \cdot 67^4 \cdot 149^4 \cdot 331^4 \cdot 443^4 \cdot 593^4 \cdot 1091^4 \cdot 1319^4, \text{ and}
\]

\[
p - 1 = 2 \cdot 3 \cdot 5 \cdot 13 \cdot 17 \cdot 31 \cdot 37 \cdot 53 \cdot 83 \cdot 109 \cdot 131 \cdot 241 \cdot 269 \cdot 277 \cdot 283 \cdot 353 \cdot 419 \cdot 499 \cdot 661 \cdot 877 \cdot 1877 \cdot 3709 \cdot 9613 \cdot 44017 \cdot 55967 \cdot R
\]

382-bit prime \( p = 2r^6 - 1 \) with \( r = 11896643388662145024 \):

\[
p + 1 = 2^{79} \cdot 3^6 \cdot 23^{12} \cdot 107^6 \cdot 127^6 \cdot 307^6 \cdot 401^6 \cdot 547^6, \text{ and}
\]

\[
p - 1 = 2 \cdot 5^2 \cdot 7 \cdot 11 \cdot 17 \cdot 19 \cdot 47 \cdot 71 \cdot 79 \cdot 109 \cdot 149 \cdot 229 \cdot 269 \cdot 283 \cdot 349 \cdot 449 \cdot 463 \cdot 1019 \cdot 1033 \cdot 1657 \cdot 2179 \cdot 2293 \cdot 4099 \cdot 5119 \cdot 10243 \cdot R
\]
NIST-V parameters

We used $n = 499$ to find a collection of 512-bit SQISign friendly primes $p = 2^{r_6} - 1$ with $r_6 = 9469787780580604464332800$:

$$p + 1 = 2^{109} \cdot 5^{12} \cdot 7^{12} \cdot 13^6 \cdot 61^6 \cdot 179^6 \cdot 281^6 \cdot 379^6 \cdot 1367^6 \cdot 1427^6,$$

and $p - 1 = 2^85 \cdot 17^12 \cdot 37^6 \cdot 59^6 \cdot 97^6 \cdot 233^6 \cdot 31^12 \cdot 911^6 \cdot 1297^6$.

A 508-bit prime $p = 2^{r_6} - 1$ with $r_6 = 26697973900446483680608256$:

$$p + 1 = 2^{85} \cdot 17^{12} \cdot 37^6 \cdot 59^6 \cdot 97^6 \cdot 233^6 \cdot 31^{12} \cdot 911^6 \cdot 1297^6,$$

and $p - 1 = 2^8 \cdot 3^2 \cdot 5 \cdot 7 \cdot 11^2 \cdot 23^2 \cdot 29 \cdot 127 \cdot 163 \cdot 173 \cdot 191 \cdot 193 \cdot 211 \cdot 277 \cdot 347 \cdot 617 \cdot 661 \cdot 761 \cdot 1039 \cdot 4637 \cdot 5821 \cdot 15649 \cdot 19139 \cdot 143443 \cdot 150151$.
We used $n = 4, 6$ to find a collection of 512-bit SQISign friendly primes.
We used \( n = 4, 6 \) to find a collection of 512-bit SQISign friendly primes.

499-bit prime \( p = 2^r - 1 \) with \( r = 9469787780580604464332800 \):

\[
\begin{align*}
p + 1 &= 2^{109} \cdot 5^{12} \cdot 7^{12} \cdot 13^6 \cdot 61^6 \cdot 179^6 \cdot 281^6 \cdot 379^6 \cdot 1367^6 \cdot 1427^6, \\
p - 1 &= 2 \cdot 3^3 \cdot 19 \cdot 23^3 \cdot 31 \cdot 43^2 \cdot 73 \cdot 139 \cdot 337 \cdot 461 \cdot 641 \cdot 971 \cdot 1069 \\
&\quad \cdot 1097 \cdot 5843 \cdot 12841 \cdot 23671 \cdot 39667 \cdot 51193 \cdot 75223 \cdot 459317 \\
&\quad \cdot 703981 \cdot R
\end{align*}
\]
We used $n = 4, 6$ to find a collection of 512-bit SQISign friendly primes

