Replication of Quantum Factorisation Records on an 8-bit Home Computer, an Abacus, and a Dog

https://eprint.iacr.org/2025/1237

Peter Gutmann and Stephan Neuhaus

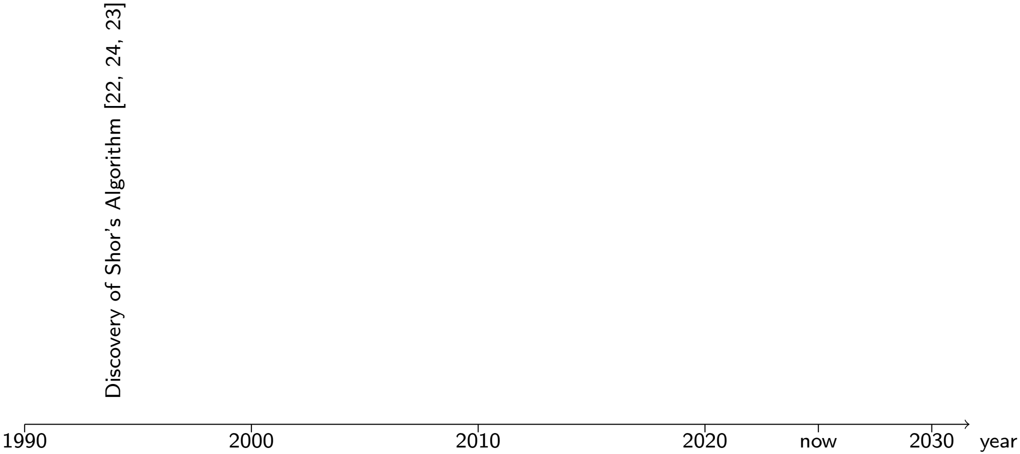
[1]U Auckland

[2]Zurich UAS

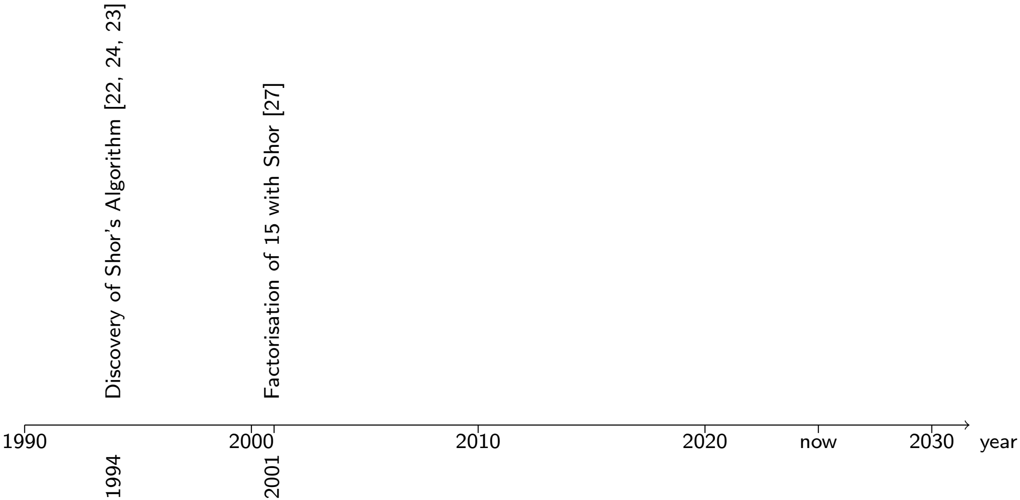
7 October 2025

Introduction

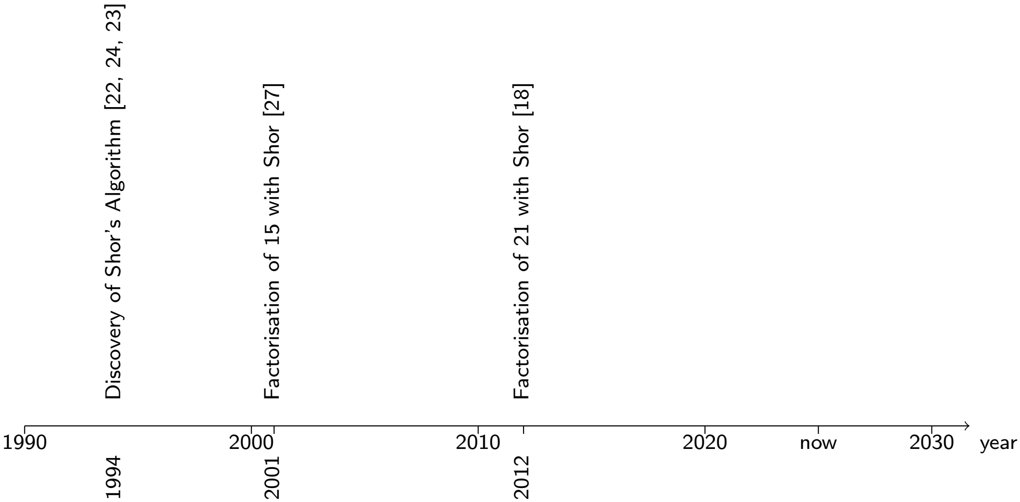
Timeline of quantum factorisations



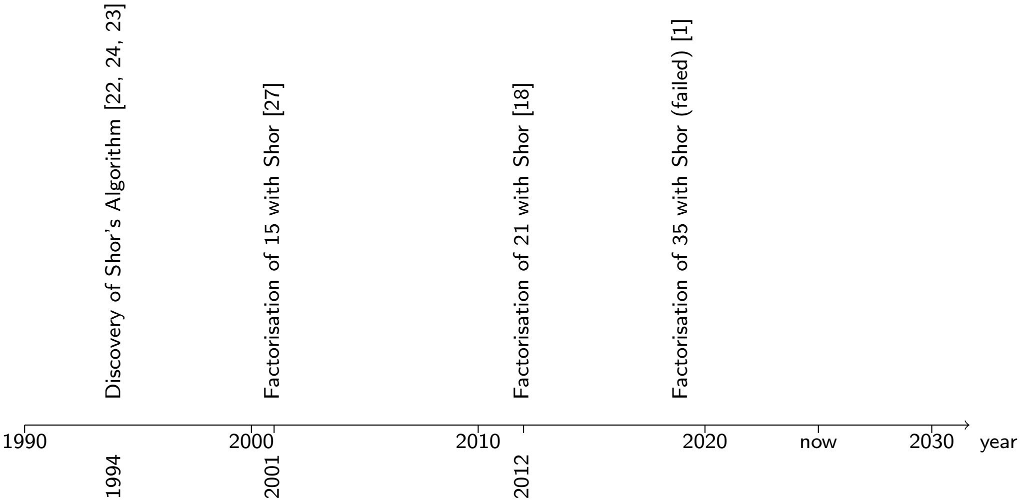
Timeline of quantum factorisations



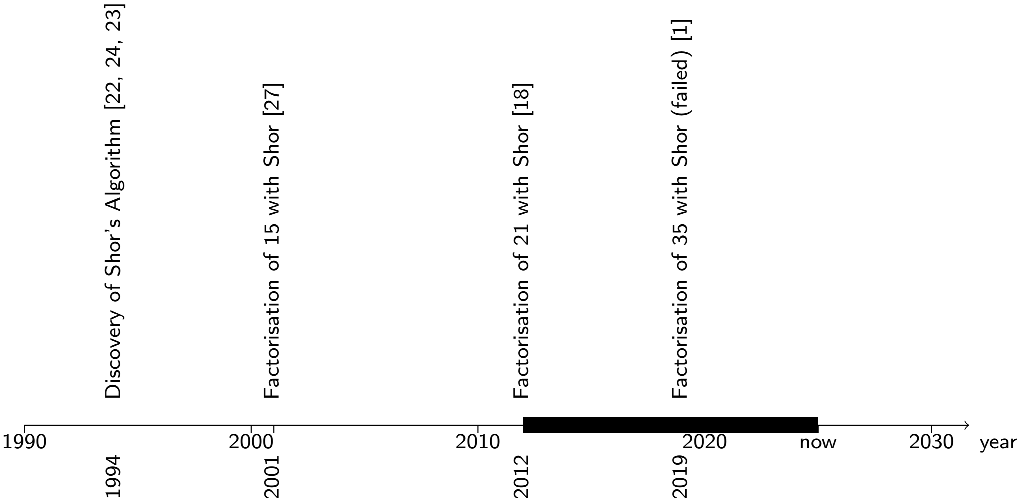
Timeline of quantum factorisations



Timeline of quantum factorisations



Timeline of quantum factorisations



Announcements of quantum factorisations, a selection

- 2012: 143 [31]
- 2019: 1 099 551 473 989 [5]
- 2020: a 6000-digit number [10]
- 2023: 383 123 885 216 472 214 589 586 724 601 136 274 484 797 633 168 671 371 [12]
- 2025: “RSA-2048”, “with D-Wave” (“D-Wave paper”) [28]
- 2025: 4096-bit numbers [3]
- How then can we claim “no progress since 2012”??

Terminology

- New technologies are typically given names that overstate their capabilities

device	initial term
digital computer	electronic brain
LLM	artificial intelligence
large physics experiment	quantum computer

term commonly used	our term	reason
quantum computer	physics experiment	not a computer
abacus	abacus	not a computer
dog	dog	not a computer

- Bonus terminology note: We're using the UK term "factorise" instead of the US terms "factor" and "factorize" in order to avoid the 40 %[1] tariff on the US term.

[1] Or whatever the percentage is today.

Quantum Factorisation Overview

Non-quantum factorisation on a digital computer

- Currently, the best method for factorising an integer with large prime factors is the General Number Field Sieve (GNFS)
- Factorising an n-bit integer takes $O(\exp(((64/9)^{1/3} + o(1))n^{1/3}(\log n)^{2/3}))$ time
- This is known as *subexponential complexity*
- It is growing slower than any exponential, but faster than any polynomial
- Shor's algorithm can factor in polynomial time ($O(n^2)$ for an n-bit number), which is why it is such a big deal
- Works only on a suitably large physics experiment
- But we can't seem make large numbers of fully entangled low-noise qubits
- That was why the factorisation of 35 failed: the computation became too noisy

