Why
Quantum Cryptanalysis
is Bollocks

Peter Gutmann, Empirical Gnostic

University of Auckland

A Lesson from History

Schwerer Gustav, proposed 1935, ready for use in 1942

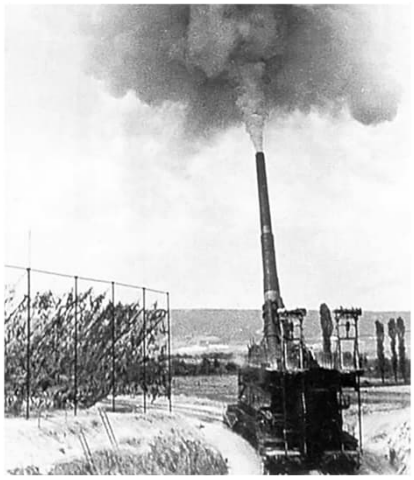
- Was intended to be used against the Maginot line in March 1940 but wasn't ready in time

A Lesson from History (ctd)

This was the headline-grabbing attack of 80 years ago

- Weighed 1,350 tons
- Could fire a 5-ton shell around 50km
- Left a crater 10m wide and deep

This is where all the action was



A Lesson from History (ctd)

Everyone who was anyone wanted to be associated with it

A Lesson from History (ctd)

Carried in a 1.5km long train with 25 freight cars

- Just the gun, supplies and crew had their own trains

Took 2,000 men (one report) / 4,000 men (another report) / 4,500 men (yet another report) to get into operation over a period of five weeks

- Required twin sets of specially reinforced railway tracks

Had two flak battalions to defend it

Fired around 50 shells in total on Sevastopol on five different days

- Lots of conflicting reports about some of these totals

A Lesson from History (ctd)

This was a considerable net loss for the war effort

- Drew significant resources *away* from the main attack

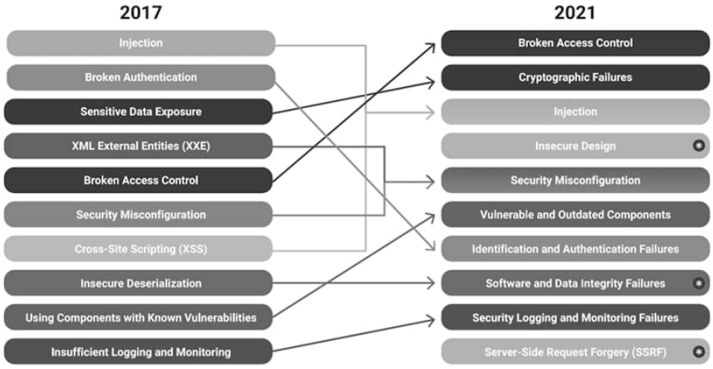
Same could have been achieved by a handful of aircraft

- The gun actually had an entire squadron of Fi 156 spotter aircraft to direct fire and observe results
  - Light aircraft but could carry bombs — just
- The means to get the boom! from source to destination was already in place and didn't involve a giant gun
  - In any case Röchling shells from conventional artillery would have had much the same effect

Surely we wouldn't still be doing the same thing today?

What are the Threats?

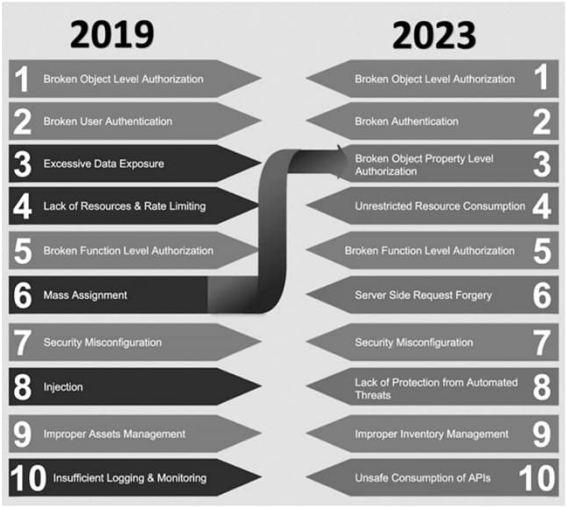
In the security field we have good data on where the problems are



OWASP (Open Source Foundation for Application Security) top 10, last two revisions

What are the Threats? (ctd)

These exist for various different targets, e.g. APIs



What are the Threats? (ctd)

The results are remarkably stable over time

Project Information

- OWASP Top 10:2021
- Making of OWASP Top 10
- OWASP Top 10:2021 - 20th Anniversary Presentation (PPTX)

