Cryptographic Smooth Neighbors

Giacomo Bruno [1] Maria Corte-Real Santos [2] Craig Costello [3] Jonathan Komada Eriksen [4]
Michael Meyer [5] Michael Naehrig [3] *Bruno Sterner* [6]

[1]IKARUS Security Software, [2]University College London, [3]Microsoft Research, [4]Norwegian University of Science and Technology, [5]University of Regensburg, [6]*University of Surrey*

Talk at Asiacrypt 2023

Motivation

Consecutive Integers

Consecutive Integers

⋮

1524094380979073513389817849
1524094380979073513389817850
1524094380979073513389817851
1524094380979073513389817852
1524094380979073513389817853
1524094380979073513389817854
1524094380979073513389817855
1524094380979073513389817856
1524094380979073513389817857
1524094380979073513389817858
1524094380979073513389817859
1524094380979073513389817860
1524094380979073513389817861
1524094380979073513389817862

⋮

Consecutive Integers

$$
\begin{array}{c}
\vdots \qquad\qquad \vdots \qquad\qquad \vdots \\
152409438097907351338981 7849 = 7 \cdot 990044713531 \cdot 219917106505997 \\
1524094380979073513389817850 = 2 \cdot 3 \cdot 5^2 \cdot 21490061513 \cdot 472805961973663 \\
1524094380979073513389817851 = 11^2 \cdot 5009 \cdot 131009 \cdot 42319423 \cdot 453559837 \\
1524094380979073513389817852 = 2^2 \cdot 433 \cdot 879962113729257224820911 \\
1524094380979073513389817853 = 3 \cdot 10211 \cdot 49753350340452241484341 \\
1524094380979073513389817854 = 2 \cdot 1697 \cdot 1017539 \cdot 441315275501735669 \\
1524094380979073513389817855 = 5 \cdot 17 \cdot 19^2 \cdot 31^2 \cdot 37^2 \cdot 53 \cdot 79^2 \cdot 139^2 \cdot 157 \cdot 191 \cdot 197 \\
1524094380979073513389817856 = 2^{19} \cdot 3^2 \cdot 7 \cdot 13^2 \cdot 23 \cdot 41 \cdot 43 \cdot 103 \cdot 109 \cdot 113 \cdot 149 \cdot 179 \cdot 199 \\
1524094380979073513389817857 = 1524094380979073513389817857 \\
1524094380979073513389817858 = 2 \cdot 1427 \cdot 6053 \cdot 138270731 \cdot 638053301789 \\
1524094380979073513389817859 = 3 \cdot 71 \cdot 74129557 \cdot 96525231907873499 \\
1524094380979073513389817860 = 2^2 \cdot 5 \cdot 181 \cdot 421020547231788263367353 \\
1524094380979073513389817861 = 101 \cdot 1987 \cdot 42437 \cdot 1097461 \cdot 163064284979 \\
1524094380979073513389817862 = 2 \cdot 3 \cdot 11 \cdot 67 \cdot 7823 \cdot 110923 \cdot 397189890942349 \\
\vdots \qquad\qquad \vdots \qquad\qquad \vdots
\end{array}
$$

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, i.e. q prime and $q \mid r(r+1) \implies q \leq B$.

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, i.e. q prime and $q \mid r(r+1) \implies q \leq B$.

For instance, the following are 7-smooth twins:

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

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, i.e. q prime and $q \mid r(r+1) \implies q \leq B$.

For instance, the following are 7-smooth twins:

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

For a fixed B, Størmer (1897) proved the set of B-smooth twins is *finite!*

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, i.e. q prime and $q \mid r(r+1) \implies q \leq B$.

For instance, the following are 7-smooth twins:

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

For a fixed B, Størmer (1897) proved the set of B-smooth twins is *finite!*

B	2	3	5	7	11	13	$\cdots$	40	$\cdots$	100	$\cdots$	113	$\cdots\cdots$	200
# B-smooth twins	1	4	10	23	40	68	$\cdots$	653	$\cdots$	13,374	$\cdots$	33,233	$\cdots\cdots$	$\geq$ 348,840

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, i.e. q prime and $q \mid r(r+1) \implies q \leq B$.

For instance, the following are 7-smooth twins:

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

For a fixed B, Størmer (1897) proved the set of B-smooth twins is *finite!*

B	2	3	5	7	11	13	$\cdots$	40	$\cdots$	100	$\cdots$	113	$\cdots\cdots$	200
# B-smooth twins	1	4	10	23	40	68	$\cdots$	653	$\cdots$	13,374	$\cdots$	33,233	$\cdots\cdots$	$\geq$ 348,840

Many applications: isogeny-based cryptography (e.g. SQISign)

Twin smooth integers

Twin smooth integers

Question: Can we find the largest B-smooth twins?

