The cost of factoring and “post-quantum RSA”

John M. Schanck
Outline

❖ Brief history of factoring and of estimating the difficulty of factoring.
❖ Quantum computing crash course and Shor’s algorithm.
❖ Quantum computing cost metrics and how trying (and failing) to break “post-quantum RSA” changed how I think about expensive quantum algorithms.
Factoring integers

Fundamental theorem of arithmetic:
Every integer $n > 1$ can be written uniquely as a product of prime powers

$$n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k},$$

where $p_1 < p_2 < \ldots < p_k$ are distinct primes and the $e_i$ are positive integers.

Definitions:
To factor $n$ is to write $n$ as above.
To split $n$ is to find some integer $d$, $1 < d < n$, that divides $n$. 
How large of a number will we ever factor?


❖ 1874, Jevons conjectures nobody (but himself) would ever know the factors of 8,616,460,799.

❖ 1967, Brillhart and Selfridge: “nothing but frustration can be expected from an attack on a number of 25 or more digits”.

❖ 1976, Guy: “I shall be surprised if anyone regularly factors numbers of size 10^80 without special form in the present century.”

❖ 1977, Rivest: factoring 129 digit number would take “40 quadrillion years”.

1931: Lehmer’s photoelectric gear sieve

Fermat’s method

\[ n = uv \quad x = \frac{u + v}{2} \quad y = \frac{u - v}{2} \]

\[ n = (x + y)(x - y) \quad x^2 - n = y^2 \]

- Pick \( t \) (small) integers, \( m_1, m_2, \ldots, m_t \).
- Make \( t \) tables. Table \( i \) contains quadratic residues \( r \) in \( \mathbb{Z}/m_i \) for which \( r + n \) is also a quadratic residue.
- Iterate over candidates \( C \) for \( x^2 - n \).
- Check if \( C \mod m_i \) is in table \( i \) for all \( i=1\ldots t \).
- If \( C \) passes all \( t \) tests, try to factor \( n \).
1931: Lehmer—Powers describe CFRAC
1970: Morrison—Brillhart implement CFRAC

- “In those days, integer factorization was not fashionable” (Odlyzko 1995).
- 45 digit factorizations may have been possible.

\[ F_7 = 2^{27} + 1 = 340,282,366,920,938,463,463,374,607,431,768,211,457 \]
1977: Rivest—Shamir—Adleman Cryptosystem

Public key: \( n = pq \)

Private key: \( d \) such that \( 3d \equiv 1 \pmod{\phi(n)} \)

Encryption

\( \mathcal{E}_n(m) := m^3 \mod n \)

Decryption

\( \mathcal{D}_n(c) := c^d \mod n \)

Notes:
- \( p \) and \( q \) should be primes of roughly equal size that are both congruent to 2 mod 3.
- \( \phi(n) \) is Euler’s totient function.

\[ \phi(n) = |(\mathbb{Z}/n)^*| \]

\[ \phi(pq) = (p - 1)(q - 1) \]
1981: Pomerance describes quadratic sieve
1982: Davis—Holdridge—Simmons implement quadratic sieve

- Factors numbers of roughly twice the bitlength that CFRAC can handle.
- Easier to parallelize than CFRAC.
- 1983: Canfield—Erdős—Pomerance give a lower bound on the number of “smooth” numbers in an interval \([1,x]\). Pomerance conjectures quadratic sieve has complexity
  \[ \exp\left(\sqrt{\log n \log \log n}\right). \]
“The easiest question to ask concerning integer factoring and the hardest to answer, is; ‘How large a number is it computationally feasible to factor using a general purpose factoring routine?’” (Davis—Holdridge—Simmons, 1984)
How large of a number will we ever factor?

More recently:


❖ 2015, NSA: 3072 bit (924 digit) numbers will be hard to factor in short term.

❖ 2017, Bernstein—Heninger—Lou—Valenta: $2^{43}$ bit (i.e. terabyte) numbers will be hard to factor in long term.
1994: Shor describes quantum factoring algorithm
(Yet to be implemented)

- Uses a randomized reduction to order finding.
- Randomly select \( a \in (Z/n)^\times \).
- Compute order of \( a \), i.e. least \( r \) such that \( a^r \equiv 1 \pmod{n} \)
- (For odd composite \( n \) more than half of the elements of \((Z/n)^*\) have even order.)
- Check \( \gcd(a^{r/2} - 1, n) \)
- Splits \( n \) as long as \( r \) is even and \( a^{r/2} \not\equiv -1 \pmod{n} \)
Quantum computing
\[ \Gamma = \{ \text{Memory Configurations} \}. \]

Quantum states are unit vectors in \( \mathbb{C}^{\Gamma} \).

