Sieve-based factoring algorithms

From bicycle chains to number fields

Eran Tromer

tromer@wisdom.weizmann.ac.il

Weizmann Institute of Science

Factoring by square root extraction

Factoring a composite n can be reduced to computing square roots modulo n. Given a black box $\boxed{\sqrt{\ }}$:

Pick $x \in \mathbb{Z}_n$ randomly and compute $x^2 \longrightarrow \boxed{\sqrt{\ }} \longrightarrow y$.

With probability at least $1/2$,

$$x^2 \equiv y^2,\ x \not\equiv \pm y \pmod{n}$$

and then $\gcd(x - y, n)$ is a non-trivial factor of n.

Fermat's method

Special case: $n = x^2 - y^2$ (*difference of squares*)

- Find $x > \sqrt{n}$ such that $x^2 - n$ is a square.
- Equivalently (by $x = a + \left\lceil\sqrt{n}\right\rceil$):

$$f(a) = \left(a - \left\lceil\sqrt{n}\right\rceil\right)^2 - n$$

$$g(a) = \left(a - \left\lceil\sqrt{n}\right\rceil\right)^2 \qquad \longleftarrow \text{ always a square}$$

Try $a = 0, 1, 2, \ldots$ until you find a such that $f(a)$ is also a square over $\mathbb{Z}$.

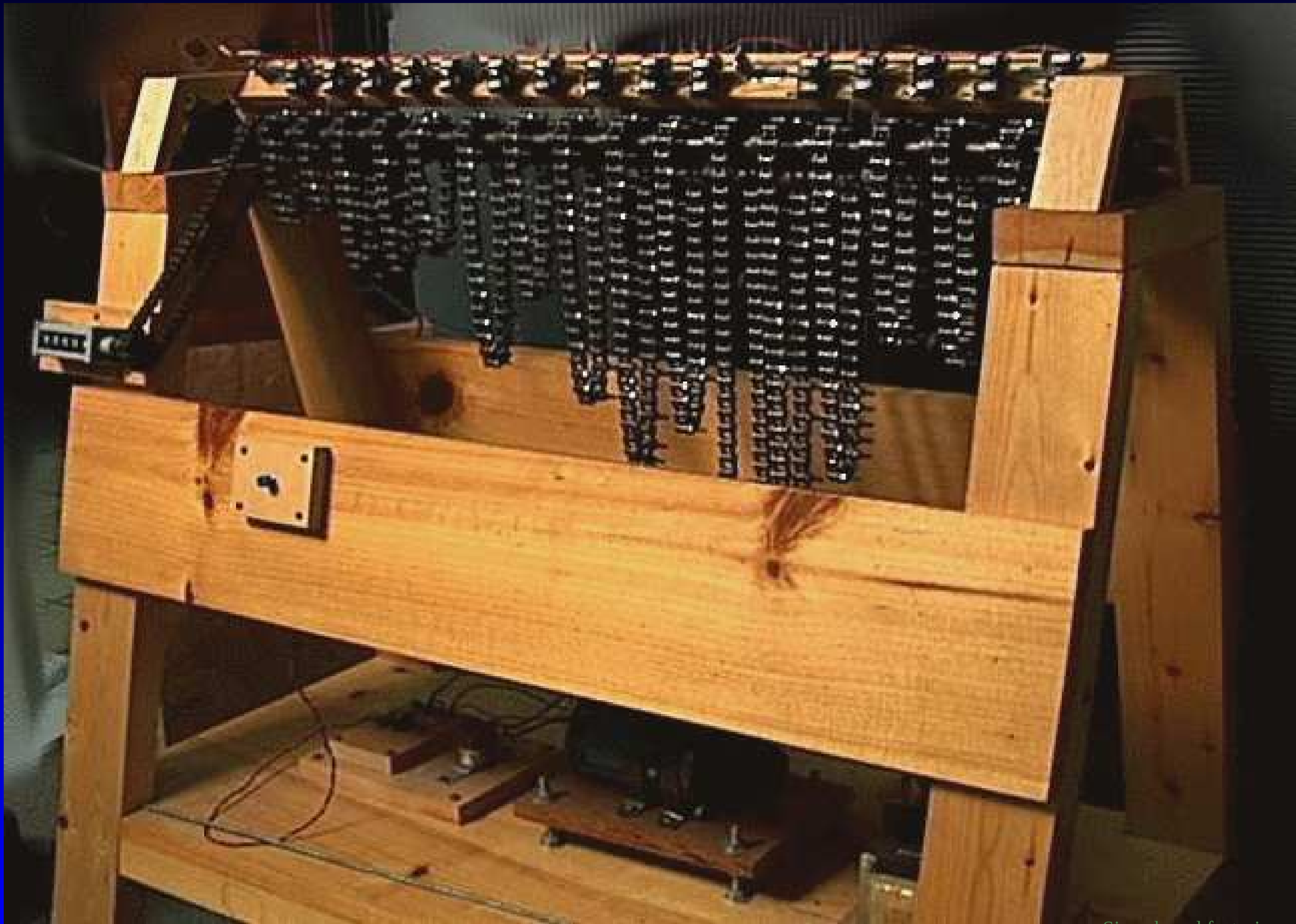
- Compute $x = \sqrt{g(a)},\ y = \sqrt{f(a)}$ over $\mathbb{Z}$.