Twin smooth integers

Question: Can we find the largest B-smooth twins?

- Pell equations
- CHM algorithm
- PTE sieve

Twin smooth integers

Question: Can we find the largest B-smooth twins?

- Pell equations
- **CHM algorithm**
- PTE sieve

Twin smooth integers

Question: Can we find the largest B-smooth twins?

- Pell equations
- **CHM algorithm**
- PTE sieve

We revisit the CHM algorithm to find record size twin smooth integers and use these twins to find new parameters for the isogeny-based cryptosystem SQISign

Outline

CHM algorithm

Isogeny-based protocols

New SQISign parameters

CHM algorithm

CHM algorithm

CHM algorithm

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

- $S^{(0)} = \{1, 2, \cdots, B-1\}$ – representing B-smooth twins $(1, 2), (2, 3), \cdots, (B-1, B)$

CHM algorithm

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

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

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

CHM algorithm

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

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

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

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

CHM algorithm

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

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

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

- $S^{(1)} := S^{(0)} \cup \{\text{new solutions } t : t' = t+1\}$
- Repeat this with $S^{(1)}$ instead of $S^{(0)}$

CHM algorithm

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

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

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

- $S^{(1)} := S^{(0)} \cup \{\text{new solutions } t : t' = t + 1\}$
- Repeat this with $S^{(1)}$ instead of $S^{(0)}$
- Algorithm terminates when $S^{(d+1)} = S^{(d)}$ for some d

CHM algorithm

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

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

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

- $S^{(1)} := S^{(0)} \cup \{\text{new solutions } t : t' = t+1\}$
- Repeat this with $S^{(1)}$ instead of $S^{(0)}$
- Algorithm terminates when $S^{(d+1)} = S^{(d)}$ for some d

When $t' = t + 1$, this equivalent to $t = \frac{r(s+1)}{s-r}$ being an integer

CHM in action

CHM in action

We illustrate the algorithm for $B = 5$. The starting set is

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

CHM in action

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 $(t, t + 1)$

CHM in action

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 $(t, t+1)$

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

CHM in action

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 $(t, t+1)$

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

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

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

CHM in action

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 $(t, t+1)$

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

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

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

The second and third iterations find two and one new twins (resp.)

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

CHM in action

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

CHM in action

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

The fourth iteration does not produce any new numbers

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

CHM in action

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

The fourth iteration does not produce any new numbers

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

This is exactly all 5-smooth twins!

CHM in action

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

The fourth iteration does not produce any new numbers

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

This is exactly all 5-smooth twins!

Warning: In general this method does not guarantee to produce all B-smooth twins

CHM in action

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

The fourth iteration does not produce any new numbers

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

This is exactly all 5-smooth twins!

Warning: In general this method does not guarantee to produce all B-smooth twins

$B = 7:$ All 7-smooth twins are found expect one

CHM in action

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

The fourth iteration does not produce any new numbers

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

This is exactly all 5-smooth twins!

Warning: In general this method does not guarantee to produce all B-smooth twins

$B = 7$: All 7-smooth twins are found expect one — $(4374, 4375)$ is not found ✗

CHM in action

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

The fourth iteration does not produce any new numbers

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

This is exactly all 5-smooth twins!

Warning: In general this method does not guarantee to produce all B-smooth twins

$B = 7$: All 7-smooth twins are found expect one — $(4374, 4375)$ is not found ✗

$11 \leq B < 41$: Finds all B-smooth twins

CHM in action

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

The fourth iteration does not produce any new numbers

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

This is exactly all 5-smooth twins!

Warning: In general this method does not guarantee to produce all B-smooth twins

$B = 7$: All 7-smooth twins are found expect one — $(4374, 4375)$ is not found ✗

$11 \leq B < 41$: Finds all B-smooth twins

$B \geq 41$: Conjecturally finds *almost all* B-smooth twins

CHM in action

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

The fourth iteration does not produce any new numbers

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

This is exactly all 5-smooth twins!

Warning: In general this method does not guarantee to produce all B-smooth twins

$B = 7$: All 7-smooth twins are found expect one — $(4374, 4375)$ is not found ✗

$11 \leq B < 41$: Finds all B-smooth twins

$B \geq 41$: Conjecturally finds *almost all* B-smooth twins

$B = 100$: Original authors found all except 41 B-smooth twins

CHM in action

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

The fourth iteration does not produce any new numbers

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

This is exactly all 5-smooth twins!

Warning: In general this method does not guarantee to produce all B-smooth twins

$B = 7$: All 7-smooth twins are found expect one — $(4374, 4375)$ is not found ✗

$11 \leq B < 41$: Finds all B-smooth twins

$B \geq 41$: Conjecturally finds *almost all* B-smooth twins

$B = 100$: Original authors found all except 41 B-smooth twins

$B = 200$: They found 346,192 such twins – which took them 2 weeks to run

Our experiments

Our experiments

We optimised the CHM algorithm and ran $B = 200$ much faster[1]!