499-bit prime $p = 2r^6 - 1$ with $r = 9469787780580604464332800$:

\[
p + 1 = 2^{109} \cdot 5^{12} \cdot 7^{12} \cdot 13^6 \cdot 61^6 \cdot 179^6 \cdot 281^6 \cdot 379^6 \cdot 1367^6 \cdot 1427^6, \text{ and}
\]
\[
p - 1 = 2 \cdot 3^3 \cdot 19 \cdot 23^3 \cdot 31 \cdot 43^2 \cdot 73 \cdot 139 \cdot 337 \cdot 461 \cdot 641 \cdot 971 \cdot 1069 \\
\cdot 1097 \cdot 5843 \cdot 12841 \cdot 23671 \cdot 39667 \cdot 51193 \cdot 75223 \cdot 459317 \\
\cdot 703981 \cdot R
\]

508-bit prime $p = 2r^6 - 1$ with $r = 26697973900446483680608256$:

\[
p + 1 = 2^{85} \cdot 17^{12} \cdot 37^6 \cdot 59^6 \cdot 97^6 \cdot 233^6 \cdot 311^{12} \cdot 911^6 \cdot 1297^6, \text{ and}
\]
\[
p - 1 = 2 \cdot 3^2 \cdot 5 \cdot 7 \cdot 11^2 \cdot 23^2 \cdot 29 \cdot 127 \cdot 163 \cdot 173 \cdot 191 \cdot 193 \cdot 211 \cdot 277 \\
\cdot 347 \cdot 617 \cdot 661 \cdot 761 \cdot 1039 \cdot 4637 \cdot 5821 \cdot 15649 \cdot 19139 \\
\cdot 143443 \cdot 150151 \cdot R
\]
| NIST security level | $n$ | $r$ | $|\log_2(p)|$ | $f$ | $B$ | $\sqrt{B}/f$ | $\log_2(T)$ |
|---------------------|-----|-----|--------------|-----|-----|-------------|------------|
| **NIST-I**          | 2   | 1211460311716772790566574529001291776 2091023014142971802357816084152713216 | 241 | 49 | 1091 | 0.67 | 1.28 |
|                     |     | 243 | 49 | 887 | 0.61 | 1.28 |
|                     | 3   | 347427281678967297357824 10227318375788227199589376 2161173603326087887680000 2046144912550374748856320 26606682403634464748953600 | 246 | 43 | 547 | 0.54 | 1.29 |
|                     |     | 251 | 31 | 383 | 0.63 | 1.31 |
|                     |     | 254 | 31 | 421 | 0.66 | 1.28 |
|                     |     | 254 | 46 | 523 | 0.50 | 1.26 |
|                     |     | 255 | 40 | 547 | 0.58 | 1.28 |
|                     | 4   | 1466873880764125184 8077251317941145600 34848218231355211776* | 243 | 49 | 701 | 0.54 | 1.28 |
|                     |     | 253 | 49 | 479 | 0.45 | 1.30 |
|                     |     | 261 | 77 | 2311 | 0.62 | 1.30 |
| **NIST-III**        | 3   | 1374002035005713149550405343373848576 | 362 | 37 | 1277 | 0.97 | 1.25 |
|                     | 4   | 5139734876262390964070873088 12326212233637463507272925184 1808075498029542546023326720 2746440030914679028660255744 | 370 | 45 | 11789 | 2.41 | 1.26 |
|                     |     | 375 | 77 | 55967 | 3.07 | 1.31 |
|                     |     | 377 | 61 | 95569 | 5.07 | 1.26 |
|                     |     | 379 | 41 | 13127 | 2.79 | 1.29 |
|                     | 6   | 2628583629218279424 541769011877495584 1189664338662145024 | 369 | 73 | 13219 | 1.58 | 1.27 |
|                     |     | 375 | 79 | 58153 | 3.05 | 1.27 |
|                     |     | 382 | 79 | 10243 | 1.28 | 1.30 |
| **NIST-V**          | 4   | 114216781548581709439512875801279791104* 123794274387474298912742543192242587136* | 507 | 65 | 75941 | 4.24 | 1.26 |
|                     |     | 508 | 41 | 15263 | 3.01 | 1.29 |
|                     | 6   | 94697877805806446432800 1223346860574068007808000 2669797300446483686068256 31929740427944870006521856 413402482000819056793600 | 499 | 109 | 703981 | 7.70 | 1.25 |
|                     |     | 502 | 73 | 376963 | 8.41 | 1.28 |
|                     |     | 508 | 85 | 150151 | 4.56 | 1.26 |
|                     |     | 510 | 91 | 550657 | 8.15 | 1.25 |
|                     |     | 512 | 67 | 224911 | 7.08 | 1.28 |