Idea #1: exploit regularity

$\boxed{f(a) \text{ is a square over } Z} \Longrightarrow$ for any prime p: $\boxed{f(a) \text{ is 0 or a quadratic residue modulo } p\text{: } \left(\frac{f(a)}{p}\right) \in \{0, 1\}}$

- $f(a) \pmod{p}$ has period p.
- About half the values in this period are "good" (quadratic residues or 0).
- We can easily compute these periods.

Lehmer's bicycle chain sieve [1928] (implementation of Fermat's method)

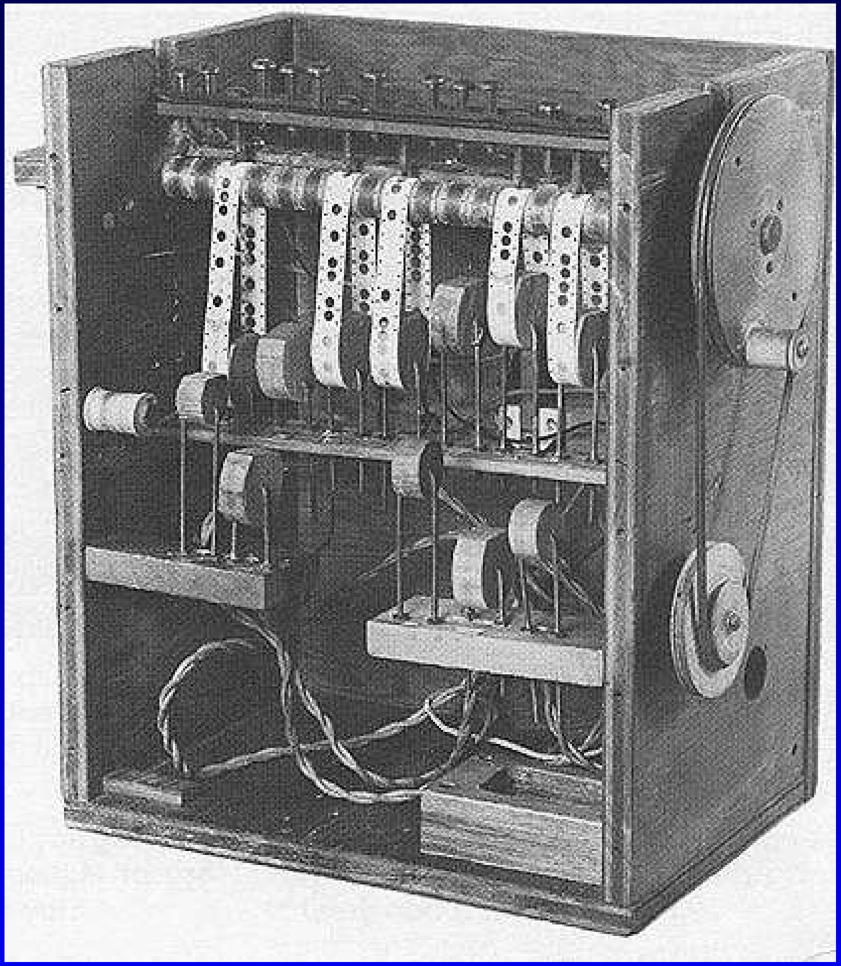


Lehmer's movie film sieve [1932] (implementation of Fermat's method)

Choose many small primes and precompute the corresponding periods of $\left(\frac{f(a)}{p}\right)$.