Normalization w.r.t. Hermitian scalar product

\[ \langle \cdot | \cdot \rangle : \mathbb{C}^{\Gamma} \times \mathbb{C}^{\Gamma} \to \mathbb{C}. \]

States are written as \( |\psi\rangle \) where \( \psi \) is a label.

\[ \{ |x\rangle : x \in \Gamma \}, \quad \text{Primes < 100} \]
Postulate: Observable properties of a physical system correspond to self-adjoint operators on its state space.

\[ \sum \lambda_i \mathbf{P}_i \leftarrow \text{Projector} \quad \left( \mathbf{P}_i^2 = \mathbf{P}_i \right) \]

\[ \uparrow \text{real eigenvalue} \]

Observe \( \lambda_i \) with probability \( \langle \psi | \mathbf{P}_i | \psi \rangle \).
Important observables:

- Configuration: $\sum_{x \in \mathbb{R}} \lambda_x |xXx1\rangle$

Measuring $|\psi\rangle = \sum_{x \in \mathbb{R}} \psi_x |x\rangle$ yields $\lambda_x$ with probability $\langle \psi | xXx1 | \psi \rangle = |\psi_x|^2$.

- Total energy / Hamiltonian: $H$
States evolve in time in accordance with the Schrödinger equation:

$$i\hbar \frac{d}{dt} |\psi(t)\rangle = H(t) |\psi(t)\rangle$$

In principle, time evolution can generate any unitary transformation.

$$UU^\dagger = U^\dagger U = I$$
Quantum circuits

Sequential composition

Parallel composition

Universal gate set

\[ |\psi\rangle \xrightarrow{U_0} U_0|\psi\rangle \]

\[ |\psi\rangle \xrightarrow{U_1} U_1|\psi\rangle \]

\[ |\psi_0\rangle \otimes |\psi_1\rangle \xrightarrow{(U_0 \otimes U_1)(|\psi_0\rangle \otimes |\psi_1\rangle)} \]
Here $n$ is the number that we want to split and $s = 2 \lceil \log_2(n) \rceil$. This particular circuit finds the order of 3.
\[ |\psi_2\rangle = \left( \frac{1}{\sqrt{2^S}} \sum_{x=0}^{2^S-1} |x\rangle \right) \otimes |0\ldots0\rangle \]

Uniform superposition over \( S \) bit strings
\[ |\psi_3\rangle = \frac{1}{\sqrt{2^s}} \sum_x |x\rangle \otimes |3^x \mod n\rangle \]
\[ |\Psi_3\rangle = \frac{1}{\sqrt{2^s}} \sum_x |x\rangle \otimes |3^x \mod n\rangle \]

\[ |x\rangle \xrightarrow{\text{QFT}_s} \frac{1}{\sqrt{2^s}} \sum_{y=0}^{2^s-1} \omega^{xy} |y\rangle \]

\[ |\Psi_4\rangle = \frac{1}{2^s} \sum_{x=0}^{2^s-1} \sum_{y=0}^{2^s-1} \omega^{xy} |y\rangle \otimes |3^x \mod n\rangle \]

\[ = \frac{1}{2^s} \sum_{a \in \mathbb{Z}/n^*} \sum_{x \in \chi(a)} \sum_{y} \omega^{xy} |y\rangle |a\rangle \]

Where
\[ \chi(a) = \{ x : 0 \leq x < 2^s \land 3^x \equiv a \mod n \} \]
\[ \chi(a) = \left\{ x : 0 \leq x < 2^5 \land 3^x \equiv a \pmod{n} \right\} \]

\[ = \left\{ x_0 + kr : 0 \leq k \leq \left\lfloor \frac{2^5 - x_0} {r} \right\rfloor \right\} \]

Order of 3

\[ |\Psi_4\rangle = \frac{1}{2^6} \sum_{a} \sum_{x \in \chi(a)} \sum_{y} \omega^{xy} |y\rangle \otimes |a\rangle \]

Measurement yields \( a' \).

\[ |\Psi_5\rangle \propto \sum_{x \in \chi(a')} \sum_{y} \omega^{xy} |y\rangle \]

\[ = \omega^{x_0} \sum_{y} \left( \sum_{k=0}^{r-1} (\omega^y)^k \right) |y\rangle \]
Shor shows that we are likely to obtain a $y$ for which there exists $d$ such that

$$\left| \frac{y}{2^s} - \frac{d}{r} \right| \leq \frac{1}{2^{s+1}}.$$ 

**Theorem**: Let $\alpha \in \mathbb{R}$, and let $\frac{a}{b} \in \mathbb{Q}$ with $a$ and $b$ coprime. If

$$\left| \alpha - \frac{a}{b} \right| < \frac{1}{b^2},$$

then $\frac{a}{b}$ appears as a convergent in the continued fraction expansion of $\alpha$.