Flagship Project
Documentation
Builder
Defender

- Previous Version (2017)

A lot of the changes are just naming or classification updates

- The underlying problems remain the same

What are the Threats? (ctd)

For a full breakdown of what's changed…

Comparison of 2003, 2004, 2007, 2010 and 2013 Releases

OWASP Top Ten Entries (Unordered)	Releases				
	2003	2004	2007	2010	2013
Unvalidated Input	A1	A1[9]	×	×	×
Buffer Overflows	A5	A5	×	×	×
Denial of Service	×	A9[2]	×	×	×
Injection	A6	A6[3]	A2	A1[10]	A1
Cross Site Scripting (XSS)	A4	A4	A1	A2	A3
Broken Authentication and Session Management	A3	A3	A7	A3	A2
Insecure Direct Object Reference	×	A2	A4[11]	A4	A4
Cross Site Request Forgery (CSRF)	×	×	A5	A5	A8
Security Misconfiguration	A10	A10[3][5]	×	A6	A5
Missing Functional Level Access Control	A2	A2[1]	A10[13]	A8	A7[16]
Unvalidated Redirects and Forwards	×	×	×	A10	A10
Information Leakage and Improper Error Handling	A7	A7[14][4]	A6	A6[8]	×
Malicious File Execution	×	×	A3	A6[8]	×
Sensitive Data Exposure	A8	A8[6][5]	A8	A7	A6[17]
Insecure Communications	×	A10	A9[7]	A9	×
Remote Administration Flaws	A9	×	×	×	×
Using Known Vulnerable Components	×	×	×	×	A9 [18][19]

[1] Renamed "Broken Access Control" from T10 2003
[2] Split "Broken Access Control" from T10 2003
[3] Renamed "Command Injection Flaws" from T10 2003
[4] Renamed "Error Handling Problems" from T10 2003
[5] Renamed "Insecure Use of Cryptography" from T10 2003
[6] Renamed "Web and Application Server " from T10 2003
[7] Split "Insecure Configuration Management" from T10 2004
[8] Reconsidered during T10 2010 Release Candidate (RC)
[9] Renamed "Unvalidated Parameters" from T10 2003
[10] Renamed "Injection Flaws" from T10 2007
[11] Split "Broken Access Control" from T10 2004
[12] Renamed "Insecure Configuration Management" from T10 2004
[13] Split "Broken Access Control" from T10 2004
[14] Renamed "Improper Error Handling" from T10 2004
[15] Renamed "Insecure Storage" from T10 2004
[16] Renamed "Failure to Restrict URL Access" from T10 2010
[17] Renamed "Insecure Cryptographic Storage" from T10 2010
[18] Split "Insecure Cryptographic Storage" from T10 2010
[19] Split "Security Misconfiguration" from T10 2010

What gets the Attention?

Consulting the OWASP top 100,000, from the Appendix to the Addendum to the Supplement to the Apocrypha, Volume 127, we see…

…
#17,245 Spectre
#17,246 POODLE
#17,247 Meltdown
#17,248 Rowhammer
#17,249 DROWN
#17,250 ROCA
….

What do all of these have in common?

What gets the Attention? (ctd)

No-one ever uses them

- There are 17,244 easier ways to carry out an attack
- This is why they've been referred to as "stunt cryptography"

Stunt cryptography attack

- You have a 0.00001% change of recovering 2 bits of plaintext from a single message

Any of the OWASP top ten

- You have a 100% chance of recovering the plaintext of all the messages

What gets the Attention? (ctd)

People really like fancy headline-grabbing (but eminently impractical) things

- Are there any known cases of a real-life attacker ever using Spectre, Rowhammer, POODLE, or other stunt cryptography?
- (To date no-one in the audience has ever identified one)

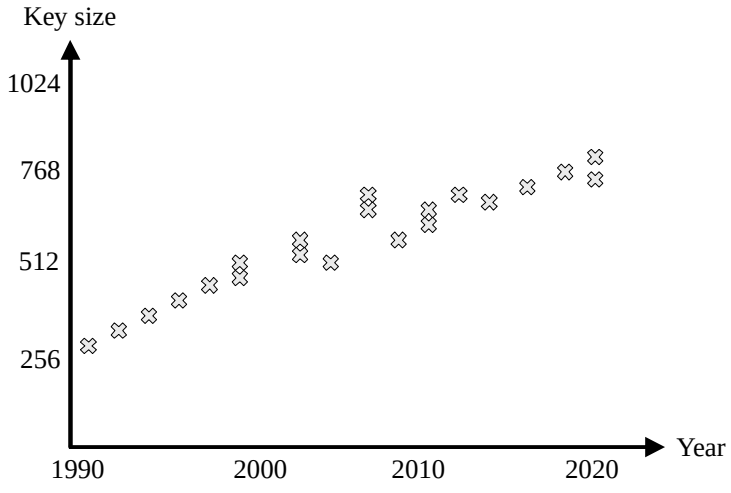
Focusing on high-profile attacks that no-one uses has a similar effect to obsessing over superguns