Scan $a = 1, 2, 3, \ldots$ sequentially while keeping track of the index in each period.

When all periods are at a "good" value, stop and (hopefully) compute $\sqrt{f(a)}$.



Complexity of Fermat's method

The efficient sieving hardware is nice, but doesn't change the asymptotic performance.

There are only $\sqrt{n}$ squares smaller than n, so for a general n we expect to sieve for about $\sqrt{n}$ steps.

This is worse than trial division!

Idea #2: Combine relations

[Morrison,Brillhart 1975]

Let $f'(a) = a^2 \bmod n$, $g'(a) = a^2$.
It's too hard to directly find $a > \sqrt{n}$ such that $f'(a)$ is a square, so instead find a nonempty set $S \subset \mathbb{Z}$ such that $\prod_{a \in S} f'(a)$ is a square. Then compute x, y such that

$$\prod_{a \in S} f'(a) = y^2 \;, \quad \prod_{a \in S} g'(a) = x^2$$

Because $\forall a : f'(a) \equiv g'(a) \pmod{n}$, we get

$$x^2 \equiv y^2 \pmod{n}$$

and $\gcd(x - y, n)$ may be a non-trivial factor of n.

Idea #2: Combine relations (example)

$$
\begin{aligned}
f'(629) &= 102 \\
f'(792) &= 33 \\
f'(120) &= 1495 \\
f'(105) &= 84 \\
f'(52) &= 616 \\
f'(403) &= 145 \\
f'(201) &= 42 \\
&\vdots
\end{aligned}
$$

Idea #2: Combine relations (example)

$f'(629)$	$=$	102	$=$	2	3					17		
$f'(792)$	$=$	33	$=$		3			11				
$f'(120)$	$=$	1495	$=$			5			13		23	
$f'(105)$	$=$	84	$=$	2^2	3		7					
$f'(52)$	$=$	616	$=$	2^3			7	11				
$f'(403)$	$=$	145	$=$			5						29
$f'(201)$	$=$	42	$=$	2	3		7					

$\vdots$

Idea #2: Combine relations (example)

	$f'(629)$	$=$	102	$=$	2	3					17		
$\longrightarrow$	$f'(792)$	$=$	33	$=$		3			11				
	$f'(120)$	$=$	1495	$=$			5			13		23	
	$f'(105)$	$=$	84	$=$	2^2	3		7					
$\longrightarrow$	$f'(52)$	$=$	616	$=$	2^3			7	11				
	$f'(403)$	$=$	145	$=$			5						29
$\longrightarrow$	$f'(201)$	$=$	42	$=$	2	3		7					

$$\vdots$$

$$\prod_{a\in\{792,52,201\}} f'(a) = 2^4 \quad 3^2 \quad 5^0 \quad 7^2 \quad 11^2$$

A square because all exponents are even.

Dixon's algorithm [1981]

How to find S such that $\prod_{a \in S} f'(a)$ is a square?
Consider only the $\pi(B)$ primes smaller than a bound B, and search for $f'(a)$ values that are B-smooth (i.e., factor into primes smaller than B).

- Pick random $a \in \{1, \ldots, n\}$ and check if $f'(a)$ is B-smooth. If so, represent it as a vector of exponents:

$$f'(a) = {p_1}^{e_1} {p_2}^{e_2} {p_3}^{e_3} \cdots {p_k}^{e_k} \mapsto (e_1, e_2, e_3, \ldots, e_k)$$

- Find a subset S of these vectors whose sum has even entries: place the vectors as the rows of a matrix A and find v such that $vA \equiv \vec{0} \pmod{2}$.