[1] The computation now takes a mere 7 minutes to run on a laptop

Our experiments

We optimised the CHM algorithm and ran $B = 200$ much faster[1]!

Subsequently we ran it fully for $B = 547$ and found 82,026,426 twins – the largest twin found was the following 122-bit twin

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

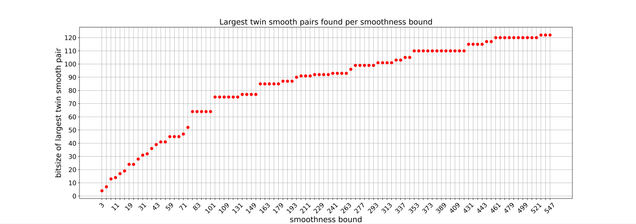
[1]The computation now takes a mere 7 minutes to run on a laptop

Our experiments

We optimised the CHM algorithm and ran $B = 200$ much faster[1]!

Subsequently we ran it fully for $B = 547$ and found 82,026,426 twins – the largest twin found was the following 122-bit twin

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



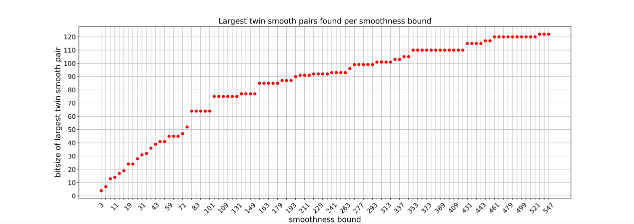
[1] The computation now takes a mere 7 minutes to run on a laptop

Our experiments

We optimised the CHM algorithm and ran $B = 200$ much faster[1]!

Subsequently we ran it fully for $B = 547$ and found 82,026,426 twins – the largest twin found was the following 122-bit twin

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



This data suggests that $B \geq 5000$ to expect to find 256-bit twins

[1] The computation now takes a mere 7 minutes to run on a laptop

Further optimisations for larger B

Further optimisations for larger B

We can run CHM for larger B by restricting the $r, s \in S^{(i)}$ to check

Further optimisations for larger B

We can run CHM for larger B by restricting the $r, s \in S^{(i)}$ to check

`global-`k:	$r < s < k \cdot r$ for fixed $1 < k \leq 2$
`constant-range`:	R successors s of r in $S^{(i)}$ for a range R

Further optimisations for larger B

We can run CHM for larger B by restricting the $r, s \in S^{(i)}$ to check

`global-`k: $r < s < k \cdot r$ for fixed $1 < k \leq 2$

`constant-range`: R successors s of r in $S^{(i)}$ for a range R

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

Table 1: Performance of our CHM optimisations for $B = 300$

Further optimisations for larger B

We can run CHM for larger B by restricting the $r, s \in S^{(i)}$ to check

`global-`k: $r < s < k \cdot r$ for fixed $1 < k \leq 2$

`constant-range`: R successors s of r in $S^{(i)}$ for a range R

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

Table 1: Performance of our CHM optimisations for $B = 300$

For example, we ran $B = 1300$ using `constant-range` with $R = 5000$

Isogeny-based protocols

Twin smooth integers in isogeny-based cryptography

Twin smooth integers in isogeny-based cryptography

Cryptographic sized primes p such that $p+1$ and $p-1$ are as smooth as possible

Twin smooth integers in isogeny-based cryptography

Cryptographic sized primes p such that $p+1$ and $p-1$ are as smooth as possible

~~*B-SIDH*~~

$$\phi : E \to E'$$
$$\#E(\mathbb{F}_{p^2}) = (p-1)^2, (p+1)^2$$

SQISign

Twin smooth integers in isogeny-based cryptography

Cryptographic sized primes p such that $p+1$ and $p-1$ are as smooth as possible

~~*B-SIDH*~~

$$\phi : E \to E'$$
$$\#E(\mathbb{F}_{p^2}) = (p-1)^2, (p+1)^2$$

SQISign

Such primes can be found from twin smooth integers, $(r, r+1)$, if $p = 2r+1$ is prime

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

Twin smooth integers in isogeny-based cryptography

Cryptographic sized primes p such that $p+1$ and $p-1$ are as smooth as possible

~~*B-SIDH*~~

$$\phi : E \to E'$$
$$\#E(\mathbb{F}_{p^2}) = (p-1)^2, (p+1)^2$$

SQISign

Such primes can be found from twin smooth integers, $(r, r+1)$, if $p = 2r+1$ is prime