**Table 3:** A table of SQISign parameters $p = p_n(r)$ found using twin-smooth integers $(r, r \pm 1)$ at each security level. The $f$ is the power of two dividing $(p^2 - 1)/2$ and $B$ is the smoothness bound of the odd cofactor $T \approx p^{5/4+\epsilon}$. The $r$ marked with an asterisk correspond to primes $p$ not found using the CHM machinery.
Smooth Twins from XGCD over Polynomial Rings
Probabilistic methods for finding smooth twins
Probabilistic methods for finding smooth twins

Integer world
Naïve method
XGCD/CRT

Polynomial world
\[ p_n(x) = 2x^n - 1 \]
Ideal PTE solutions

Is there something that can bind these methods together?
Probabilistic methods for finding smooth twins

Is there something that can bind these methods together?

$p_n(x) = 2x^n - 1$
Ideal PTE solutions

Naïve method
XGCD/CRT

Integer world
Polynomial world

37
Smooth Twins from XGCD over $\mathbb{Q}[x]$
Smooth Twins from XGCD over $\mathbb{Q}[x]$

Choose two polynomials, $F, G \in \mathbb{Z}[x]$, that split completely into linear factors and the number of distinct roots of $F \cdot G$ is small\textsuperscript{5}

\textsuperscript{5}These points are not strictly necessary but they help the smoothness probabilities
Choose two polynomials, $F, G \in \mathbb{Z}[x]$, that split completely into linear factors and the number of distinct roots of $F \cdot G$ is small\(^5\)

Use the XGCD algorithm over $\mathbb{Q}[x]$ to find two polynomials $S, T \in \mathbb{Q}[x]$ such that

$$FS + GT \equiv 1$$

\(^5\)These points are not strictly necessary but they help the smoothness probabilities
Choose two polynomials, $F, G \in \mathbb{Z}[x]$, that split completely into linear factors and the number of distinct roots of $F \cdot G$ is small\(^5\)

Use the XGCD algorithm over $\mathbb{Q}[x]$ to find two polynomials $S, T \in \mathbb{Q}[x]$ such that

$$FS + GT \equiv 1$$

Then the polynomials $\hat{F} := F \cdot S$ and $\hat{G} := -G \cdot T$ differ by 1

\(^5\)These points are not strictly necessary but they help the smoothness probabilities
Smooth Twins from XGCD over $\mathbb{Q}[x]$

Choose two polynomials, $F, G \in \mathbb{Z}[x]$, that split completely into linear factors and the number of distinct roots of $F \cdot G$ is small\(^5\)

Use the XGCD algorithm over $\mathbb{Q}[x]$ to find two polynomials $S, T \in \mathbb{Q}[x]$ such that

$$FS + GT \equiv 1$$

Then the polynomials $\hat{F} := F \cdot S$ and $\hat{G} := -G \cdot T$ differ by 1

For simplicity, assume that $\hat{F}$ and $\hat{G}$ have a positive leading coefficient and that $S, T \in \mathbb{Z}[x]$

\(^5\)These points are not strictly necessary but they help the smoothness probabilities
Choose two polynomials, $F, G \in \mathbb{Z}[x]$, that split completely into linear factors and the number of distinct roots of $F \cdot G$ is small\(^5\).