$\prod_{a \in S} f'(a) = {p_1}^{e_1} {p_2}^{e_2} {p_3}^{e_3} \cdots {p_k}^{e_k}$ where e_i are all even, so it's a square.

Complexity of Dixon's algorithm

- Let $\rho(\gamma, B)$ be the probability that a random number around γ is B-smooth. Since $f'(a)$ is distributed randomly in $\{0, \ldots, n-1\}$, each trial finds a relation with probability $\sim \rho(n, B)$.
- We need $\pi(B) + 1$ relations. Thus, we need roughly $\pi(B)/\rho(n, B)$ trials.
- In each trial we check divisibility by $\pi(B)$ primes.
- Tradeoff:
  - Decrease $B \longrightarrow$ relations are rarer (smaller ρ).
  - Increase $B \longrightarrow$ more relations are needed and they are harder to identify. Also, computing S is harder since the matrix is larger.

Complexity of Dixon's algorithm (cont.)

- Time complexity for optimal choice of B:
  $e^{(c+o(1))\cdot(\log n)^{1/2}\cdot(\log\log n)^{1/2}}$
- Simple implementation: $c = 2$
  (trial division, Guassian elimination)
- Improved implementation: $c = \sqrt{2}$
  (ECM factorization, Lancos/Wiedemann kernel)
- The approach of combining relations works very well, but we can do better.

The Quadratic Sieve [Pomerance]

Dixon's method looks at the values $f'(a), g'(a)$ for random a.

$$f'(a) = a^2 \bmod n \quad \sim n$$
$$g'(a) = a^2$$

Instead look at $f(a), g(a)$ for $a = 0, 1, 2, \ldots$ as in Fermat's method:

$$f(a) = \left(a - \left\lceil \sqrt{n} \right\rceil\right)^2 - n \quad \sim 2a\sqrt{n}$$
$$g(a) = \left(a - \left\lceil \sqrt{n} \right\rceil\right)^2$$

Smaller number are more likely to be smooth!

$\rho(\sqrt{n}, B) \gg \rho(n, B)$.

The Quadratic Sieve – remember Lehmer

- Task: find many a for which $f(a)$ is B-smooth.
- We look for a such that $p|f(a)$ for many large p:

$$\sum_{p:p|f(a)} \log p > T \approx \log f(a)$$

- Each prime p "hits" at arithmetic progressions:

$$\begin{aligned}\{a : p|f(a)\} &= \{a : f(a) \equiv 0 \pmod{p}\} \\ &= \bigcup_i \{r_i + kp : k \in \mathbb{Z}\}\end{aligned}$$

where r_i are the roots modulo p of the polynomial f (there are either 0 or 2; one on average).

The Quadratic Sieve (cont.)