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

This p makes all of p^2-1 smooth, but in isogeny-based cryptosystems a large smooth divisor of p^2-1 is sufficient (i.e. a large factor $T' \mid p^2-1$ that is smooth)

Twin smooth integers in isogeny-based cryptography

Cryptographic sized primes p such that $p+1$ and $p-1$ are as smooth as possible

~~*B-SIDH*~~ $\phi : E \to E'$ *SQISign*

$$\#E(\mathbb{F}_{p^2}) = (p-1)^2, (p+1)^2$$

Such primes can be found from twin smooth integers, $(r, r+1)$, if $p = 2r+1$ is prime

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

This p makes all of p^2-1 smooth, but in isogeny-based cryptosystems a large smooth divisor of p^2-1 is sufficient (i.e. a large factor $T' \mid p^2-1$ that is smooth)

B-SIDH (pre Kani): $M \mid p-1$ and $N \mid p+1$ with $M \approx N$ large smooth divisors

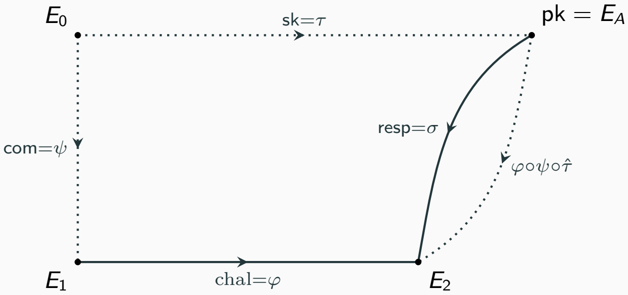
Signing with isogeny skies

Signing with isogeny skies

SQISign: builds a signature from an identification protocol by solving an isogeny problem

Signing with isogeny skies

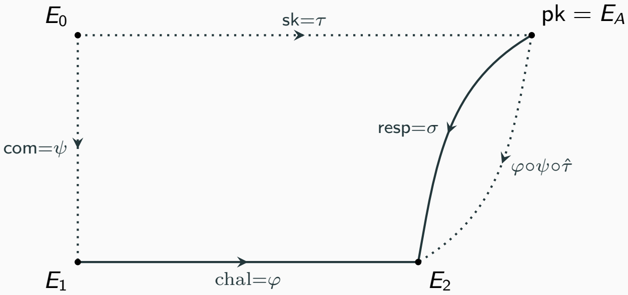
SQISign: builds a signature from an identification protocol by solving an isogeny problem



Dotted isogenies are secret and the other isogenies are public

Signing with isogeny skies

SQISign: builds a signature from an identification protocol by solving an isogeny problem



Dotted isogenies are secret and the other isogenies are public

σ is computed from $\varphi \circ \psi \circ \hat{\tau}$ and the secret knowledge of $\mathrm{End}(E_A)$ and $\mathrm{End}(E_2)$

SQISign requirements

SQISign requirements

State-of-the-art requirements on the prime p

$$2^f T \mid p^2 - 1, \quad f \text{ is as large as possible}, \quad T \approx p^{5/4+\epsilon} \text{ is } B\text{-smooth}, \quad \sqrt{B}/f \text{ is small}$$

SQISign requirements

State-of-the-art requirements on the prime p

$$2^f T \mid p^2 - 1, \quad f \text{ is as large as possible}, \quad T \approx p^{5/4+\epsilon} \text{ is } B\text{-smooth}, \quad \sqrt{B}/f \text{ is small}$$

T is used in the signing to compute ψ and ϕ; $\sqrt{B}/f$ is a rough signing cost metric

SQISign requirements

State-of-the-art requirements on the prime p

$$2^f T \mid p^2 - 1, \quad f \text{ is as large as possible}, \quad T \approx p^{5/4+\epsilon} \text{ is } B\text{-smooth}, \quad \sqrt{B}/f \text{ is small}$$

T is used in the signing to compute ψ and ϕ; $\sqrt{B}/f$ is a rough signing cost metric

2^f is used in the verification to compute σ

SQISign requirements

State-of-the-art requirements on the prime p

$$2^f T \mid p^2 - 1, \quad f \text{ is as large as possible,} \quad T \approx p^{5/4+\epsilon} \text{ is } B\text{-smooth,} \quad \sqrt{B}/f \text{ is small}$$

T is used in the signing to compute ψ and ϕ; $\sqrt{B}/f$ is a rough signing cost metric

2^f is used in the verification to compute σ

Thus verification is fast and signing is slow

SQISign requirements

State-of-the-art requirements on the prime p

$$2^f T \mid p^2 - 1, \quad f \text{ is as large as possible}, \quad T \approx p^{5/4+\epsilon} \text{ is } B\text{-smooth}, \quad \sqrt{B}/f \text{ is small}$$

T is used in the signing to compute ψ and ϕ; $\sqrt{B}/f$ is a rough signing cost metric

2^f is used in the verification to compute σ

Thus verification is fast and signing is slow

How big does p need to be?