Factorising with Shor's Algorithm

1. Let N be the number to be factorised (note we don't know p or q)
2. Choose random a with $0 < a < N$ (write this down, it'll be important later)
3. Perform quantum magic to compute r, the period of $a^x \bmod N$
4. If r is odd, or if $a^{r/2} \equiv -1 \pmod{N}$, go back to step 3 to try again with different a
5. Otherwise, $p, q = \gcd(a^{r/2} \pm 1, N)$

- Needs $O(n)$ qubits and $O(n^2)$ time, where n is the number of bits in N
- "In 2023, Jin-Yi Cai showed that in the presence of noise, Shor's algorithm fails asymptotically almost surely for large semiprimes that are products of two primes. [Prime factors of large semiprimes] have a positive density in the set of all primes." (Wikipedia)
- Real-life implementations of Shor's algorithm will therefore need more qubits (for error correction due to noise)
- We don't know how to make many fully entangled low-noise qubits

Sleight-of-hand tricks

- Has anyone ever factorised a number with Shor's algorithm?

Sleight-of-hand tricks

- Has anyone ever factorised a number with Shor's algorithm?
- No.

Sleight-of-hand tricks

- Has anyone ever factorised a number with Shor's algorithm?
- No.
- Instead, various tricks are employed

Trick: use a modified version form of Shor's Algorithm

- Recall from above that Shor's algorithm tries various values of a
- Knowing the right value of a up front speeds up the algorithm: runs in $O(1)$ time!
- (For large values of 1)
- Further, *if you already know p and q* (i.e., if you cheat), there are always values of a so that the experiment works with fewer qubits. . . (a *modified version* of Shor's Algorihm)
- . . . and an a can be precomputed that needs only two qubits, with the "Smolin–Smith–Vargo Algorithm" [25] (the *compiled form* of Shor's Algorithm)
- Obviously, the main point is not to run Shor's Algorithm at all, but the preprocessing that enables a highly modified version of Shor's Algorithm to run on a physics experiment with a greatly constrained (i.e., small) number of qubits

How to spot modified or compiled forms

- *In the absence of noise*, $2 + 1.5 \log N$ is a practical lower bound on the number of qubits [32] (though $1.1 \log N$ not impossible)
- If the paper needs less than this to factor N, it's probably using the compiled or otherwise modified form
- In practice, noise is always present, especially for "larger" numbers, like, say, 35

N	citation	lower bound	used	compiled?	modified?
15	[27]	8	7		yes
21	[18]	9[2]	2.6[3]		yes
35	[1]	10	7		yes
RSA-768	[25]	1154	2[4]	yes	
N-20000	[25]	30 002	2[3]	yes	

[2] Reference [25] gives 10, but I think that's wrong

[3] 1 qubit plus 1 qutrit (three-state system), giving $1 + \log 3 = 2.585\ldots$ qubits

[4] Designed, but not actually performed

Trick: use trivially factorised values

$$N = pq = (2^n - 1)(2^m + 1), \quad n \leq m$$

"Callas Normal Form", first described by cryptographer Jon Callas [2] (named by us)

n						$m-n$				n					
1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1

- Was the form used (not by Jon!) to claim factorisation of 4096-bit numbers [3]
- Obviously easily detected and factorised on a digital computer (once you recover m and n, you recover p and q)
- Obviously never generated by a proper RSA key-generation routine

Trick: use trivially factorised values

$$N = pq = (2^n - 1)(2^m + 1), \quad n \leq m$$

"Callas Normal Form", first described by cryptographer Jon Callas [2] (named by us)

n						$m-n$				n					
1	1	1	1	1	1	0	0	0	0	1	1	1	1	1	1

- Was the form used (not by Jon!) to claim factorisation of 4096-bit numbers [3]
- Obviously easily detected and factorised on a digital computer (once you recover m and n, you recover p and q)
- Obviously never generated by a proper RSA key-generation routine
- This paper also had $p = 3$ throughout, which makes factorisation ahem somewhat easier

Trick: make factorisation depend only on a few bits

$N =$ 23 442 210 895 296 466 551 510 681 543 619 831 978 102 581 799 736 611 246 976 521 590 191 893 224 135 789 025 070 678 051 976 867 349 306 593 332 331 728 775 086 731 364 111 282 889 875 974 451 560 408 740 146 015 934 986 990 476 214 270 640 086 817 425 581 538 170 373 870 259 313 066 583 768 903 697 048 280 641 467 367 411 589 939 100 414 611 356 011 513 397 978 038 218 669 709 747 247 868 727 724 676 001 584 905 770 525 234 976 669 382 895 464 232 871 732 123 454 572 174 833 964 467 804 115 311 936 850 586 791 492 844 973 560 905 229 429 892 438 926 204 188 174 490 543 755 080 972 621 652 831 650 930 277 431 113 028 745 929 593 171 025 639 518 249 955 921 255 776 393 078 247 519 734 666 509 055 776 152 948 501 360 345 202 224 227 559 964 438 653 352 949 732 541 506 721 438 058 592 990 053 089 448 078 211 591

The smaller of the two prime factors

$p =$ 153 108 493 870 511 529 343 183 982 694 581 037 554 816 693 901 893 186 090 279 800 600 449 285 091 109 272 578 071 066 427 336 070 321 693 601 562 274 433 098 580 619 600 099 663 905 410 279 023 148 152 523 939 650 071 615 596 077 413 516 469 321 466 486 454 921 404 568 342 497 216 591 961 439 354 064 844 258 200 738 732 434 241 527 208 989 488 198 329 400 820 115 825 335 921 585 482 389 611 993 667 849 537