primes

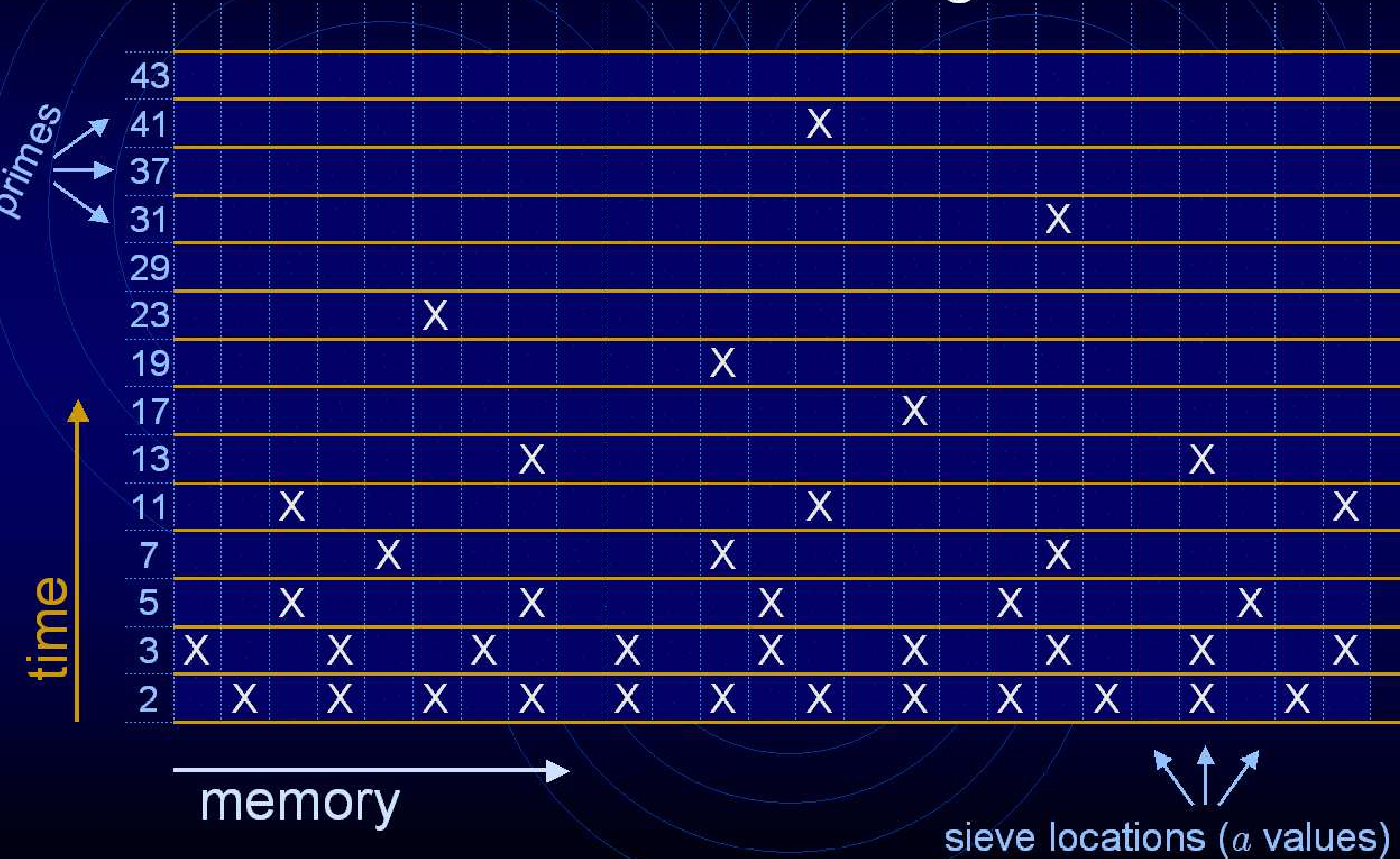
43																									
41														X											
37																									
31																			X						
29																									
23						X																			
19												X													
17																X									
13								X														X			
11			X											X											X
7					X							X							X						
5			X					X					X					X					X		
3	X			X			X			X			X			X			X			X			X
2		X		X		X		X		X		X		X		X		X		X		X		X	

sieve locations (a values)

TWINKLE

[Shamir 1999]

PC-based sieving



Complexity of the Quadatic Sieve

- Conjectured time for optimal choice of B:
$$e^{(c+o(1))\cdot(\log n)^{1/2}\cdot(\log\log n)^{1/2}}$$
with $c = 1$, compared to $c = \sqrt{2}$ for Dixon's algorithm
$\Rightarrow$ can factor integers that are twice longer.
- Variants — self initializing multiple polynomial quadratic sieve.
- Subexponential time, subexponential space
but can practically factor integers up to ~ 400 bits.
- Can we decrease the $(\log n)^{1/2}$ term in the exponent?

Can we?

Key observation: the $e^{\cdots(\log n)^{1/2}\cdots}$ is there essentially because we sieve over numbers of size $\sim n^{1/2}$, which aren't very likely to be smooth.

Find a way to sieve over smaller numbers!

But we don't know how to do that with quadratic polynomials...

The Number Field Sieve

[Pollard,Lenstra,Lenstra,Manasse,Adleman,Montgomery,... 1988–]

As before we have two polynomials f, g that are related modulo n, but now the relation is more subtle: f and g both have a known root m modulo n.

$$f(m) \equiv g(m) \equiv 0 \pmod{n}$$

Also, suppose f and g are monic and irreducible. Let α be a complex root of f. Consider the ring $\mathbb{Z}[\alpha]$ (equivalently, $\mathbb{Z}[x]/(f(x))$). The members of this ring are of the form

$q(\alpha) = q_0 + q_1\alpha + q_2\alpha^2 + \cdots q + q_d\alpha^{\deg f - 1}$.