NIST security level	p(bits)	Existed?
I	256	✓
III	384	✗
V	512	✗

State-of-the-art prime prior to this work

State-of-the-art prime prior to this work

254-bit prime $p = \texttt{0x348757EADF5C9530B7311A63633F03DB535805FA6E9E48B1FFFFFFFFFFFFFFFF}$:

$$p+1 = 2^{65} \cdot 5^2 \cdot 7 \cdot 11 \cdot 19 \cdot 29^2 \cdot 37^2 \cdot 47 \cdot 197 \cdot 263 \cdot 281 \cdot 461 \cdot 521 \cdot 3923 \cdot R, \text{ and}$$

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

State-of-the-art prime prior to this work

254-bit prime $p = \texttt{0x348757EADF5C9530B7311A63633F03DB535805FA6E9E48B1FFFFFFFFFFFFFFFF}$:

$$p + 1 = 2^{65} \cdot 5^2 \cdot 7 \cdot 11 \cdot 19 \cdot 29^2 \cdot 37^2 \cdot 47 \cdot 197 \cdot 263 \cdot 281 \cdot 461 \cdot 521 \cdot 3923 \cdot R, \text{ and}$$

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

This was found using the *extended Euclidean algorithm* method from Costello (2020):

- Force the large power of two and three in $p \pm 1$ as well as some small primes
- Use XGCD to recover the integer p
- Repeat by changing the distribution of the small prime divisors

State-of-the-art prime prior to this work

254-bit prime $p = \texttt{0x348757EADF5C9530B7311A63633F03DB535805FA6E9E48B1FFFFFFFFFFFFFFFF}$:

$$p + 1 = 2^{65} \cdot 5^2 \cdot 7 \cdot 11 \cdot 19 \cdot 29^2 \cdot 37^2 \cdot 47 \cdot 197 \cdot 263 \cdot 281 \cdot 461 \cdot 521 \cdot 3923 \cdot R, \text{ and}$$

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

This was found using the *extended Euclidean algorithm* method from Costello (2020):

- Force the large power of two and three in $p \pm 1$ as well as some small primes
- Use XGCD to recover the integer p
- Repeat by changing the distribution of the small prime divisors

While $\sqrt{B}/f \approx 0.96$ is not optimally small[2], it performs the best due to the large power of three

[2] Some existing primes have $\sqrt{B}/f$ as small as 0.63

New SQISign parameters

Boosting CHM twins

Boosting CHM twins

Use CHM twins can be combined with $p_n(x) = 2x^n - 1$ to find SQISign parameters

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

Boosting CHM twins

Use CHM twins can be combined with $p_n(x) = 2x^n - 1$ to find SQISign parameters

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

- Find small CHM twins $(r, r \pm 1)$

Boosting CHM twins

Use CHM twins can be combined with $p_n(x) = 2x^n - 1$ to find SQISign parameters

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

- Find small CHM twins $(r, r \pm 1)$
- Choose n and evaluate $p = p_n(r)$

Boosting CHM twins

Use CHM twins can be combined with $p_n(x) = 2x^n - 1$ to find SQISign parameters

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

- Find small CHM twins $(r, r \pm 1)$
- Choose n and evaluate $p = p_n(r)$
- Compute the smooth factor $T' = 2^f \cdot T \mid p^2 - 1$, with T odd

Boosting CHM twins

Use CHM twins can be combined with $p_n(x) = 2x^n - 1$ to find SQISign parameters

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

- Find small CHM twins $(r, r \pm 1)$
- Choose n and evaluate $p = p_n(r)$
- Compute the smooth factor $T' = 2^f \cdot T \mid p^2 - 1$, with T odd
- Keep p if it is prime, $T \approx p^{5/4+\epsilon}$ and $\sqrt{B}/f$ is small

Boosting CHM twins

Use CHM twins can be combined with $p_n(x) = 2x^n - 1$ to find SQISign parameters

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

- Find small CHM twins $(r, r \pm 1)$
- Choose n and evaluate $p = p_n(r)$
- Compute the smooth factor $T' = 2^f \cdot T \mid p^2 - 1$, with T odd
- Keep p if it is prime, $T \approx p^{5/4+\epsilon}$ and $\sqrt{B}/f$ is small

The amount of *guaranteed smoothness* in T' is $\approx p^{1+1/n}$ coming from $(r, r \pm 1)$

Boosting CHM twins

Use CHM twins can be combined with $p_n(x) = 2x^n - 1$ to find SQISign parameters

