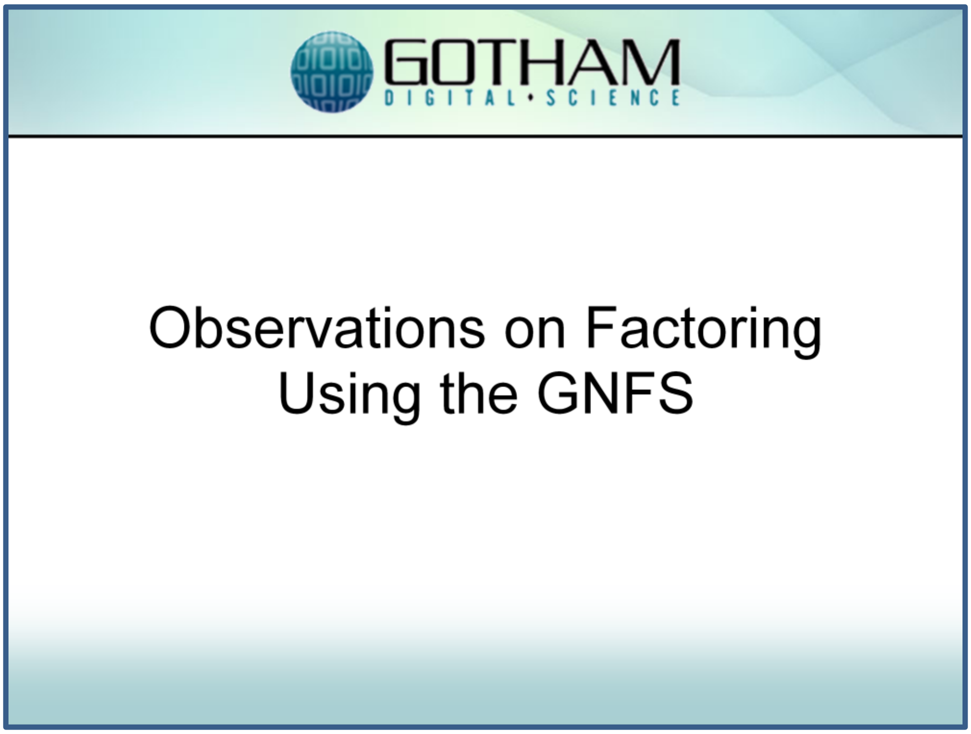
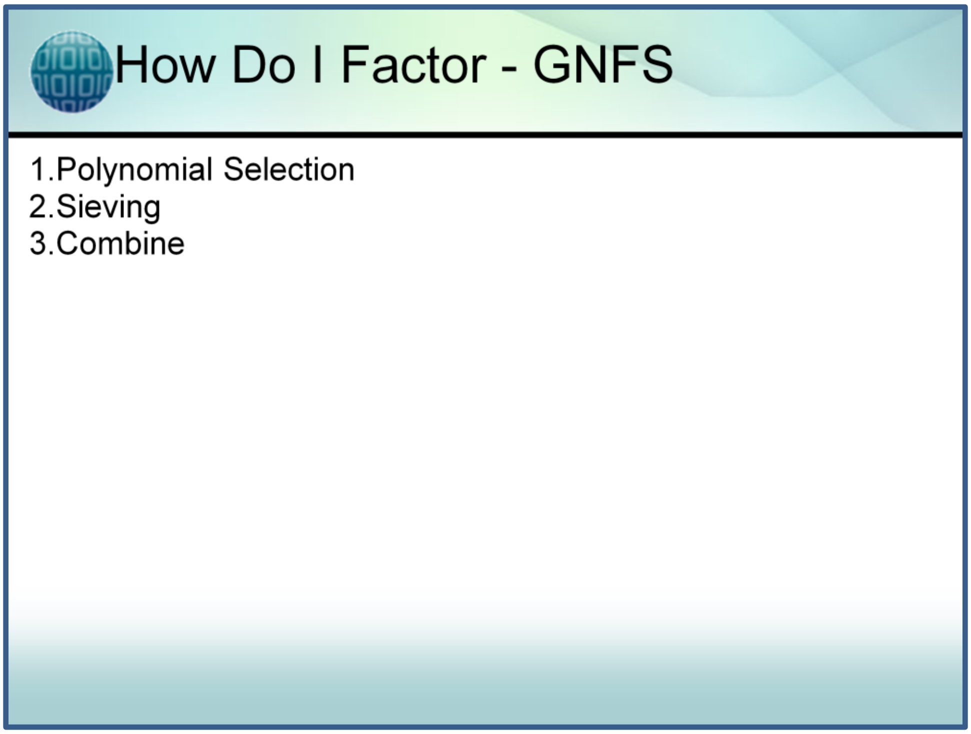
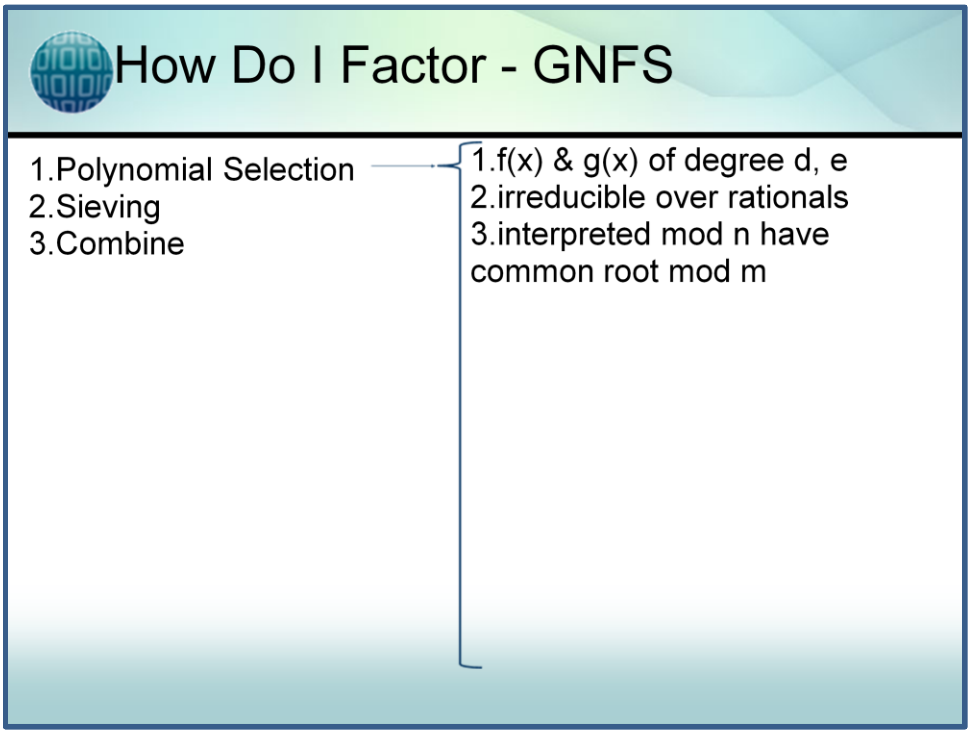