Operations in this ring are done modulo $f(\alpha)$, simply because $f(\alpha) = 0$.

Similarly define $\mathbb{Z}[\beta]$, where β is a complex root of g.

The Number Field Sieve (cont.)

Consider the ring homomorphism $\phi : \mathbb{Z}[\alpha] \longrightarrow \mathbb{Z}_n$ defined by $\phi : \alpha \mapsto m$ (i.e., replacing all occurances of α with m). Likewise, $\psi : \mathbb{Z}[\beta] \longrightarrow \mathbb{Z}, \psi : \beta \mapsto m$.

Suppose we found a set of integer pairs $S \subset \mathbb{Z} \times \mathbb{Z}$ and also $q(\alpha) \in \mathbb{Z}[\alpha]$ and $t(\beta) \in \mathbb{Z}[\beta]$ such that

$$q(\alpha)^2 = \prod_{(a,b)\in S}(a - b\alpha) \quad \text{over } \mathbb{Z}[\alpha]$$

$$t(\beta)^2 = \prod_{(a,b)\in S}(a - b\beta) \quad \text{over } \mathbb{Z}[\beta]$$

Then $\bmod n$...

$$\left.\begin{array}{l} \phi(q(\alpha))^2 \equiv \phi\left(\prod_{(a,b)\in S}(a - b\alpha)\right) \\ \psi(t(\beta))^2 \equiv \psi\left(\prod_{(a,b)\in S}(a - b\beta)\right) \end{array}\right\} \equiv \prod_{(a,b)\in S}(a - bm)$$

The Number Field Sieve

$$\prod_{(a,b)\in S}(a - b\alpha) = q(\alpha)^2 \quad \text{over } \mathbb{Z}[\alpha]$$

?!

The Number Field Sieve

Let $F(a,b) = b^{\deg f} f(a/b)$. It turns out that

$\boxed{\prod_{(a,b)\in S}(a - b\alpha) \text{ is a square in } \mathbb{Z}[\alpha]}$

$$\implies \boxed{\prod_{(a,b)\in S} F(a,b) \text{ is a square in } \mathbb{Z}}$$

Moreover, the converse “almost” holds.

Therefore, we can work as before: find (a,b) pairs such that $F(a,b)$ is B-smooth, compute their exponent vectors and find dependencies.
To find pairs, we fix values of b and sieve over a.

We do the same for g and $\mathbb{Z}[\beta]$, and find S that satisfies both conditions.

Complexity of the Number Field Sieve

- The point: wisely choose f, g so that their values near 0 are small and therefore likely to be smooth.
- Specifically, we choose f, g of degree $d \approx (\log n / \log\log n)^{1/3}$ and the $F(a, b)$ and $G(a, b)$ values we test for smoothness have size roughly $n^{2/d}$ (vs. $n^{1/2}$ for the QS).
- Conjectured time for optimal parameter choice: $e^{(c+o(1))\cdot(\log n)^{1/3}\cdot(\log\log n)^{2/3}}$ with $c \approx 2$.
- Successfully factored 512-bit and 524-bit composites, at considerable effort.
- Appears scalable to 1024-bit composites using custom-built hardware.

Probability of smoothness

Let $\rho(\gamma, \beta)$ be the probability that a random number around γ is β-smooth. Asymptotically:

$$\left.\begin{array}{l} \gamma = e^{c_1 (\log n)^{d_1} \cdot (\log \log n)^{1-d_1}} \\ \beta = e^{c_2 (\log n)^{d_2} \cdot (\log \log n)^{1-d_2}} \end{array}\right\} \longrightarrow$$

$$\rho(\gamma, \beta) =$$

$$e^{(-c_1(d_1-d_2)/c_2+o(1)) \cdot (\log n)^{d_1-d_2} \cdot (\log \log n)^{1-(d_1-d_2)}}$$