The larger of the two prime factors

$q = 153\,108\,493\,870\,511\,529\,343\,183\,982\,694\,581\,037\,554\,816\,693\,901\,893\,186\,090\,279\,800\,600$
$449\,285\,091\,109\,272\,578\,071\,066\,427\,336\,070\,321\,693\,601\,562\,274\,433\,098\,580\,619\,600\,099\,663$
$905\,410\,279\,023\,148\,152\,523\,939\,650\,071\,615\,596\,077\,413\,516\,469\,321\,466\,486\,454\,921\,404\,568$
$342\,497\,216\,591\,961\,439\,354\,064\,844\,258\,200\,738\,732\,434\,241\,527\,208\,989\,488\,198\,329\,400\,820$
$115\,825\,335\,921\,585\,482\,389\,611\,993\,667\,849\,543$

Sleight-of-hand numbers

- We have $|p-q|=6$, making it possible to perform the "factorisation" through a simple integer square root calculation (see later)
- "Instead of waiting for the hardware to improve by yet further orders of magnitude, researchers began inventing better and better tricks for factorising numbers by exploiting their hidden structure" ([10])
- We call numbers such as the Callas Normal form or the small-difference factors from the D-Wave paper *sleight-of-hand numbers*
- Specially designed to make factorisation easy (or feasible) on physics experiments

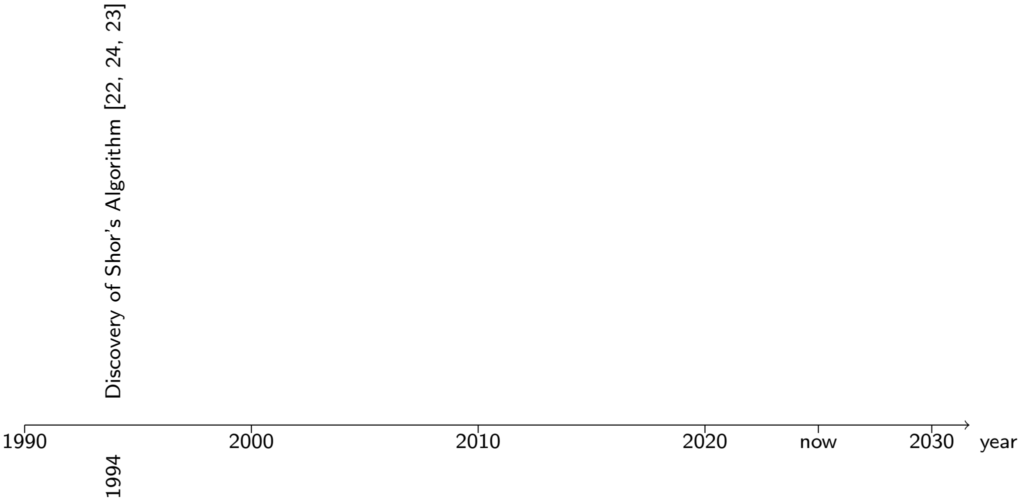
Trick: preprocessing

- Idea: use preprocessing on a computer to transform the value being factorised into an entirely different form or even a different problem
- For example, use compiled/modified form of Shor's algorithm (see above)
- For example, the 2019 quantum factorisation of 1 099 551 473 989 relied on processing with a computer to transform the problem into one that was solveable with a three-qubit circuit [16].
- For example, transforming a factorisation into a minimisation problem allows one to use D-Wave, a quantum annealing machine, not a qubit-based physics experiment [28]. It is not even clear whether a D-Wave has any speed advantage over a conventional computer [4, 17, 26].
- Also called "stunt factorisations"
- Note that the D-Wave paper uses both sleight-of-hand numbers (p close to q) and stunt factorisation techniques (transformation of factorisation into minimisation)

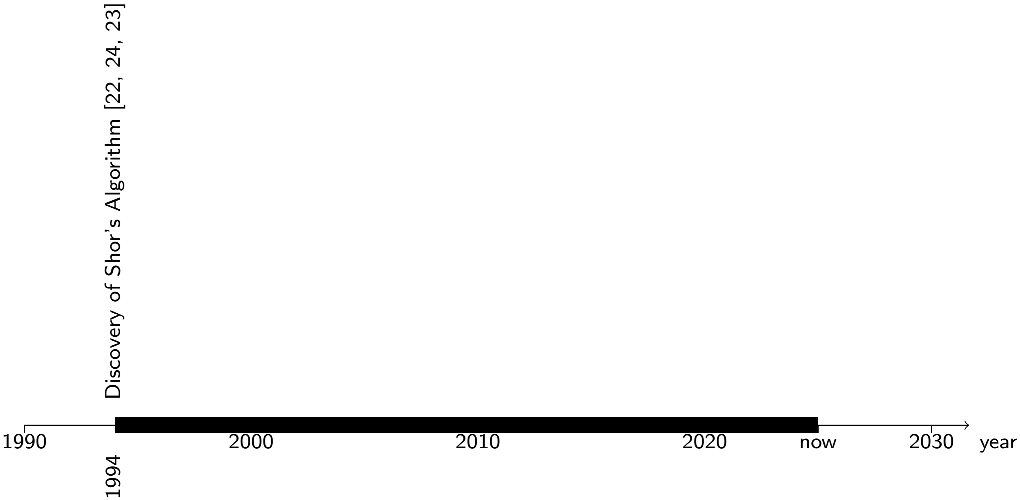
Trick: extension

- For example the main effort in the 2012 factorisation of 143 into 11×13 [31] consisted of finding a value with the special properties required that allowed it to be "factorised" by a physics experiment
- Was extended in 2014 to 56 153 [7]
- Was extended in 2018 to 4 088 459 [6]
- Was extended later in 2018 to 383 123 885 216 472 214 589 586 724 601 136 274 484 797 633 168 671 371 [11]