Observations on Factoring Using the GNFS



How Do I Factor - GNFS

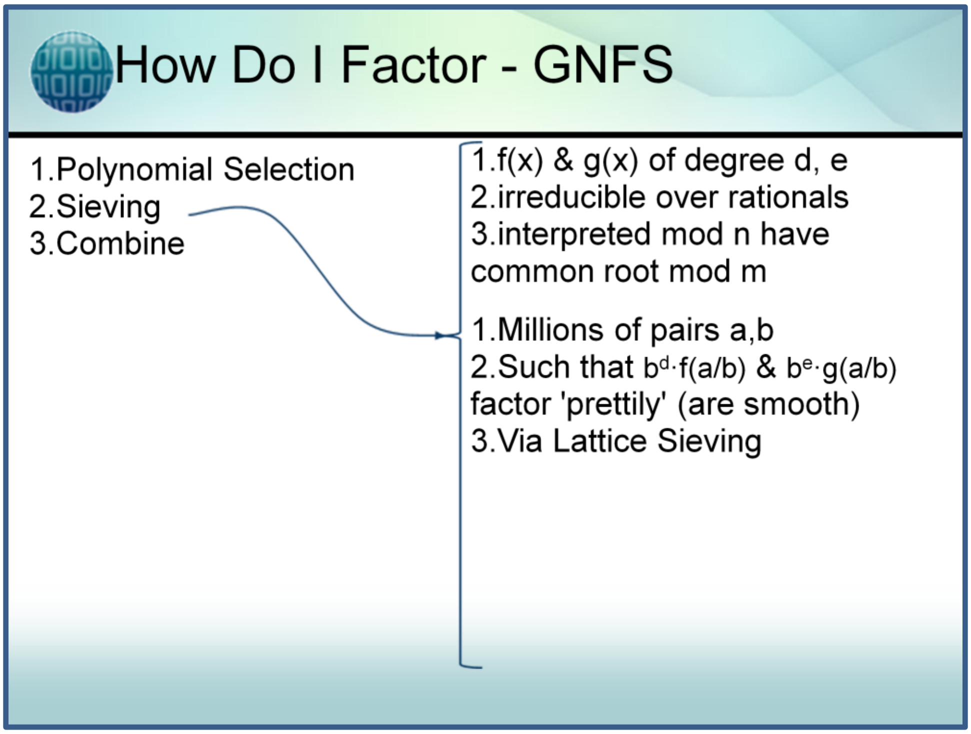
1. Polynomial Selection
2. Sieving
3. Combine



How Do I Factor - GNFS

1. Polynomial Selection
2. Sieving
3. Combine

1. f(x) & g(x) of degree d, e
2. irreducible over rationals
3. interpreted mod n have common root mod m



How Do I Factor - GNFS

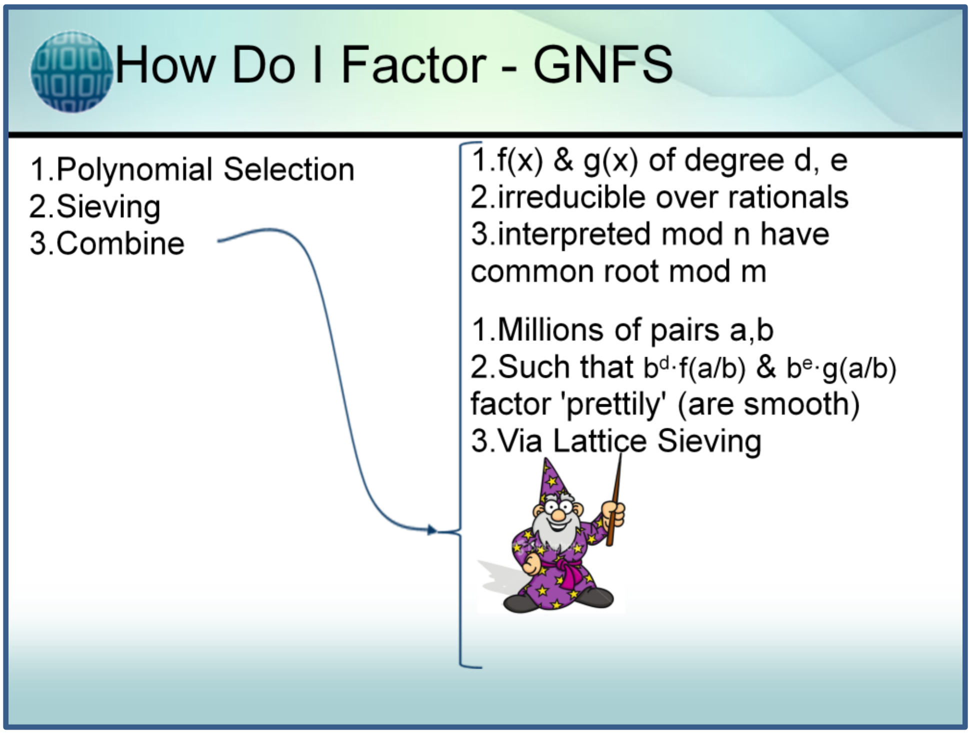
1. Polynomial Selection
2. Sieving
3. Combine

1. f(x) & g(x) of degree d, e
2. irreducible over rationals
3. interpreted mod n have common root mod m

1. Millions of pairs a,b
2. Such that $b^d \cdot f(a/b)$ & $b^e \cdot g(a/b)$ factor 'prettily' (are smooth)
3. Via Lattice Sieving

Some more on this:

http://mersenneforum.org/showthread.php?t=15796

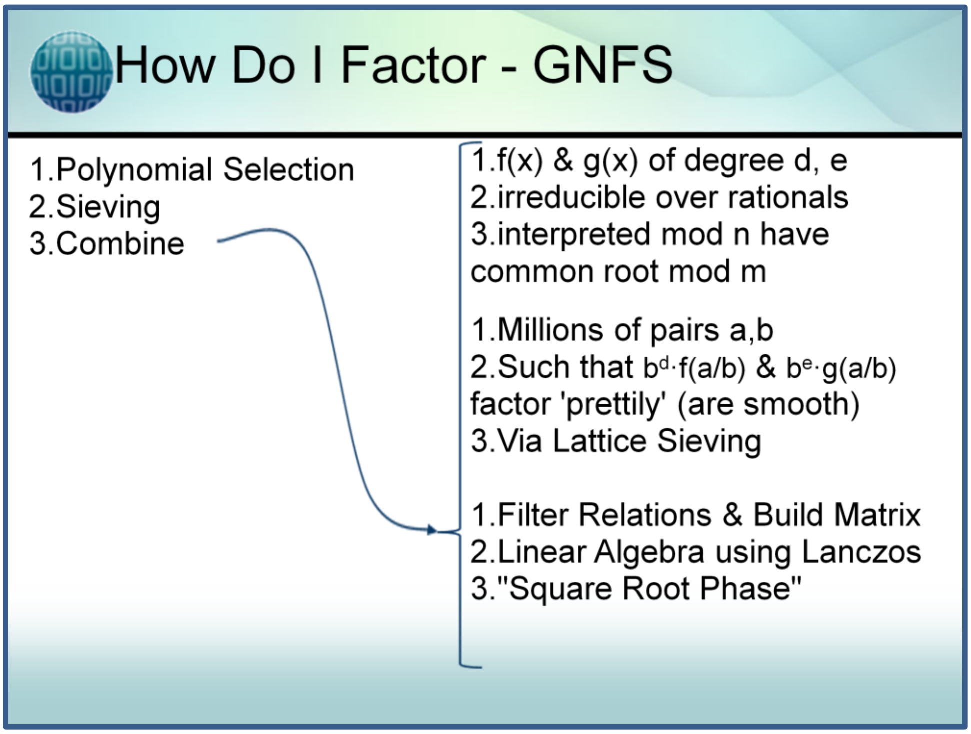


How Do I Factor - GNFS

1. Polynomial Selection
2. Sieving
3. Combine

1. f(x) & g(x) of degree d, e
2. irreducible over rationals
3. interpreted mod n have common root mod m

1. Millions of pairs a,b
2. Such that $b^d \cdot f(a/b)$ & $b^e \cdot g(a/b)$ factor 'prettily' (are smooth)
3. Via Lattice Sieving