Use the XGCD algorithm over $\mathbb{Q}[x]$ to find two polynomials $S, T \in \mathbb{Q}[x]$ such that

$$FS + GT \equiv 1$$

Then the polynomials $\hat{F} := F \cdot S$ and $\hat{G} := -G \cdot T$ differ by 1.

For simplicity, assume that $\hat{F}$ and $\hat{G}$ have a positive leading coefficient and that $S, T \in \mathbb{Z}[x]$.

Sieve an interval of integers, $r$, such that $r - a$ is smooth for each root, $a$, in $F \cdot G$.

\(^5\)These points are not strictly necessary but they help the smoothness probabilities.
Smooth Twins from XGCD over $\mathbb{Q}[x]$

Choose two polynomials, $F, G \in \mathbb{Z}[x]$, that split completely into linear factors and the number of distinct roots of $F \cdot G$ is small\(^5\)

Use the XGCD algorithm over $\mathbb{Q}[x]$ to find two polynomials $S, T \in \mathbb{Q}[x]$ such that

$$FS + GT \equiv 1$$

Then the polynomials $\hat{F} := F \cdot S$ and $\hat{G} := -G \cdot T$ differ by 1

For simplicity, assume that $\hat{F}$ and $\hat{G}$ have a positive leading coefficient and that $S, T \in \mathbb{Z}[x]$

Sieve an interval of integers, $r$, such that $r - a$ is smooth for each root, $a$, in $F \cdot G$

Then $(\hat{F}(r), \hat{G}(r))$ generates a smooth twin if and only if $S(r)T(r)$ is smooth

\(^5\)These points are not strictly necessary but they help the smoothness probabilities
Realising the generalisation

This naturally generalises the integer-based XGCD method but also generalises the polynomial techniques: We can reverse the latter remark - i.e. use XGCD over $\mathbb{Q}[x]$ as a tool to find ideal PTE solutions. In fact, we were able to find a completely new class of ideal size 4 PTE solutions that haven't appeared in the literature or any known database.
Realising the generalisation

This naturally generalises the integer-based XGCD method but also generalises the polynomial techniques:
Realising the generalisation

This naturally generalises the integer-based XGCD method but also generalises the polynomial techniques:

- Computing the XGCD of $F(x) = x^n$ and $G(x) = x - 1$ results in the polynomials

  \[ S(x) = 1, \quad \text{and} \quad T(x) = -x^{n-1} - \cdots - x - 1 \]

  Hence we get $\hat{F}(x) = x^n$ and $\hat{G}(x) = x^n - 1$
Realising the generalisation

This naturally generalises the integer-based XGCD method but also generalises the polynomial techniques:

- Computing the XGCD of $F(x) = x^n$ and $G(x) = x - 1$ results in the polynomials
  
  $$S(x) = 1, \text{ and } T(x) = -x^{n-1} - \cdots - x - 1$$

  Hence we get $\hat{F}(x) = x^n$ and $\hat{G}(x) = x^n - 1$

- The generalisation of the method using ideal PTE solutions is technical but is of a similar vein
Realising the generalisation

This naturally generalises the integer-based XGCD method but also generalises the polynomial techniques:

- Computing the XGCD of \( F(x) = x^n \) and \( G(x) = x - 1 \) results in the polynomials

  \[
  S(x) = 1, \quad \text{and} \quad T(x) = -x^{n-1} - \cdots - x - 1
  \]

  Hence we get \( \hat{F}(x) = x^n \) and \( \hat{G}(x) = x^n - 1 \)

- The generalisation of the method using ideal PTE solutions is technical but is of a similar vein

We can reverse the latter remark - i.e. use XGCD over \( \mathbb{Q}[x] \) as a tool to find ideal PTE solutions
Realising the generalisation

This naturally generalises the integer-based XGCD method but also generalises the polynomial techniques:

- Computing the XGCD of \( F(x) = x^n \) and \( G(x) = x - 1 \) results in the polynomials

\[
S(x) = 1, \quad \text{and} \quad T(x) = -x^{n-1} - \cdots - x - 1
\]

Hence we get \( \hat{F}(x) = x^n \) and \( \hat{G}(x) = x^n - 1 \)