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

- Find small CHM twins $(r, r \pm 1)$
- Choose n and evaluate $p = p_n(r)$
- Compute the smooth factor $T' = 2^f \cdot T \mid p^2 - 1$, with T odd
- Keep p if it is prime, $T \approx p^{5/4+\epsilon}$ and $\sqrt{B}/f$ is small

The amount of *guaranteed smoothness* in T' is $\approx p^{1+1/n}$ coming from $(r, r \pm 1)$

Depending on n and the power of two f, extra smooth factors might be required[3] to get $T \approx p^{5/4+\epsilon}$

[3] Which comes with an associated smoothness probability

Choosing n

$$(r-1, r) \quad \Longrightarrow \quad p = p_n(r) = 2r^n - 1$$

Choosing n

$$(r-1, r) \quad \Longrightarrow \quad p = p_n(r) = 2r^n - 1$$

$\mathbf{n = 2}$: $p_2(r)^2 - 1 = r^2(r-1)(r+1)$

guaranteed smoothness $T' \approx p^{3/2}$, requires $\log_2(r) \approx 128$ for $\log_2(p) \approx 256$

Choosing n

$$(r-1, r) \quad\Longrightarrow\quad p = p_n(r) = 2r^n - 1$$

$\mathbf{n = 2}$: $p_2(r)^2 - 1 = r^2(r-1)(r+1)$
guaranteed smoothness $T' \approx p^{3/2}$, requires $\log_2(r) \approx 128$ for $\log_2(p) \approx 256$

$\mathbf{n = 3}$: $p_2(r)^2 - 1 = r^3(r-1)(r^2+r+1)$
guaranteed smoothness $T' \approx p^{4/3}$, requires $\log_2(r) \approx 85$ for $\log_2(p) \approx 256$

Choosing n

$$(r-1, r) \quad \Longrightarrow \quad p = p_n(r) = 2r^n - 1$$

$\mathbf{n = 2}$: $p_2(r)^2 - 1 = r^2(r-1)(r+1)$

guaranteed smoothness $T' \approx p^{3/2}$, requires $\log_2(r) \approx 128$ for $\log_2(p) \approx 256$

$\mathbf{n = 3}$: $p_2(r)^2 - 1 = r^3(r-1)(r^2+r+1)$

guaranteed smoothness $T' \approx p^{4/3}$, requires $\log_2(r) \approx 85$ for $\log_2(p) \approx 256$

$\mathbf{n = 4}$: $p_2(r)^2 - 1 = r^4(r-1)(r+1)(r^2+1)$

guaranteed smoothness $T' \approx p^{5/4}$, requires $\log_2(r) \approx 64$ for $\log_2(p) \approx 256$

Choosing n

$$(r-1, r) \quad \Longrightarrow \quad p = p_n(r) = 2r^n - 1$$

$\mathbf{n = 2}$: $p_2(r)^2 - 1 = r^2(r-1)(r+1)$

guaranteed smoothness $T' \approx p^{3/2}$, requires $\log_2(r) \approx 128$ for $\log_2(p) \approx 256$

$\mathbf{n = 3}$: $p_2(r)^2 - 1 = r^3(r-1)(r^2+r+1)$

guaranteed smoothness $T' \approx p^{4/3}$, requires $\log_2(r) \approx 85$ for $\log_2(p) \approx 256$

$\mathbf{n = 4}$: $p_2(r)^2 - 1 = r^4(r-1)(r+1)(r^2+1)$

guaranteed smoothness $T' \approx p^{5/4}$, requires $\log_2(r) \approx 64$ for $\log_2(p) \approx 256$

$\mathbf{n = 6}$: $p_2(r)^2 - 1 = r^6(r-1)(r+1)(r^2-r+1)(r^2+r+1)$

guaranteed smoothness $T' \approx p^{7/6}$, requires $\log_2(r) \approx 43$ for $\log_2(p) \approx 256$

Choosing n

$$(r-1, r) \qquad \Longrightarrow \qquad p = p_n(r) = 2r^n - 1$$

$\mathbf{n = 2}$: $p_2(r)^2 - 1 = r^2(r-1)(r+1)$
guaranteed smoothness $T' \approx p^{3/2}$, requires $\log_2(r) \approx 128$ for $\log_2(p) \approx 256$