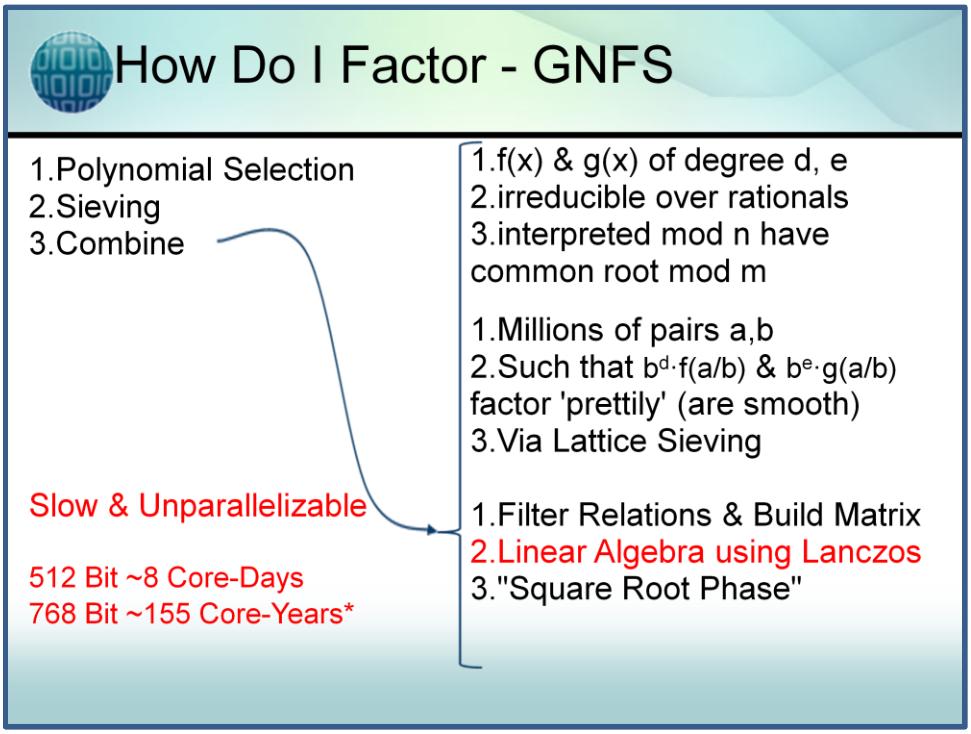
How Do I Factor - GNFS

1. Polynomial Selection
2. Sieving
3. Combine

1. f(x) & g(x) of degree d, e
2. irreducible over rationals
3. interpreted mod n have common root mod m

1. Millions of pairs a,b
2. Such that $b^d \cdot f(a/b)$ & $b^e \cdot g(a/b)$ factor 'prettily' (are smooth)
3. Via Lattice Sieving

1. Filter Relations & Build Matrix
2. Linear Algebra using Lanczos
3. "Square Root Phase"



How Do I Factor - GNFS

1. Polynomial Selection
2. Sieving
3. Combine

Slow & Unparallelizable

512 Bit ~8 Core-Days
768 Bit ~155 Core-Years*

1. f(x) & g(x) of degree d, e
2. irreducible over rationals
3. interpreted mod n have common root mod m

1. Millions of pairs a,b
2. Such that $b^d \cdot f(a/b)$ & $b^e \cdot g(a/b)$ factor 'prettily' (are smooth)
3. Via Lattice Sieving

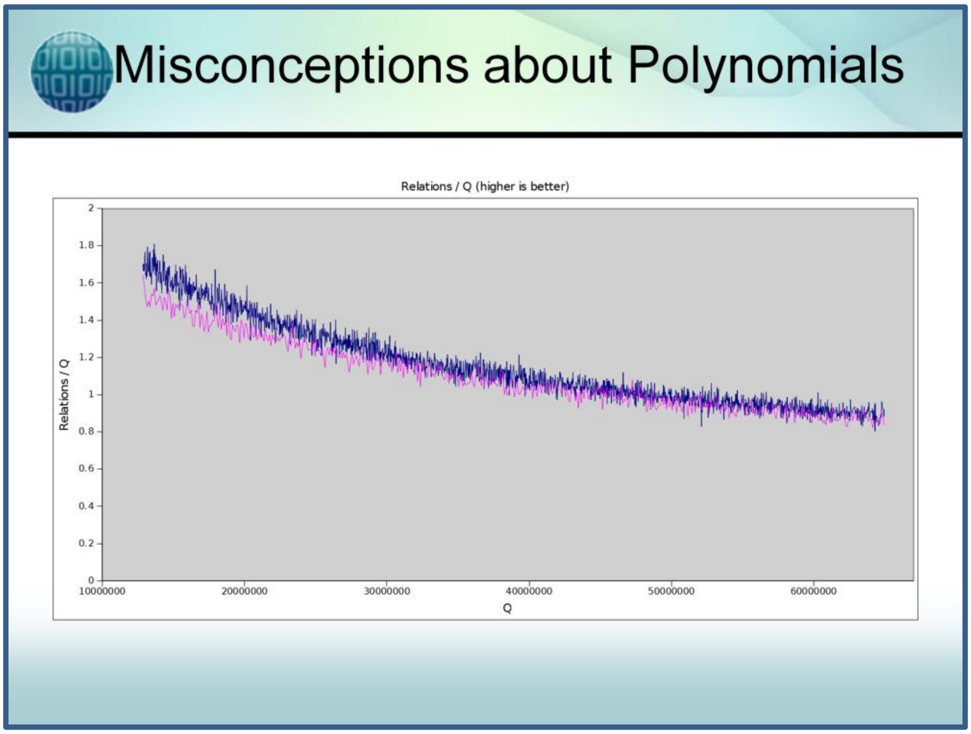
1. Filter Relations & Build Matrix
2. Linear Algebra using Lanczos
3. "Square Root Phase"

Why is it unparrellizable?
http://www.mersenneforum.org/showthread.php?t=15361

* is because the 768 bit semiprime used Block Weildmann as opposed to msieve's block lanczos algorithm.
http://www.mersenneforum.org/showthread.php?t=12958

Some Details on Factoring

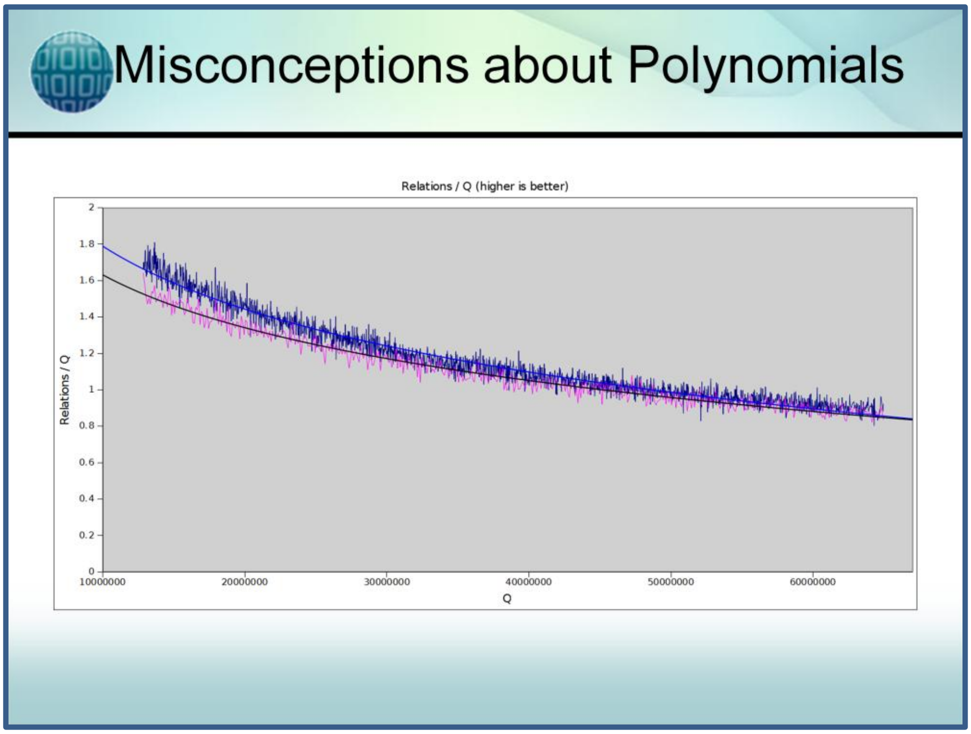
- Polynomial Selection
- Siever Comparisons
- Oversieving