- The generalisation of the method using ideal PTE solutions is technical but is of a similar vein

We can reverse the latter remark - i.e. use XGCD over \( \mathbb{Q}[x] \) as a tool to find ideal PTE solutions

In fact, we were able to find a completely new class of ideal size 4 PTE solutions that haven’t appeared in the literature or any known database
### Table 4: List of all inequivalent and normalised sized 4 ideal PTE solutions of the form \([0, a, a, c] =_3 [b, b, d, e]\) with \(0 < b < a < 50000\) and \(c, d, e > 0\).

| a   | b   | c   | d   | e   | a   | b   | c   | d   | e   | a   | b   | c   | d   | e   | a   | b   | c   | d   | e   |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 17  | 5   | 35  | 27  | 32  | 6620| 1940| 13289| 10985| 11664| 22572| 6660| 47545| 35152| 44217|
| 86  | 26  | 221 | 125 | 216 | 6830| 2210| 53261| 9261 | 53240| 22715| 6755| 50759| 34295| 48384|
| 171 | 51  | 391 | 256 | 375 | 7398| 2250| 20125| 10648| 19773| 23579| 7619 | 176039| 32000| 175959|
| 243 | 75  | 775 | 343 | 768 | 7749| 2289| 16459| 12000| 15379| 26010| 8070 | 88501| 36501| 87880|
| 524 | 164 | 2009| 729 | 2000| 8021| 2561| 43931| 10976| 43875| 26672| 8720 | 314465| 35937| 314432|
| 594 | 174 | 1189| 1000| 1029| 8987| 2915| 76055| 12167| 76032| 28170| 8790 | 103429| 39304| 102885|
| 605 | 185 | 1739| 864 | 1715| 10269|3129| 28459| 14739| 28000| 29358| 8610 | 59245| 48013| 52728|
| 965 | 305 | 4331| 1331| 4320| 11556|3756| 105481| 15625| 105456| 31160| 9320 | 72929| 46305| 70304|
| 1463| 455 | 5135| 2048| 5103| 12015|3855| 73759| 16384| 73695| 31437| 10185| 255595| 42592| 255507|
| 1602| 510 | 8245| 2197| 8232| 12386|3806| 37541| 17576| 37125| 31841| 10421| 396611| 42875| 396576|
| 1790| 530 | 3869| 2744| 3645| 13076|3836| 26441| 21296| 23625| 33561| 10461| 121411| 46875| 120736|
| 2471| 791 | 14351|3375| 14336| 14472|4440| 43105| 20577| 42592| 33885| 9945 | 68731| 54880| 61731|
| 2628| 780 | 5785| 3993| 5488| 14573|4745| 142715| 19683| 142688| 34047| 10335| 90895| 49152| 89167|
| 2889| 909 | 12019|4000| 11979| 15930|4710| 34069| 24565| 31944| 35684| 10604| 79289| 54000| 75449|
| 3608| 1160| 23345|4913| 23328| 17153|5525| 116675| 23328| 116603| 37638| 12330| 493885| 50653| 493848|
| 3735| 1095| 7519| 6144| 6655| 18074|5894| 189029| 24389| 189000| 39542| 12410| 158045| 54872| 157437|
| 3962| 1190| 9605| 5832| 9317| 19214|5954| 64349| 27000| 63869| 40871| 13271| 359471| 55296| 359375|
| 4455| 1335| 10591|6591| 10240| 20195|5915| 40391| 34391| 34560| 41445| 12465| 101659| 60835| 98784|
| 5027| 1595| 24215|6912| 24167| 22095|7215| 245791| 29791| 245760| 44099| 14459| 608039| 59319| 608000|

New ideal PTE solutions of size 4
Strategy for finding SQISign primes

The idea is to replace the polynomials $p_n(x)$ with other polynomials $p_i, j(x)$ such that $x^i(x+1)^j | p_{2i, j}(x) - 1$, with $i, j \geq 2$, $i \neq j$ and $\deg(p_{i, j}(x)) < i + j$.