- Draws resources away from the real goal, the actual attacks that are happening

Only when you've fixed the top ten are you allowed to look at the fancy named attacks on crypto, side-channels, etc

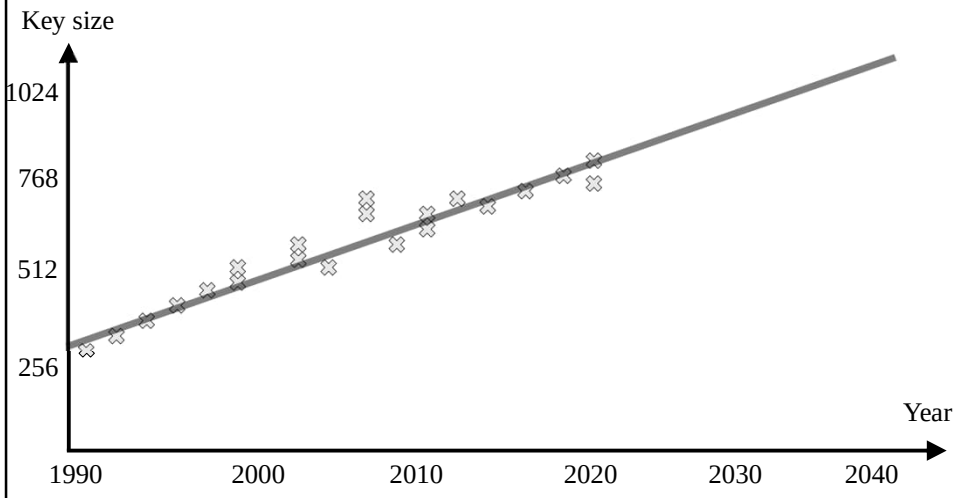
Ignoring Measurements

There are other cases where we also have very good measurements, e.g. RSA key sizes (factoring)



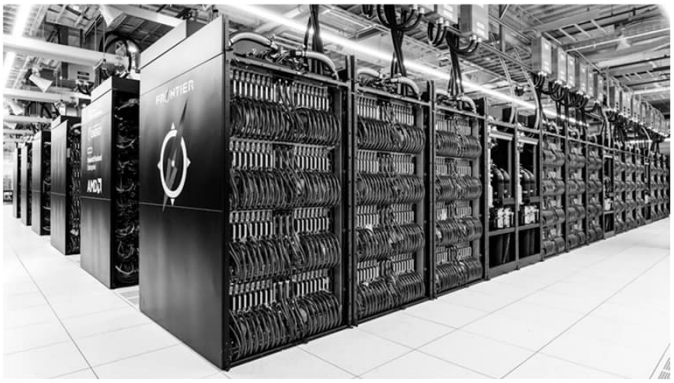
Ignoring Measurements (ctd)

From which we can extrapolate…



Ignoring Measurements (ctd)

But wait, we can (theoretically) break 1024-bit keys today



Yup, with one of these

- Takes around a year's work to factor a 1024-bit RSA key on this class of machine

Ignoring Measurements (ctd)

Let's explore this a bit...

NSA employee: There's a 1024-bit key I'd like to factor

NSA boss: Tell me more

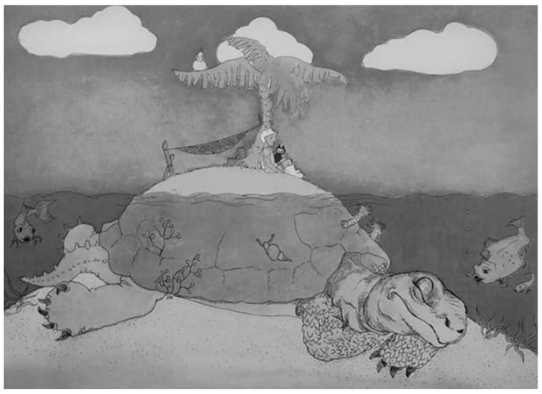
NSA employee: It's pretty straightforward, we just need to shut down Los Alamos (Oak Ridge, LLNL, whatever) for a year to do it

NSA boss: *Makes note to ping HR about their employee mental health screening procedures*

Ignoring Measurements (ctd)

Making it more applicable to individuals...