Misconceptions about Polynomials

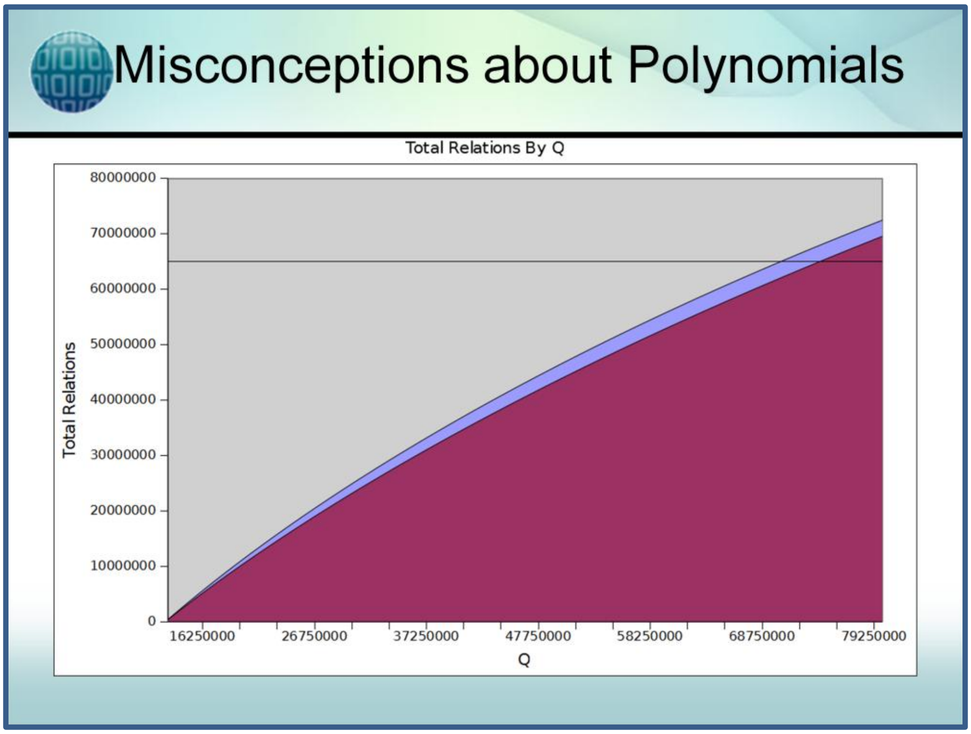
Comparison of sieve results for two polynomials

- Murphy 2.615e-12
- Murphy 3.023e-12



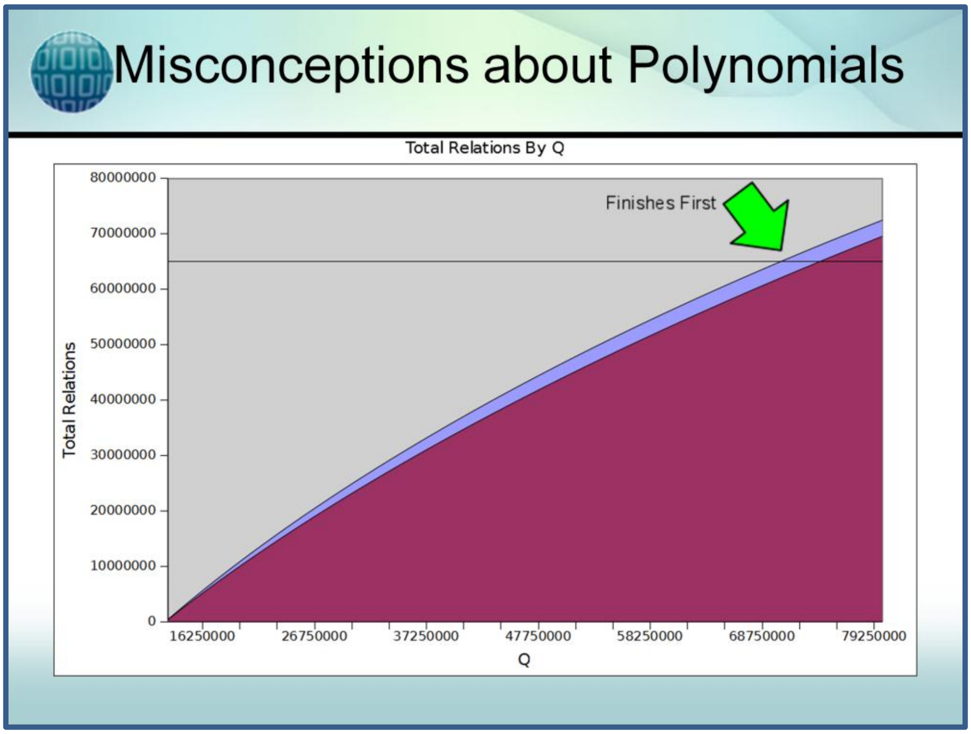
Misconceptions about Polynomials

Trend Lines. We can integrate under these curves to get...