To do this, we compute the XGCD of $F_i(x) = x^i$ and $G_j(x) = (x+1)^j$, which gives us $S_{i, j}(x) = \sum_{k=0}^{i+j-1}(i+k-1)^k(x+1)^k$ and $T_{i, j}(x) = \sum_{k=0}^{i+j-1}(-1)^k(j+k-1)^k x^k$. 

41
Strategy for finding SQISign primes

The idea is to replace the polynomials \( p_n(x) \) with other polynomials \( p_{i,j}(x) \) such that

\[
x^i(x + 1)^j \mid p_{i,j}^2(x) - 1, \quad \text{with } i, j \geq 2, i \neq j \text{ and } \deg(p_{i,j}) < i + j
\]
Strategy for finding SQISign primes

The idea is to replace the polynomials $p_n(x)$ with other polynomials $p_{i,j}(x)$ such that

$$x^i(x + 1)^j \mid p_{i,j}^2(x) - 1, \quad \text{with } i, j \geq 2, i \neq j \text{ and } \deg(p_{i,j}) < i + j$$

To do this, we compute the XGCD of $F_i(x) = x^i$ and $G_j(x) = (x + 1)^j$, which gives us

$$S_{i,j}(x) = (-1)^i \sum_{k=0}^{j-1} \binom{i + k - 1}{k} (x + 1)^k$$

$$T_{i,j}(x) = \sum_{k=0}^{i-1} (-1)^k \binom{j + k - 1}{k} x^k$$
Strategy for finding SQISign primes

\[ S_{i,j}(x) = (-1)^i \sum_{k=0}^{j-1} \binom{i + k - 1}{k} (x + 1)^k \]

\[ T_{i,j}(x) = \sum_{k=0}^{i-1} (-1)^k \binom{j + k - 1}{k} x^k \]
Strategy for finding SQISign primes

\[ S_{i,j}(x) = (-1)^i \sum_{k=0}^{j-1} \binom{i+k-1}{k} (x+1)^k \]

\[ T_{i,j}(x) = \sum_{k=0}^{i-1} (-1)^k \binom{j+k-1}{k} x^k \]

Then we set

\[ p_{i,j}(x) := (-1)^i \left( x^i S_{i,j}(x) - (x+1)^j T_{i,j}(x) \right) \]

\[ = (-1)^i \left( 2x^i S_{i,j}(x) - 1 \right) \]

\[ = (-1)^{i+1} \left( 2(x+1)^j T_{i,j}(x) + 1 \right) \]
Strategy for finding SQISign primes

\[ S_{i,j}(x) = (-1)^i \sum_{k=0}^{j-1} \binom{i + k - 1}{k} (x + 1)^k \]

\[ T_{i,j}(x) = \sum_{k=0}^{i-1} (-1)^k \binom{j + k - 1}{k} x^k \]

Then we set

\[ p_{i,j}(x) := \begin{cases} (-1)^i \left( x^i S_{i,j}(x) - (x + 1)^j T_{i,j}(x) \right) \\ = (-1)^i \left( 2x^i S_{i,j}(x) - 1 \right) \\ = (-1)^{i+1} \left( 2(x + 1)^j T_{i,j}(x) + 1 \right) \end{cases} \]

Note that \( \deg(p_{i,j}) = i + j - 1 < i + j \) and, by the uniqueness of XGCD, no other polynomials exists whose degree is smaller than this one.
Strategy for finding SQISign primes

For instance when $i, j \in \{2, 3\}$ with $i \neq j$, we have

$$p_{2,3}(x) = 6x^4 + 16x^3 + 12x^2 - 1$$

$$p_{3,2}(x) = 6x^4 + 8x^3 + 1$$

We can adopt the same strategy as before, namely take a smooth twin $(r, r+1)$ and compute the evaluation $p_{i,j}(r)$ and see whether it is a suitable SQISign parameter. We limit ourselves to small $i, j \geq 2$ since the polynomials $S_{i,j}, T_{i,j}$ are irreducible for small $i, j \geq 2$.