- I give you a black box that will factor a 1024-bit key in a year
- To prove your dedication to the task, you agree to live on a desert island for the time it takes
  - No Internet, TV, radio
  - No companions
- Monthly airdrop of a months' worth of canned baked beans and a replacement butane cartridge



Ignoring Measurements (ctd)

Who would accept this offer?

Is there any known 1024-bit key worth attacking?

- Informal polling to date hasn't indicated any known 1024-bit key that's worth attacking, whether by shutting down Los Alamos or becoming a hermit for a year

Ignoring Measurements, Example 1

Perhaps the absence of rational attacks is why some organisations switched to numerology

- Arithmancy for Harry Potter fans

Date	Security Strength	Symmetric Algorithms	Factoring Modulus	Discrete Logarithm Key	Discrete Logarithm Group	Elliptic Curve	Hash (A)	Hash (B)
Legacy (1)	80	2TDEA	1024	160	1024	160	SHA-1 (2)	
2019 - 2030	112	(3TDEA) (3) AES-128	2048	224	2048	224	SHA-224 SHA-512/224 SHA3-224	
2019 - 2030 & beyond	128	AES-128	3072	256	3072	256	SHA-256 SHA-512/256 SHA3-256	SHA-1 KMAC128
2019 - 2030 & beyond	192	AES-192	7680	384	7680	384	SHA-384 SHA3-384	SHA-224 SHA-512/224 SHA3-224
2019 - 2030 & beyond	256	AES-256	15360	512	15360	512	SHA-512 SHA3-512	SHA-256 SHA-512/256 SHA-384 SHA-512 SHA3-256 SHA3-384 SHA3-512 KMAC256

Source: NIST

Ignoring Measurements, Example 1 (ctd)

Where do these figures come from?

The practical limits on achievable computation are around 2^{110} or so

- For reference, the entire global Bitcoin hash rate is 2^{94} per year
  - This is not the same as key brute-forcing, but serves as a proxy

This means keys for 3DES (112 bits), AES-128 (128 bits), AES-192 (192 bits), and AES-256 (256 bits) are all equally out of reach

- However, numerology requires that we treat them as distinct

Ignoring Measurements, Example 1 (ctd)

For symmetric crypto, each bit added doubles the work factor

- For asymmetric crypto, doubling the work factor isn't nearly as simple

To match each (irrelevant) size difference in symmetric crypto keys, we need corresponding huge size increases in asymmetric crypto keys

Ignoring Measurements, Example 1 (ctd)

Forget large-size asymmetric keys, we need ludicrous-size keys to match the (irrelevant) symmetric work-factor doubling

- 15,360 bits, go!

Ignoring Measurements, Example 2

But wait, there's a better one!

The first quantum factorisation was done in 2001

- It factored the number 15
- Not a 15-digit number
- Not even a 15-bit number
- The product of 3×5
- The same could be achieved with a dog trained to bark three times

The next record was set in 2012

- The number factored was 21, 3×7
- The same dog was used to match this new record

Ignoring Measurements, Example 2

Another attempt was tried in 2019

- The attempted factorisation was of 35, 5×7
- It failed

Since then there have been no new factorisation records using Shor's Algorithm

- There have been records announced for a range of special-case numbers
- One case involved taking a known factorisation and working backwards to create a quantum physics experiment for it
- In another case there was uncertainty over what had actually been factored

Ignoring Measurements, Example 2 (ctd)

The scientific breakthrough in all of these cases was in finding techniques to manufacture values that could then be "factored" by simple quantum physics experiments

Standard technique employed for this

- Manufacture a small value that can be "factored" by a physics experiment
- In later papers figure out how to stretch the value to more digits that the same physics experiment can "factor"

These techniques have been termed "stunt factorisations" (François Grieu)

- An in-progress paper will examine this in more detail

Ignoring Measurements, Example 2 (ctd)

Even the factorisation of 15 and 21 took advantage of special tricks

- Knowing the factors in advance allowed the application of the "compiled Shor's algorithm"