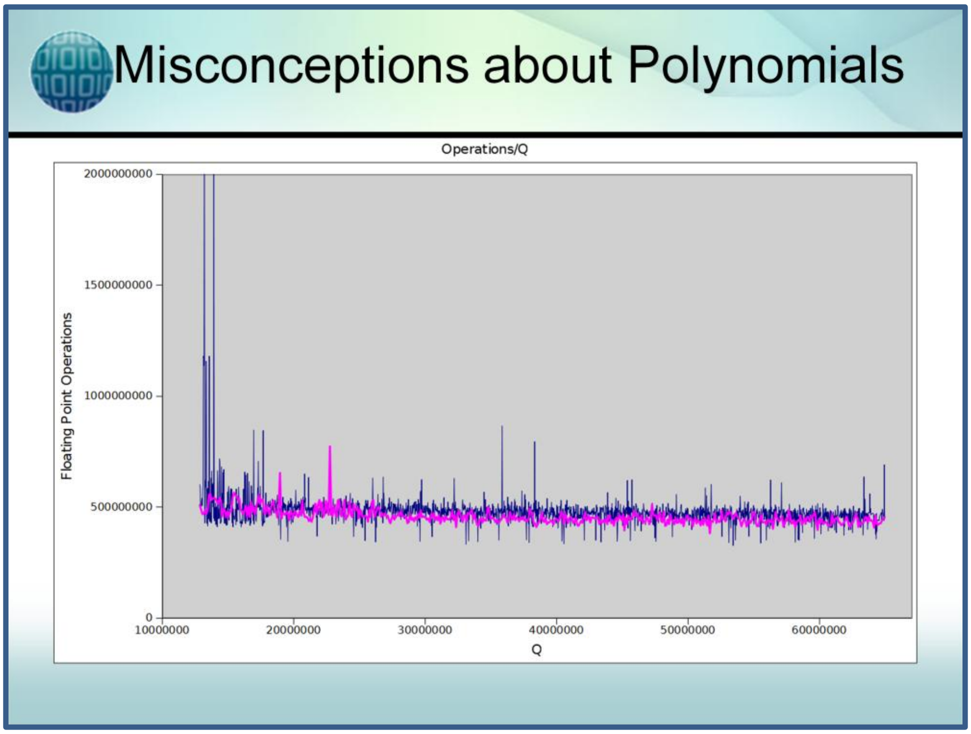


Misconceptions about Polynomials

The total sieve pairs as a function of Q

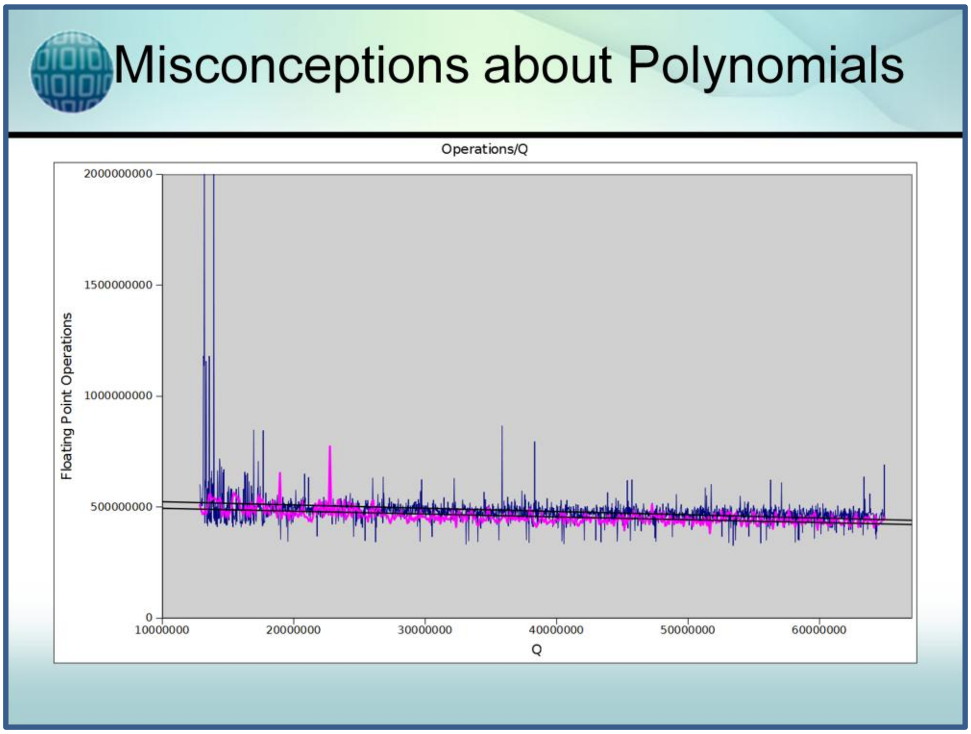


Misconceptions about Polynomials



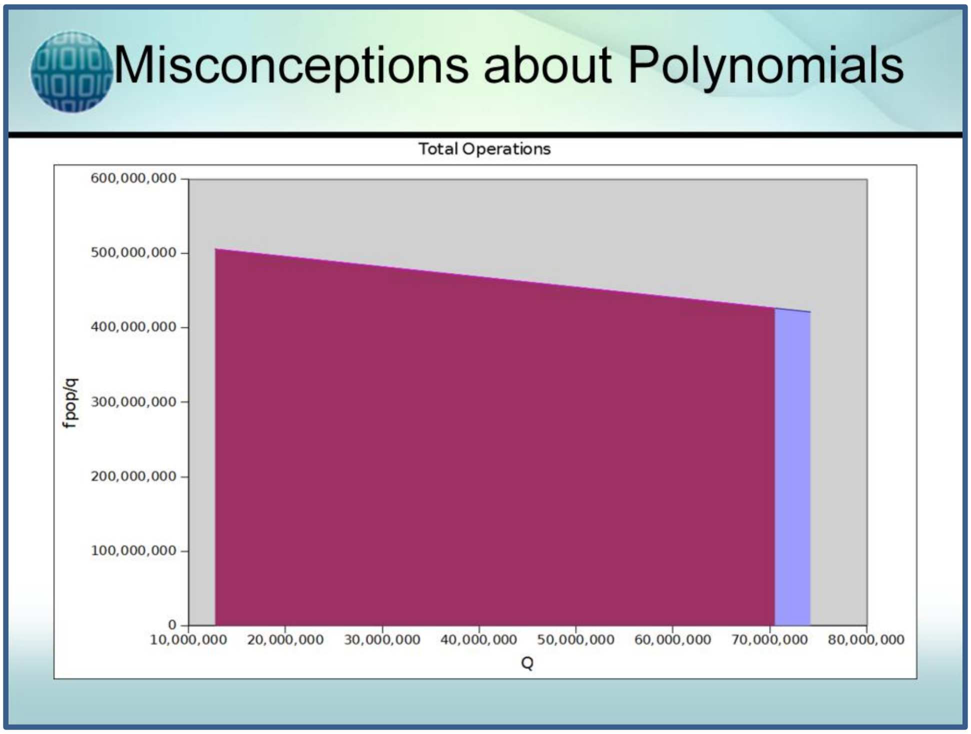
Misconceptions about Polynomials

Floating Point Operations per polynomial



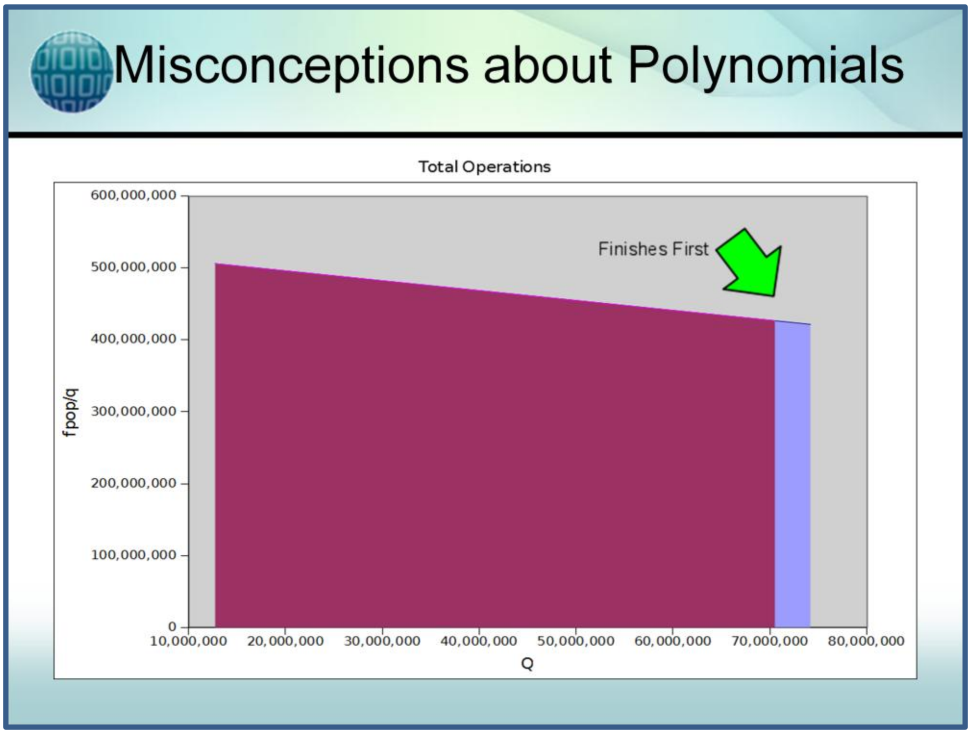
Misconceptions about Polynomials

They're pretty much the same.

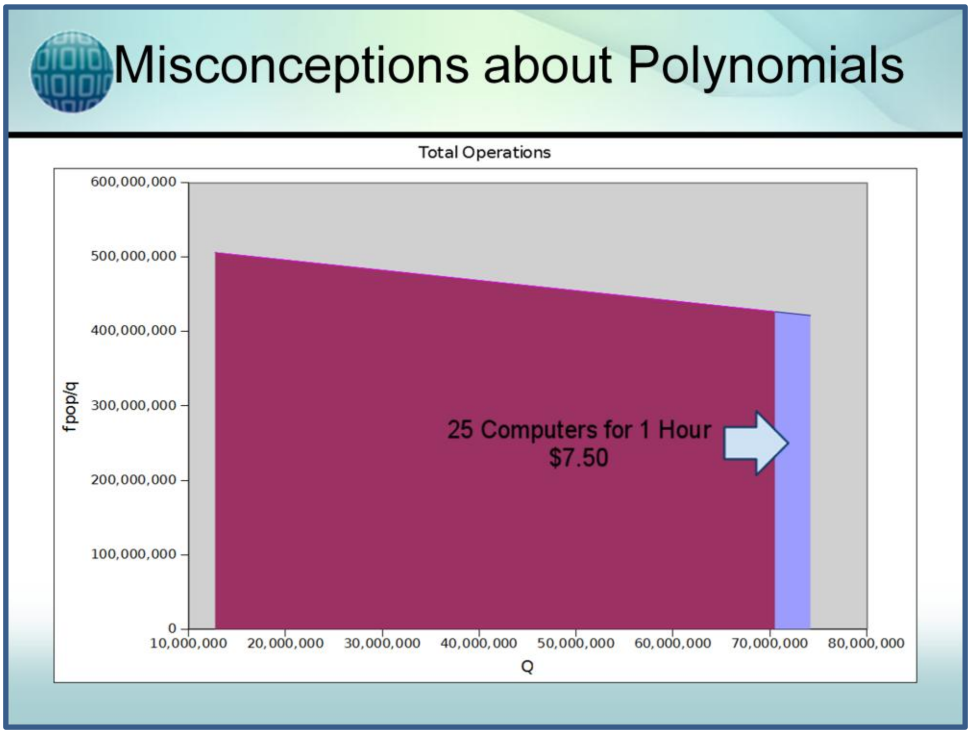


Misconceptions about Polynomials

Integrate under that line (the average actually) and cut it off when the polynomial finishes gathering enough relations, and we have the total amount of work done for each polynomial to achieve the requisite number of sieve pairs.