Corrected timeline of quantum factorisations



Corrected timeline of quantum factorisations



Selecting replication targets

Difficulty of selecting targets

- Published results consist exclusively of stunt factorisations and sleight-of-hand numbers, so how to select targets for replication?
- More or less arbitrarily selected the (to us) least slight-of-handy instances of 15, 21, and 35; and the 2048-bit numbers from the D-Wave paper [28]
- 15, 21, 35 because they were simple proof-of-concept factorisations and did not aspire to be a sensationalist record of any kind (here, taking simple shortcuts is OK)
- "RSA-2048" also because they were called a "wake-up call for cybersecurity" ([26])

Performing Quantum Factorisation operations with a VIC-20

Image source: Wikipedia, public domain

- Very popular 1981 home computer
- 8-bit 6502 CPU at 1 MHz
- 20 KiB ROM, 5 KiB RAM
- Uses transistors (quantum effects!)
- Is therefore as much a "quantum device" as, say, a D-Wave
- Was called "VC-20" in Germany because apparently "VIC" was considered too risqué for the (mostly male) teenagers that were the main target audience

Using a multiplication table to factorise

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	8	10	12	14
3	0	3	6	9	12	15	18	21
4	0	4	8	12	16	20	24	28
5	0	5	10	15	20	25	30	35
6	0	6	12	18	24	30	36	42
7	0	7	14	21	28	35	42	49

How to factor the RSA-2048 moduli?

- Random 2048-bit RSA moduli are way too large to be factorised on a VIC-20 (or any other computer, for that matter)
- But the ten moduli N in the D-Wave paper are not random 2048-bit RSA moduli; they are sleight-of-hand numbers!
- They have been specially chosen so that if $N = pq$, then $|p - q|$ is either 2 or 6
- Key idea: factors p, q will be close to $\lfloor \sqrt{N} \rfloor$
- John von Neumann adapted an integer square root algorithm apparently created for use with an abacus to the EDVAC in 1945 [21]
- Translated by Henry S. Warren, Jr. into modern notation [29, p. 210]
- Needs neither multiplication nor division
- Ideal for implementation on VIC-20 which also has neither instruction

Some assembler code statistics

What	how much
Lines of code[6]	427
Start address	$b000
Code size (text segment)	704 B[7]
RAM requirement	1792 B
Available RAM	about 3.5 KiB
Space enough for WOZMON?	yes
Space enough for MS Basic?	yes
Cost for (bare) 6502 kit	USD 89.00[8]

[6] No comment lines, no empty lines
[7] including 256 bytes for the modulus to be factorised
[8] `https://eater.net/6502`

Correctness and efficiency considerations

#	ticks (μs)
0	16609726
1	16352636
2	16704327
3	16281246
4	16422636
5	16321994
6	16367815
7	16549115
8	16188092
9	16446609

- There were ten moduli (from 0 to 9)
- Our code factorised all of them correctly
- With a 1 MHz machine, this will take roughly 16.5 s
- Running the code on the emulator on a ThinkPad X1 took less than one second
- Including loading the modulus from an input file and writing the factors to output files.

VIC-20 summary

All hail the mighty VIC-20

We have broken, or at least also achieved,[a] all quantum factorisation records, and have additionally replicated a 2025 result with 1981 technology using a 1945 algorithm.

[a] It's hard to tell whether our factorising is faster than D-Wave; the D-Wave paper gives no timing information beyond claiming that factorisation happened in an "extremely short timeframe" (p.1278).

Performing Quantum Factorisation operations with an Abacus

Factorising 15 on an abacus

- Division on an abacus begins at the leftmost (most significant) digit (here: 1)
- The rule for dividing a one digit (in the tens column) is "one by three is three plus one", so our ten becomes a three with the remainder added to the next column along [19]
- We now have the value 36 as shown below, and move on to the next digit, 6.
- The rule for this is "cancel the six, forward two", which means clear the value 6 and add two to the column to the left, which is now 5. So 15 divided by 3 is 5.

Factorising the RSA-2048 numbers on an abacus

- Von Neumann's 1945 square root algorithm used to factor the RSA-2048 numbers was apparently originally created for use with an abacus [14]
- Given a suitably large abacus (at least 616 columns) and enough time, we can also factorise the D-Wave values
- Construction of such a bignum abacus left as an exercise to the reader[9]

[9] Or perhaps to an enthusiastic woodworking hobbyist with a YouTube channel.

Abacus summary

Abaci are great!

We have achieved all quantum factorisation records with an abacus. Factorising the D-Wave moduli is at least plausible, given enough resources.