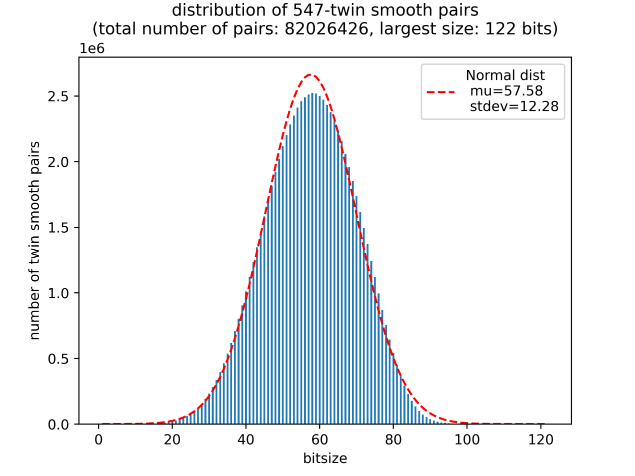
$\mathbf{n = 3}$: $p_2(r)^2 - 1 = r^3(r-1)(r^2+r+1)$
guaranteed smoothness $T' \approx p^{4/3}$, requires $\log_2(r) \approx 85$ for $\log_2(p) \approx 256$

$\mathbf{n = 4}$: $p_2(r)^2 - 1 = r^4(r-1)(r+1)(r^2+1)$
guaranteed smoothness $T' \approx p^{5/4}$, requires $\log_2(r) \approx 64$ for $\log_2(p) \approx 256$

$\mathbf{n = 6}$: $p_2(r)^2 - 1 = r^6(r-1)(r+1)(r^2-r+1)(r^2+r+1)$
guaranteed smoothness $T' \approx p^{7/6}$, requires $\log_2(r) \approx 43$ for $\log_2(p) \approx 256$

For other n, the smoothness probability is too small

Distribution of CHM twins with $B = 547$



NIST-I parameters

NIST-I parameters

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

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

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

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

NIST-I parameters

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

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

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

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

Comparison with the state-of-the-art:

- $\sqrt{B}/f \approx 0.45$
- Expect signing to be $\approx$ 30 - 50% faster
- Expect verification to be $\approx$ 31% slower (which is still very fast)

NIST-I parameters

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

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

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

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

Comparison with the state-of-the-art:

- $\sqrt{B}/f \approx 0.45$
- Expect signing to be $\approx$ 30 - 50% faster
- Expect verification to be $\approx$ 31% slower (which is still very fast)

Remark: True comparison can only be done with an implementation

NIST-III parameters

NIST-III parameters

We report the first NIST-III and NIST-V parameters

NIST-III parameters

We report the first NIST-III and NIST-V parameters

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

NIST-III parameters

We report the first NIST-III and NIST-V 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 $r = 12326212283367463507272925184$:

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

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

NIST-III parameters

We report the first NIST-III and NIST-V 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 $r = 1232621228336746350727292518 4$:

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

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

382-bit prime $p = 2r^6 - 1$ with $r = 118966433886621450 24$:

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

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

NIST-V parameters

We report the first NIST-III and NIST-V parameters

NIST-V parameters

We report the first NIST-III and NIST-V parameters

We used $n = 4, 6$ to find a collection of 512-bit SQISign friendly primes

NIST-V parameters

We report the first NIST-III and NIST-V parameters

We used $n = 4, 6$ to find a collection of 512-bit SQISign friendly primes

499-bit prime $p = 2r^6 - 1$ with $r = 946978778058060446433280 0$:

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

NIST-V parameters

We report the first NIST-III and NIST-V parameters

We used $n = 4, 6$ to find a collection of 512-bit SQISign friendly primes

499-bit prime $p = 2r^6 - 1$ with $r = 946978778058060446433280\,0$:

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

$$p - 1 = 2 \cdot 3^3 \cdot 19 \cdot 23^3 \cdot 31 \cdot 43^2 \cdot 73 \cdot 139 \cdot 337 \cdot 461 \cdot 641 \cdot 971 \cdot 1069 \cdot 1097 \cdot 5843$$
$$\cdot 12841 \cdot 23671 \cdot 39667 \cdot 51193 \cdot 75223 \cdot 459317 \cdot 703981 \cdot R$$

508-bit prime $p = 2r^6 - 1$ with $r = 2669797390044648368060825\,6$:

$$p + 1 = 2^{85} \cdot 17^{12} \cdot 37^6 \cdot 59^6 \cdot 97^6 \cdot 233^6 \cdot 311^{12} \cdot 911^6 \cdot 1297^6, \text{ and}$$

$$p - 1 = 2 \cdot 3^2 \cdot 5 \cdot 7 \cdot 11^2 \cdot 23^2 \cdot 29 \cdot 127 \cdot 163 \cdot 173 \cdot 191 \cdot 193 \cdot 211 \cdot 277 \cdot 347 \cdot 617$$
$$\cdot 661 \cdot 761 \cdot 1039 \cdot 4637 \cdot 5821 \cdot 15649 \cdot 19139 \cdot 143443 \cdot 150151 \cdot R$$