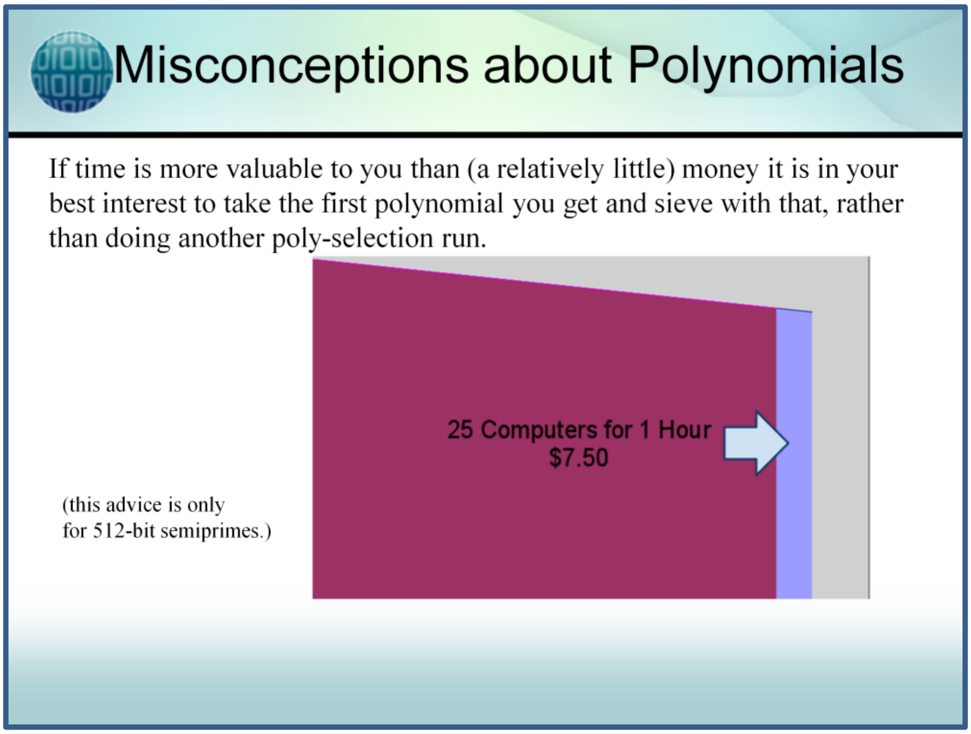
Misconceptions about Polynomials



Misconceptions about Polynomials

25 Computers for 1 Hour
$7.50

Now, because sieving scales horizontally perfectly, and I was working in EC2, that extra bit of work has a real dollar amount on it. And it's not very much.

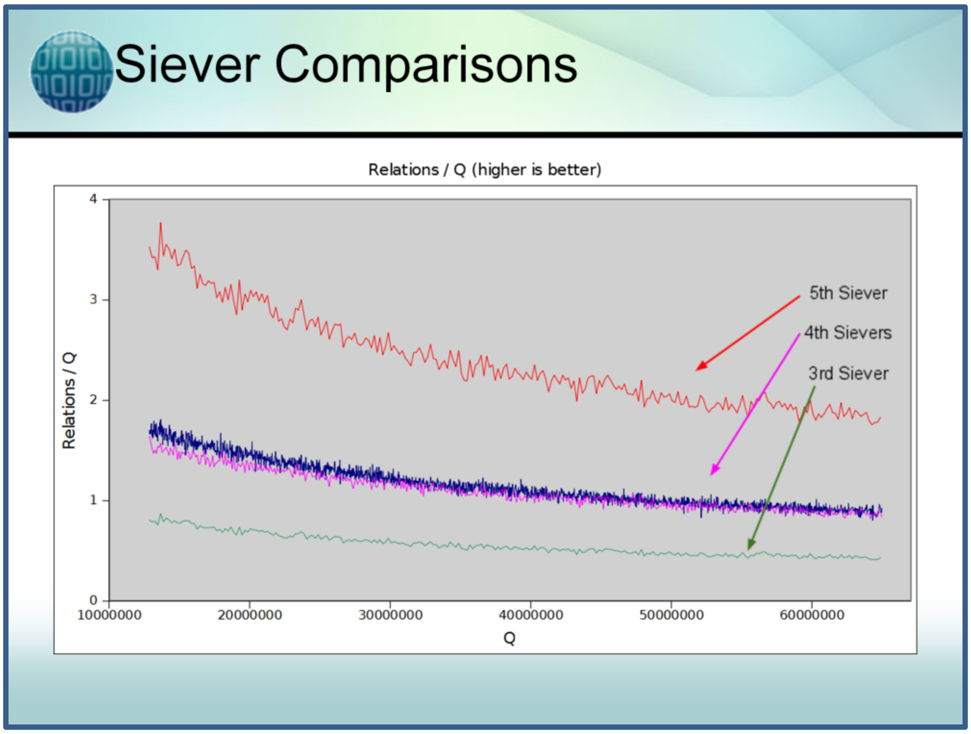


Misconceptions about Polynomials

If time is more valuable to you than (a relatively little) money it is in your best interest to take the first polynomial you get and sieve with that, rather than doing another poly-selection run.

25 Computers for 1 Hour
$7.50

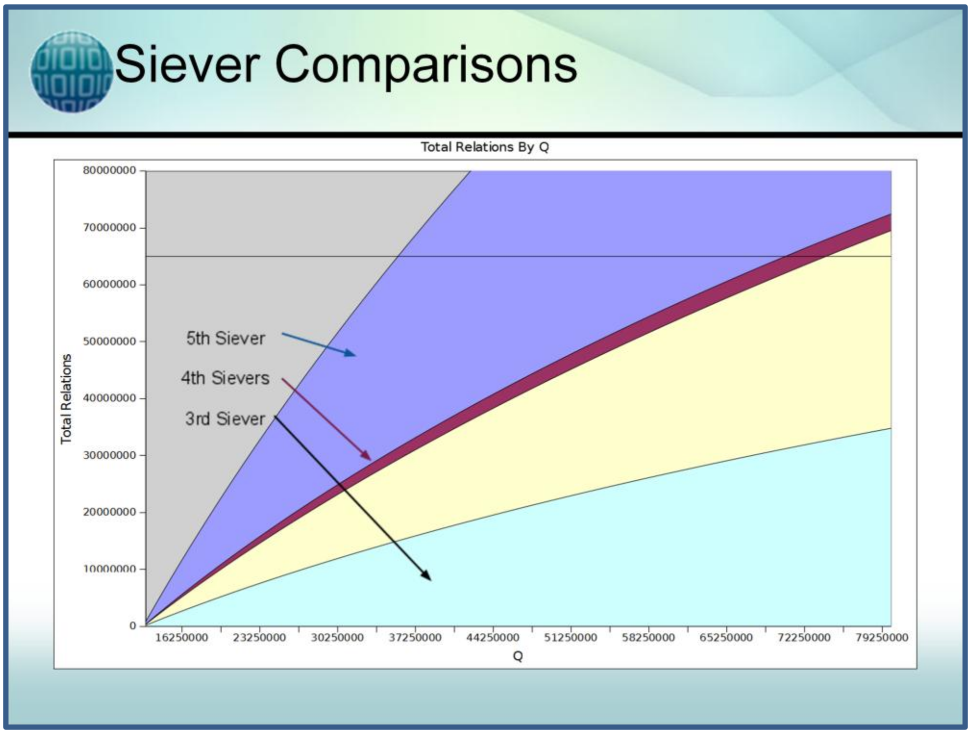
(this advice is only
for 512-bit semiprimes.)