Performing Quantum Factorisation operations with a Dog

How to factorise with a dog

This morning: "The 2001 and 2012 quantum factorisation records may be easily matched with a dog trained to bark three times" ([13]), an unverified and therefore unscientific claim!

Image source: Peter Gutmann. With permission. Model Release Form available.

- Verification by taking a recently-calibrated reference dog, Scribble(†), and having him bark three times
- Simultaneously factorising both 15 and 21
- Not as simple as it first appeared because Scribble is very well behaved and almost never barks; required having his owner play with him with a ball
- Special performance just for this publication, because he understands the importance of evidence-based science

Image source: Peter Gutmann. With permission. Model Release Form available.

- Verification by taking a recently-calibrated reference dog, Scribble(†), and having him bark three times
- Simultaneously factorising both 15 and 21
- Not as simple as it first appeared because Scribble is very well behaved and almost never barks; required having his owner play with him with a ball
- Special performance just for this publication, because he understands the importance of evidence-based science
- Scribble's contribution to this paper does not rise to the level where he gets co-authorship

Image source: Peter Gutmann. With permission. Model Release Form available.

- Verification by taking a recently-calibrated reference dog, Scribble(†), and having him bark three times
- Simultaneously factorising both 15 and 21
- Not as simple as it first appeared because Scribble is very well behaved and almost never barks; required having his owner play with him with a ball
- Special performance just for this publication, because he understands the importance of evidence-based science
- Scribble's contribution to this paper does not rise to the level where he gets co-authorship
- At the same time, he's not the subject of an experiment, so we don't need IRB approval

Introducing Ripley

Introducing Ripley

Image source: Wikipedia

Introducing Ripley

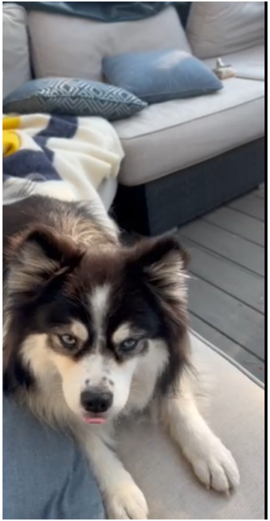


Image source: Ripley's owner. Used with permission.

Introducing Ripley

- I'm sure this Ripley, too, would have liked to nuke the site from orbit

Introducing Ripley

- I'm sure this Ripley, too, would have liked to nuke the site from orbit
- After all, it's the only way to be sure

Dog summary

Dogs are awesome

Canine-based factorisation technology outperforms current physics-experiment based factorisation technology.

Proposed Quantum Factorisation Evaluation Criteria

Need for quantum factorisation evaluation criteria

- *All* demonstrations of quantum factorisation to date have involved either sleight-of-hand numbers, stunt factorisations, or both
- (Factorisations of 15, 21, and the attempted factorisation of 35 were proofs-of-concept, so they get a pass)
- *None* of them pose any danger to RSA whatsoever, notwithstanding any text to the contrary in the paper itself[11]
- In order to be able to judge whether a future physics experiment constitutes a genuine advancement of factorisation capabilities, we propose standard evaluation criteria

[11] For example, "This experiment verifies that the Q[uantum] A[nnealing] algorithm based on D-Wave is an effective method to attack RSA." ([28]) It is not.

Properties of evaluation criteria

- The factors are of a nontrivial size, 64 or 128 bits
- The factors are prime values containing a 50:50 mix of 0 and 1 bits, randomly distributed
- No preprocessing of the value to be factorised on a computer is permitted
- The factors are unknown to the experimenters
- The factorisation is performed on ten different values with the properties given above

As an aside, the above criteria also move the problem out of the space in which it is readily solvable using a VIC-20, an abacus, or a dog.

Future work

Other types of mammal-based quantum factorisation

Image source: Wikipedia, public domain

- Recently, scientists have found evidence of entanglement in other mammals [8]
- This would open up an entirely new research field of mammal-based quantum factorisation
- We hypothesise that the production of fully entangled sheep is easy, given how hard it can be to disentangle their coats in the first place
- The logistics of assembling the tens of thousands of sheep necessary to factorise RSA-2048 numbers is left as an open problem.

Replication Guide

Replicating the work

- VIC-20 software: on Codeberg at `https://codeberg.org/sten13/rsa6502` [20]
- VIC-20 hardware: a number of 6502 kits are available [9, 30, 15]
- Abacus: any standard abacus
- Since only two or three columns are required for the replication of the quantum factorisations of 15, 21, and 35, any abacus of size 9, 11, or 13 columns (digits) may be employed.
- Apparatus for the canine-based factorisation may be obtained from any animal shelter
- Our experiment used a Staffy, but almost any dog breed should be suitable
- Caution: smaller yappy dogs may over-report values

Appendix

Questions?

Where we get these from

- They're all real questions/comments/allegations from responses to the article

“You’ve got an agenda!”

- That’s not a question

“You’ve got an agenda!”

- That’s not a question
- Damn right we have an agenda!

“You’ve got an agenda!”

- That’s not a question
- Damn right we have an agenda!
- At the moment, lots of people are running around like headless chickens, solving a problem that at the moment doesn’t need solving
- Our agenda is to inject some data into that discussion

"You've got an agenda!"