Table of primes

NIST security level	n	r	$\lceil \log_2(p) \rceil$	f	B	$\sqrt{B}/f$	$\log_p(T)$
NIST-I	2	1211460311716772790566574529001291776	241	49	1091	0.67	1.28
		2091023014142971802357816084152713216	243	49	887	0.61	1.28
	3	3474272816789867297357824	246	43	547	0.54	1.29
		10227318375788227199589376	251	31	383	0.63	1.31
		21611736033260878876800000	254	31	421	0.66	1.28
		20461449125500374748856320	254	46	523	0.50	1.26
		26606682403634464748953600	255	40	547	0.58	1.28
	4	1466873880764125184	243	49	701	0.54	1.28
		8077251317941145600	253	49	479	0.45	1.30
		34848218231355211776*	261	77	2311	0.62	1.30
NIST-III	3	1374002035005713149550405343373848576	362	37	1277	0.97	1.25
	4	5139734876262390964070873088	370	45	11789	2.41	1.26
		12326212283367463507272925184	375	77	55967	3.07	1.31
		18080754980295452456023326720	377	61	95569	5.07	1.26
		27464400309146790228660255744	379	41	13127	2.79	1.29
	6	2628583629218279424	369	73	13219	1.58	1.27
		5417690118774595584	375	79	58153	3.05	1.27
		11896643388662145024	382	79	10243	1.28	1.30
NIST-V	4	114216781548581709439512875801279791104*	507	65	75941	4.24	1.26
		123794274387474298912742543819242587136*	508	41	15263	3.01	1.29
	6	9469787780580604464332800	499	109	703981	7.70	1.25
		12233468605740686007808000	502	73	376963	8.41	1.28
		26697973900446483680608256	508	85	150151	4.56	1.26
		31929740427944870006521856	510	91	550657	8.15	1.25
		41340248200900819056793600	512	67	224911	7.08	1.28

Table 2: SQISign parameters $p = p_n(r)$ found using CHM twins. The f is the power of two dividing $(p^2 - 1)/2$ and B is the smoothness bound of $T \approx p^{5/4+\epsilon}$. Those marked with an asterisk correspond to primes p not found using the CHM machinery.

Summary

Summary

Summary

Question: Can we find the largest B-smooth twins?

Summary

Question: Can we find the largest B-smooth twins?

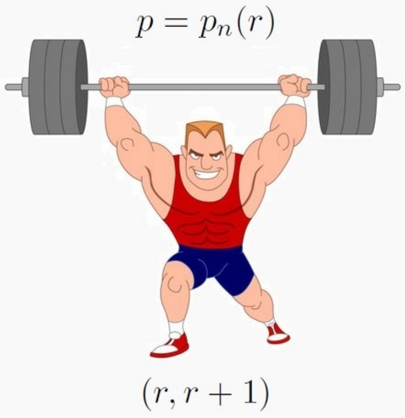
Answer: Yes, but up to 128-bit twins

Summary

Question: Can we find the largest B-smooth twins?

Answer: Yes, but up to 128-bit twins

Can be powerlifted using $p_n(x) = 2x^n - 1$ to find new SQISign parameters



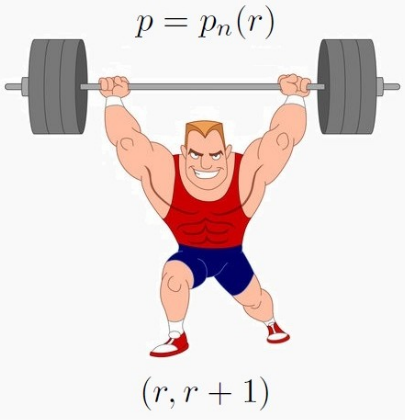
Summary

Question: Can we find the largest B-smooth twins?

Answer: Yes, but up to 128-bit twins

Can be powerlifted using $p_n(x) = 2x^n - 1$ to find new SQISign parameters

Including first parameters targeting higher security levels



Summary

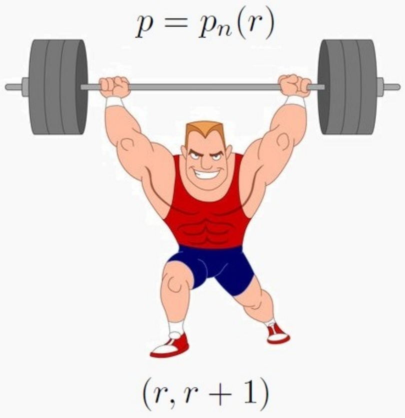
Question: Can we find the largest B-smooth twins?

Answer: Yes, but up to 128-bit twins

Can be powerlifted using $p_n(x) = 2x^n - 1$ to find new SQISign parameters

Including first parameters targeting higher security levels

Open question: Can we find cryptographic sized twins with a small B?



Thanks for listening
Questions?

ia.cr/2022/1439