Siever Comparisons

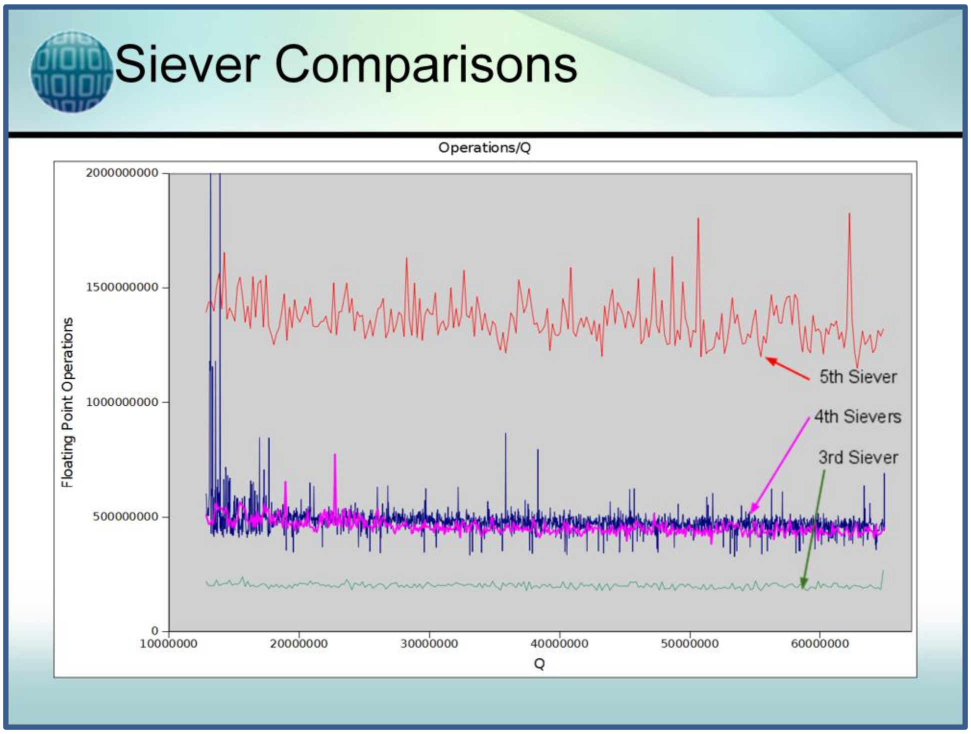
Comparison of

- gnfslasievel13e
- gnfslasievel14e for two polynomials
- gnfslasievel15e



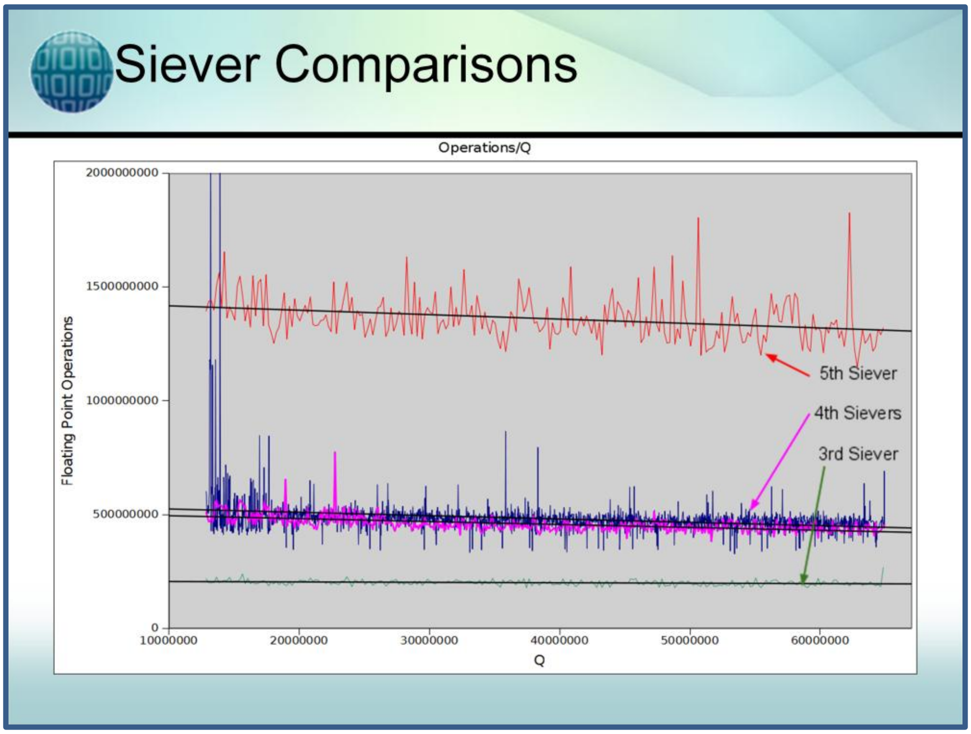
Siever Comparisons

We can again fit trend lines and see where they each finish.



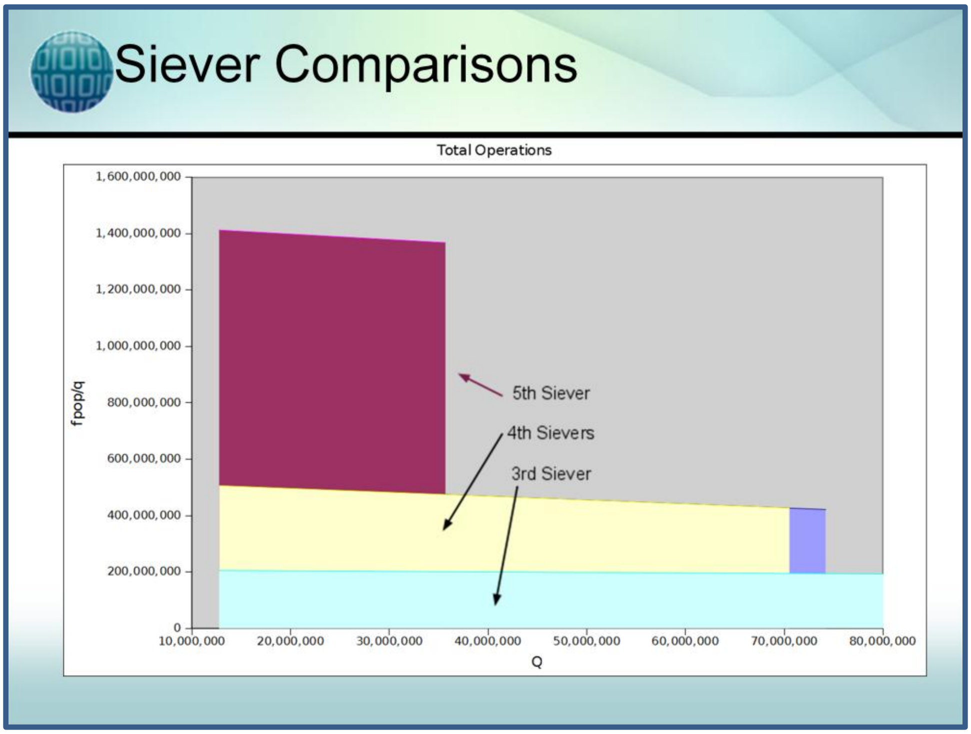
Siever Comparisons

But we need to do an apples-to-apples comparison. While the 5th siever gathers relations much earlier in Q, it also takes much more CPU time to gather those relations.