Hence the choice of $s$ with $2^s \approx n^2$. 
Gate cost of Shor’s algorithm

- Gate count is dominated by modular exponentiation.
- If $M(t)$ is the cost of $t$-bit multiplication then total gate count is $O(sM(\log n))$.
- Assuming fast multiplication and using $s = 2 \lceil \log n \rceil$ as Shor recommends, this is $O(\log^{2+\epsilon} n)$.
Post-quantum RSA

Bernstein—Henninger—Lou—Valenta recently proposed a variant of RSA with:

- $n = p_1 p_2 \cdots p_k$,
- primes of bit length $(\log \log n)^{2+\epsilon}$
- and operations (key generation, encryption, and decryption) that all cost $(\log n)(\log \log n)^{O(1)}$.

bit operations.
Post-quantum RSA

Example

\[ n = p_1 p_2 \cdots p_{2^{31}} \]

\( p_i \) is a 4096 bit prime for all \( i \)

Efficiency:

“Our heterogeneous cluster was able to generate primes at a rate of 750–1585 primes per core-hour. Generating all 2^{31} primes took approximately 1,975,000 core-hours. In calendar time, prime generation completed in four months running on spare compute capacity of a 1,400-core cluster.”

Security:

“Each multiplication modulo \( n \) inside Shor’s algorithm then uses 2^{56} qubit operations, and overall Shor’s algorithm consumes an astonishing 2^{100} qubit operations.”
Can we reduce ‘s’ in Shor’s algorithm?

- pqRSA assumes $M(2^{43}) = 2^{56}$, and that the modular exponentiation step Shor’s algorithm requires $s = 2^{44}$ multiplications. Only clear path to an improved attack is to reduce $s$.

- One strategy: replace 3 with an element of smaller expected order and take $s$ to be the square of the expected order.

- I don’t see how to do it.
Multi-power post-quantum RSA

An easier problem?

- $n = p_1^{π_1}p_2^{π_2}⋯p_k^{π_k}$,
- primes of bit length $(\log \log n)^{2+ε}$
- and operations (key generation, encryption, and decryption) all still cost
  $(\log n)(\log \log n)^{O(1)}$.
bit operations (but fewer primes are needed).
Multi-power pqRSA

Consider the order of $3^n \mod n$

$$\varphi(n) = \varphi \left( p_1^{\pi_1} p_2^{\pi_2} \cdots p_k^{\pi_k} \right) = \prod p_i^{\pi_i - 1} (p_i - 1)$$

$$3^n = 3 \prod p_i^{\pi_i} \equiv \left( 3 \prod p_i^{\pi_i - 1} \right)^{\prod p_i} \pmod{n}$$

Has order dividing

$$\prod p_i - 1 \approx \exp(k(\log \log n)^{2+\epsilon})$$
Multi-power pqRSA

Example (also with one terabyte \( n \))

\[
 n = p_1^2 p_2^3 p_3^5 p_4^7 \cdots p_{225287}^{20044}
\]

\( p_i \) is a 4096 bit prime for all \( i \)

In Shor’s algorithm we can take \( s = 2 \cdot 4096 \cdot 20044 \approx 2^{27} \)

Again assume \( M(2^{43}) = 2^{56} \), then Shor’s algorithm costs \( 2^{83} \) qubit operations.

But wait! The precomputation, computing \( 3^n \mod n \), costs \( 2^{99} \) bit operations!

What is the most expensive part of this attack?
NIST post-quantum cryptography standardization effort

- Large effort to choose new cryptography (public key encryption, digital signatures, and key encapsulation mechanisms) to replace RSA and other systems vulnerable to quantum attack.
- Over 60 proposals, many sacrificing size/efficiency for security against quantum attacks.
- Better cost analysis of Shor’s algorithm will help us tune these other systems.
Thanks!
Quantum computing

- 1973: Bennett constructs logically reversible Turing machine. Does not propose a physically reversible process.
- 1985: Feynman gives a (non-dissipative) quantum Hamiltonian description of an arbitrary reversible circuit.
- 1984: Zurek shows that Benioff and Feynman’s proposals can be made reliable using dissipative error correction.
- 1989: Deutsch proposes (dissipative) circuits with gates that generate the full unitary group.
- 1993: Yao introduces “gate count” as a measure of quantum circuit complexity