- That's not a question
- Damn right we have an agenda!
- At the moment, lots of people are running around like headless chickens, solving a problem that at the moment doesn't need solving
- Our agenda is to inject some data into that discussion
- Normal cryptography is a mix of maths and engineering: If you make the key random and so long, you can expect to need this long to break it

"You've got an agenda!"

- That's not a question
- Damn right we have an agenda!
- At the moment, lots of people are running around like headless chickens, solving a problem that at the moment doesn't need solving
- Our agenda is to inject some data into that discussion
- Normal cryptography is a mix of maths and engineering: If you make the key random and so long, you can expect to need this long to break it
- Post-Quantum Cryptography isn't engineering, it's augury

"You've got an agenda!"

- That's not a question
- Damn right we have an agenda!
- At the moment, lots of people are running around like headless chickens, solving a problem that at the moment doesn't need solving
- Our agenda is to inject some data into that discussion
- Normal cryptography is a mix of maths and engineering: If you make the key random and so long, you can expect to need this long to break it
- Post-Quantum Cryptography isn't engineering, it's augury
- "A great machine shall arise, and it will cast aside all existing cryptography, there shall be Famine, Plague, War, and a long arable field"

“You’re ignoring incremental progress!”

- Not a question either

"You're ignoring incremental progress!"

- Not a question either
- We're criticising progress that doesn't exist!

“You’re ignoring incremental progress!”

- Not a question either
- We’re criticising progress that doesn’t exist!
- We’ve also made about 2000 years of “incremental progress” on the whereabouts of Legio IX Hispana, but we still don’t know what happened

But integer factorisation is not what QC is all about!

- Still not a question

But integer factorisation is not what QC is all about!

- Still not a question
- That's odd because whenever we read something about QC, the first thing we read is that the world is about to end because of it

But at least the switch to PQC can do no harm!

- Yup, still not a question!

But at least the switch to PQC can do no harm!

- Yup, still not a question!
- First, cryptography is not where current security problems lie

But at least the switch to PQC can do no harm!

- Yup, still not a question!
- First, cryptography is not where current security problems lie
- When was the last time you read about some data theft or break-in because someone had broken the crypto? Exactly. (Not talking about key management problems, these are real, as are problems with hard-to-securely-use APIs that permit IV reuse and whatnot.)

But at least the switch to PQC can do no harm!

- Yup, still not a question!
- First, cryptography is not where current security problems lie
- When was the last time you read about some data theft or break-in because someone had broken the crypto? Exactly. (Not talking about key management problems, these are real, as are problems with hard-to-securely-use APIs that permit IV reuse and whatnot.)
- Diverting attention and funds away from solving these real problems will make sure that these real problems persist far longer

But at least the switch to PQC can do no harm!

- Yup, still not a question!
- First, cryptography is not where current security problems lie
- When was the last time you read about some data theft or break-in because someone had broken the crypto? Exactly. (Not talking about key management problems, these are real, as are problems with hard-to-securely-use APIs that permit IV reuse and whatnot.)
- Diverting attention and funds away from solving these real problems will make sure that these real problems persist far longer
- Second, all of this PQC stuff is untested in the real world. Remember how long it took to get TLS right, to get even AES right (because of timing attacks on software-only implementations!)?

But at least the switch to PQC can do no harm!

- Yup, still not a question!
- First, cryptography is not where current security problems lie
- When was the last time you read about some data theft or break-in because someone had broken the crypto? Exactly. (Not talking about key management problems, these are real, as are problems with hard-to-securely-use APIs that permit IV reuse and whatnot.)
- Diverting attention and funds away from solving these real problems will make sure that these real problems persist far longer
- Second, all of this PQC stuff is untested in the real world. Remember how long it took to get TLS right, to get even AES right (because of timing attacks on software-only implementations!)?
- We're bound to make some rather spectacular mistakes when we rip out all the things that we've built over the last decades and that now work

References

References (1)

[1] Mirko Amico, Zain Saleem, and Muir Kumph. "An experimental study of Shor's factoring algorithm using the IBM Q Experience". In: *Physical Review A* 100.1 (July 8, 2019). DOI: 10.1103/PhysRevA.100.012305. URL: https://link.aps.org/doi/10.1103/PhysRevA.100.012305.

[2] Jon Callas. *Re: [Cryptography] Has quantum cryptanalysis actually achieved anything?* Posting BFC54170-DDC4-4292-9A75-377B6D85406B@callas.org on the cryptography@metzdowd.com mailing list. Feb. 19, 2025.

[3] Abel C. H. Chen. *Implementation of Shor Algorithm: Factoring a 4096-Bit Integer Under Specific Constraints*. Apr. 7, 2025. DOI: 10.48550/arXiv.2505.03743. URL: https://arxiv.org/abs/2505.03743.

[4] Adrian Cho. *Quantum or not, controversial computer runs no faster than a normal one*. June 19, 2014. URL: https://www.science.org/content/article/quantum-or-not-controversial-computer-runs-no-faster-normal-one.

References (2)

[5] Leah Crane. "Quantum computer sets new record for finding prime number factors". In: *New Scientist* (Dec. 13, 2019). URL: https://www.newscientist.com/article/2227387-quantum-computer-sets-new-record-for-finding-prime-number-factors.

[6] Avinash Dash et al. *Exact search algorithm to factorize large biprimes and a triprime on IBM quantum computer*. May 26, 2018. URL: https://arxiv.org/abs/1805.10478.