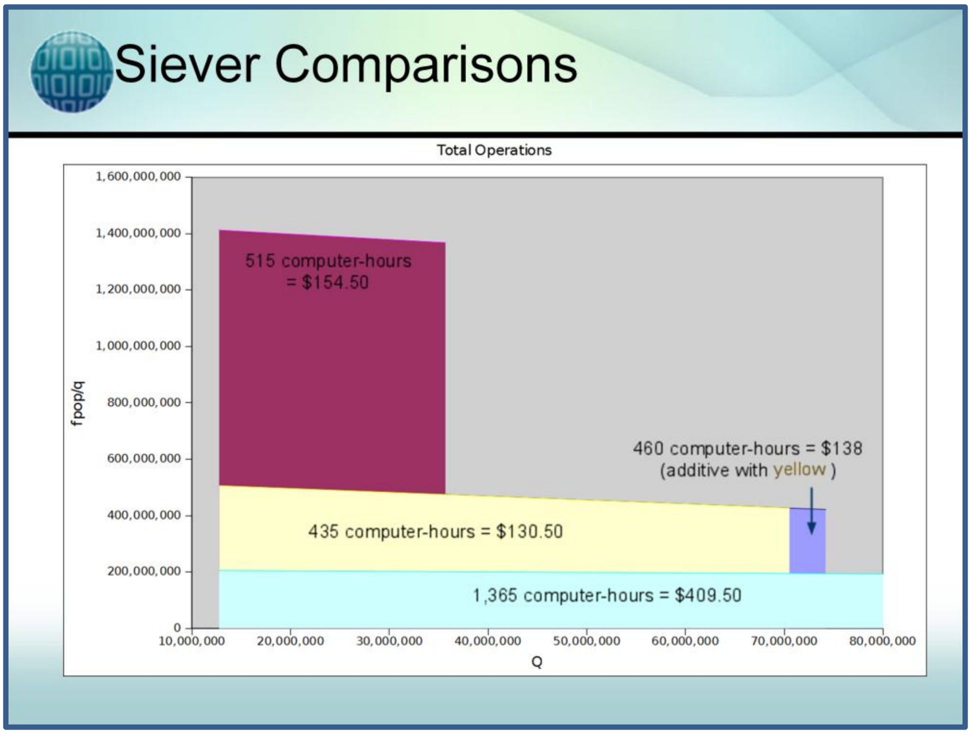
Siever Comparisons

Trend Lines



Siever Comparisons

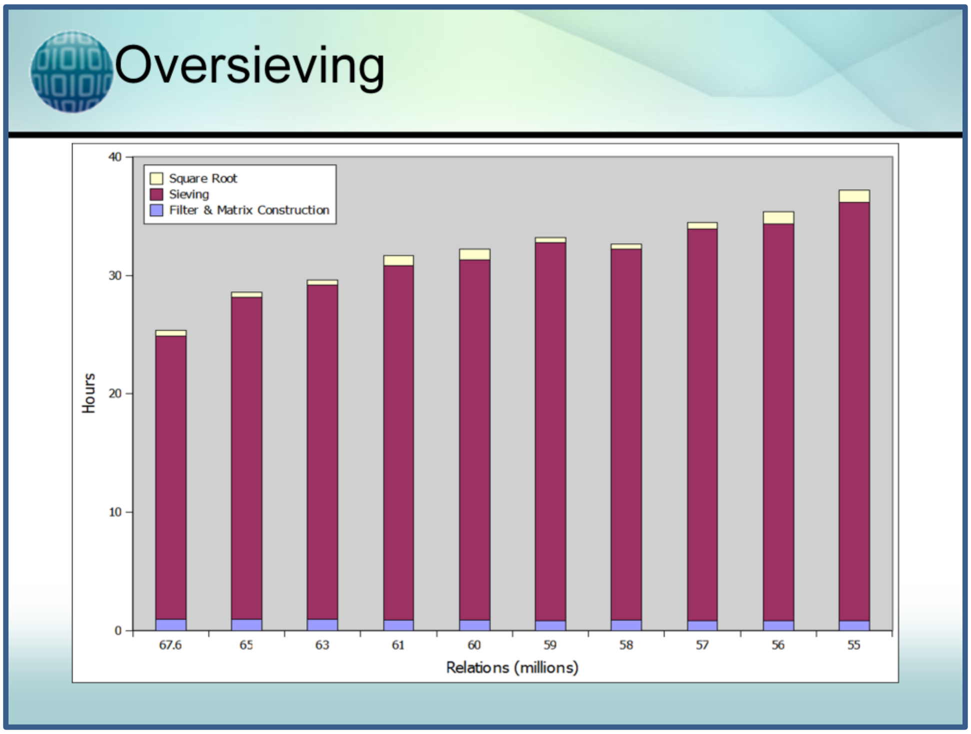
By integrating under the floating point operations trend lines, and stopping when they achieve enough relations we can compare the sievers total work done.



Siever Comparisons

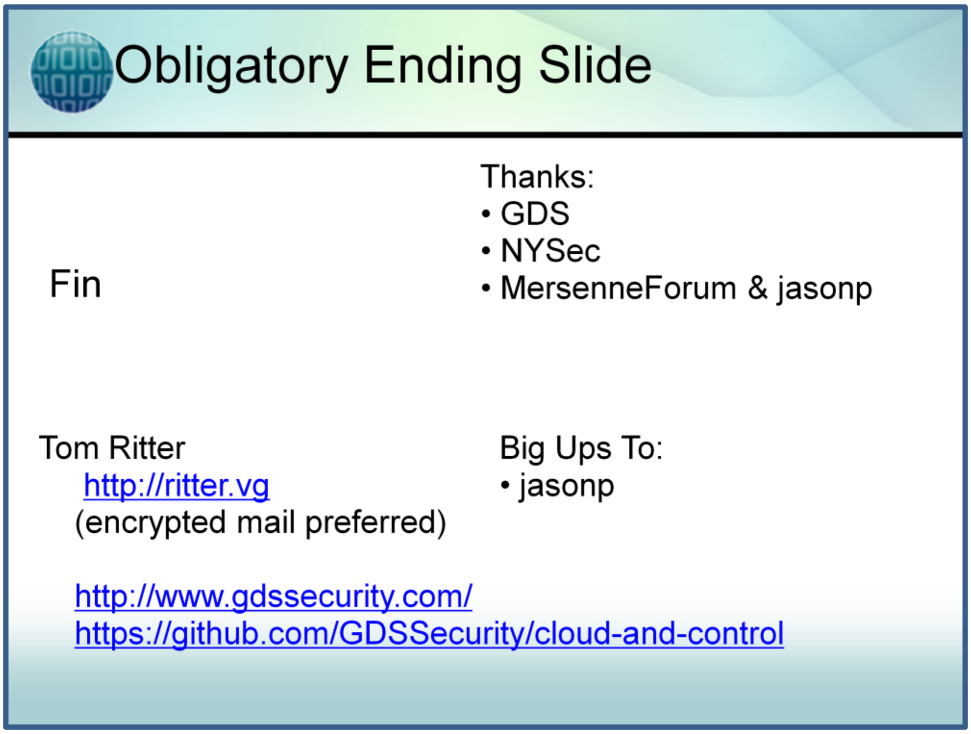
And again, because this is in EC2, we can put this into dollar figures.

This matches up reasonably well with: http://eprint.iacr.org/2011/254



Oversieving

This is a comparison of the Combine done with various numbers of relations. It is clearly in your best interest to oversieve. Sieving scales perfectly out to more machines, and can save you 10 hours in the last step, which is not parallelizable.



Obligatory Ending Slide

Fin

Thanks:
- GDS
- NYSec
- MersenneForum & jasonp

Tom Ritter
http://ritter.vg
(encrypted mail preferred)

Big Ups To:
- jasonp

http://www.gdssecurity.com/
https://github.com/GDSSecurity/cloud-and-control