Moreover, we conjecture that these polynomials are irreducible for all $i, j \geq 2$. 
For instance when $i, j \in \{2, 3\}$ with $i \neq j$, we have

$$p_{2,3}(x) = 6x^4 + 16x^3 + 12x^2 - 1$$
$$p_{3,2}(x) = 6x^4 + 8x^3 + 1$$
Strategy for finding SQISign primes

For instance when $i, j \in \{2, 3\}$ with $i \neq j$, we have

$$p_{2,3}(x) = 6x^4 + 16x^3 + 12x^2 - 1$$
$$p_{3,2}(x) = 6x^4 + 8x^3 + 1$$

We can adopt the same strategy as before, namely take a smooth twin $(r, r + 1)$ and compute the evaluation

$$p = p_{i,j}(r)$$

and see whether it is a suitable SQISign parameter.
Strategy for finding SQISign primes

For instance when $i, j \in \{2, 3\}$ with $i \neq j$, we have

$$p_{2,3}(x) = 6x^4 + 16x^3 + 12x^2 - 1$$
$$p_{3,2}(x) = 6x^4 + 8x^3 + 1$$

We can adopt the same strategy as before, namely take a smooth twin $(r, r + 1)$ and compute the evaluation

$$p = p_{i,j}(r)$$

and see whether it is a suitable SQISign parameter.

We limit ourselves to small $i, j \geq 2$ since the polynomials $S_{i,j}$, $T_{i,j}$ are irreducible\(^6\) for small $i, j \geq 2$

\(^6\)Moreover, we conjecture that these polynomials are irreducible for all $i, j \geq 2$
255-bit prime $p = p_{3,2}(r)$ with $r = 5964933197580566528$:

$$p + 1 = 2 \cdot 3^5 \cdot 19 \cdot 31^2 \cdot 37^2 \cdot 67 \cdot 83^2 \cdot 89^2 \cdot 113^2 \cdot 157^4 \cdot 173^2 \cdot 233 \cdot 487^2 \cdot 641 \cdot R, \text{ and}$$

$$p - 1 = 2^{48} \cdot 11^3 \cdot 29^2 \cdot 47^3 \cdot 53^3 \cdot 79 \cdot 131^3 \cdot 323^3 \cdot 349^3 \cdot 439^3 \cdot 691 \cdot R'$$

382-bit prime $p = p_{3,2}(r)$ with $r = 24412952691406071260714369024$:

$$p + 1 = 2 \cdot 3^7 \cdot 7^{10} \cdot 19^6 \cdot 67^2 \cdot 131 \cdot 241^2 \cdot 313^2 \cdot 379^2 \cdot 641 \cdot 83^2 \cdot 1103^2 \cdot 1117^2 \cdot 2689 \cdot 11177 \cdot R, \text{ and}$$

$$p - 1 = 2^{66} \cdot 5 \cdot 13^3 \cdot 17^3 \cdot 23^3 \cdot 41^3 \cdot 59^3 \cdot 61^3 \cdot 83^6 \cdot 127 \cdot 329 \cdot 491^3 \cdot 787^3 \cdot 983 \cdot 1549^3 \cdot R'$$
Concluding Remarks
Concluding Remarks

We have explored novel methods for finding these twins:

- In isogeny-based cryptography (CHM);
- Within their own right (XGCD over $\mathbb{Q}[x]$)

CHM Polynomials $p_n, p_i, j \rightarrow \text{SQISign Parameters}$

The general strategies deployed to find these primes can be applied in future applications.

Smooth sandwiches is the “new normal” for isogeny-based cryptography
We have explored novel methods for finding these twins:

- In isogeny-based cryptography (CHM);
- Within their own right (XGCD over \(\mathbb{Q}[x]\))
Concluding Remarks

We have explored novel methods for finding these twins:

- In isogeny-based cryptography (CHM);
- Within their own right (XGCD over \( \mathbb{Q}[x] \))
Concluding Remarks

We have explored *novel* methods for finding these twins:

- In isogeny-based cryptography (CHM);
- Within their own right (XGCD over $\mathbb{Q}[x]$)

The general strategies deployed to find these primes can be applied in future applications.
Merci pour votre attention

Questions?

ia.cr/2022/1439