[7] Nikesh Dattani and Nathaniel Bryans. *Quantum factorization of 56153 with only 4 qubits*. Nov. 25, 2014. URL: https://arxiv.org/abs/1411.6758.

[8] Naomi Dinmore. *CERN scientists find evidence of quantum entanglement in sheep*. Apr. 1, 2025. URL: https://home.cern/news/news/cern/cern-scientists-find-evidence-quantum-entanglement-sheep.

[9] Ben Eater. *Build a 6502 computer*. URL: https://eater.net/6502.

References (3)

[10] Craig Gidney. *Factoring the largest number ever with a quantum computer*. (The date refers to the "factorisation" of a six thousand digit number in the latter part of the writeup, not the analysis portion.) Apr. 1, 2020. URL: https://algassert.com/post/2000.

[11] François Grieu. *Largest integer factored by Shor's algorithm?* June 2018. URL: https://crypto.stackexchange.com/questions/59795/largest-integer-factored-by-shors-algorithm/59796#59796.

[12] François Grieu. *Largest integer factored by Shor's algorithm?* Jan. 5, 2023. URL: https://crypto.stackexchange.com/questions/59795/largest-integer-factored-by-shors-algorithm/59796.

[13] Peter Gutmann. *Why quantum cryptanalysis is bollocks: A lesson from history*. 2024. URL: https://www.cs.auckland.ac.nz/~pgut001/pubs/bollocks.pdf.

References (4)

[14] Martin Guy. *Square root by abacus algorithm*. This references a publication by a Mr.C.Woo who is listed as the co-author of a later edition of the abacus book referenced here also. June 1985. URL: http://medialab.freaknet.org/martin/src/sqrt/sqrt.c.

[15] Aleksander Kamiński. *Pocket265*. URL: https://github.com/agkaminski/Pocket265.

[16] Amir Karamlou et al. "Analyzing the performance of variational quantum factoring on a superconducting quantum processor". In: *Quantum Information* 7 (2021). Article No. 156.

[17] Chris Lee. *Is D-Wave's quantum processor really* 10^8 *times faster than a normal computer?* Feb. 16, 2016. URL: https://arstechnica.com/science/2016/02/is-d-waves-quantum-processor-really-10%E2%81%B8-times-faster-than-a-normal-computer/.

References (5)

[18] Enrique Martín-López et al. "Experimental realization of Shor's quantum factoring algorithm using qubit recycling". In: *Nature Photon* 6 (2012), pp. 773–776. DOI: 10.1038/nphoton.2012.259.

[19] Kwa Tak Ming. *Fundamental Operations in Bead Arithmetic*. 1922. URL: https://archive.computerhistory.org/resources/access/text/2016/12/B1671.01-05-01-acc.pdf.

[20] Stephan Neuhaus. *Code to factor the 2048-bit RSA "moduli"*. URL: https://codeberg.org/sten13/rsa6502.

[21] John von Neumann. *First Draft of a Report on the EDVAC*. Tech. rep. University of Pennsylvania, June 30, 1945.

[22] Peter Shor. "Algorithms for quantum computation: discrete Logarithms and Factoring". In: *Proceedings of the 35th Annual IEEE Symposium on the Foundations of Computer Science*. IEEE Press, 1994, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.

References (6)

[23] Peter W Shor. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer". In: *SIAM Journal on Computing* 26.5 (Oct. 1997), pp. 1484–1509. DOI: 10.1137/S0097539795293172.

[24] Peter W. Shor. "Discrete logarithms and factoring". In: *Proceedings of the 35th Annual IEEE Symposium on the Foundations of Computer Science*. 1994, p. 124.

[25] John Smolin, Graeme Smith, and Alex Vargo. *Pretending to factor large numbers on a quantum computer*. Jan. 29, 2013. URL: https://arxiv.org/pdf/1301.7007.

[26] Kalab Tenadeg. *Chinese Researchers Break RSA Encryption with Quantum Computer: A Wake-Up Call for Cybersecurity*. Oct. 19, 2024. URL: https://medium.com/@kalabtenadeg/chinese-researchers-break-rsa-encryption-with-quantum-computer-a-wake-up-call-for-cybersecurity-813247dd9585.

References (7)

[27] Lieven Vandersypen et al. "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance". In: *Nature* 414 (2001), pp. 883–887. DOI: 10.1038/414883a.

[28] Chao Wang et al. "A First Successful Factorization of RSA-2048 Integer by D-Wave Quantum Computer". In: *Tsinghua Science and Technology* 30.3 (June 2025), pp. 1270–1282. DOI: 10.26599/TST.2024.9010028.

[29] Henry Warren Jr. *Hacker's Delight*. Addison-Wesley, 2002.

[30] Western Design Centre. *W65C134SXB — 6502 based Microcomputer Board*. URL: https://wdc6502store.com/products/w65c134sxb-single-board-computer.

[31] Nanyang Xu et al. "Quantum Factorization of 143 on a Dipolar-Coupling NMR system". In: *Physical Review Letters* 108 (Mar. 30, 2012). URL: https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.108.130501.

[32] Christof Zalka. *Shor's algorithm with fewer (pure) qubits*. 2006. arXiv: quant-ph/0601097 [quant-ph]. URL: https://arxiv.org/abs/quant-ph/0601097.