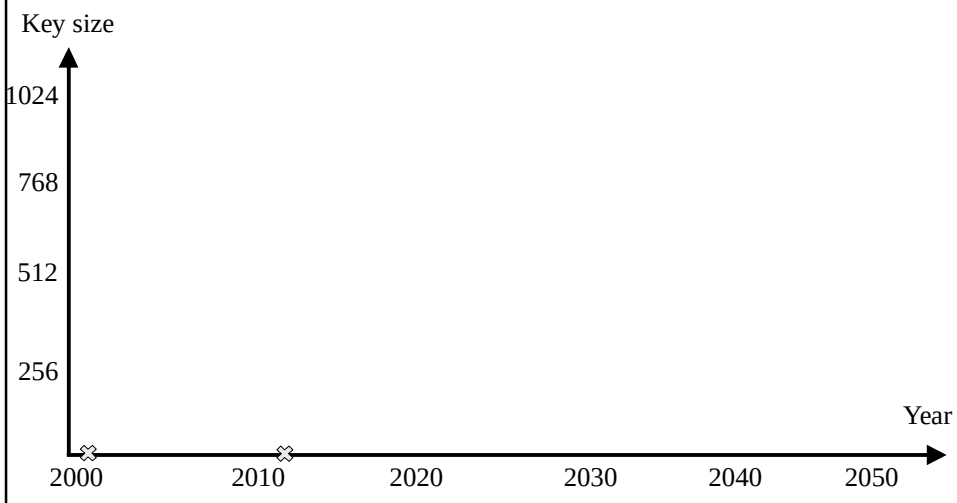
It is not legitimate for a compiler to know the answer to the problem being solved. To even call such a procedure compilation is an abuse of language

- "Pretending to factor large numbers on a quantum computer"

In any case we have the necessary two (!!) data points to draw a line on a graph

Ignoring Measurements, Example 2 (ctd)

Quantum cryptanalysis (factoring)

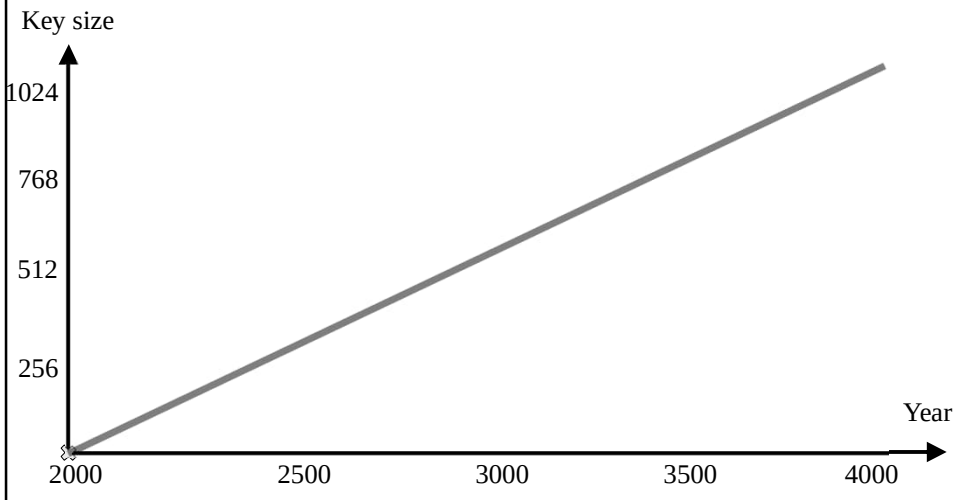


Ignoring Measurements, Example 2 (ctd)

We're gonna need a bigger ~~boat~~ graph

Ignoring Measurements, Alternative 2 (ctd)

Quantum cryptanalysis (factoring)



Ignoring Measurements, Example 2 (ctd)

Disclaimer: This makes the highly optimistic assumption that quantum physics experiments scale linearly

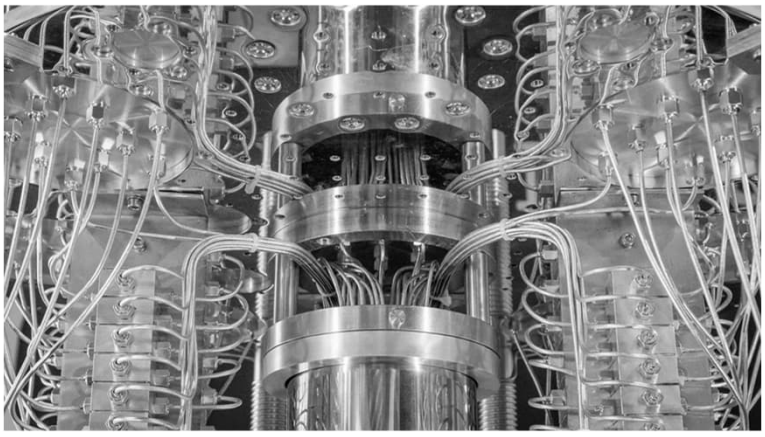
- We have no evidence that this is the case
- The evidence we have, shown by the lack of progress so far, is that this is not the case

> Our stopped clock technology is still in its infancy, but it's already reached an accuracy rate of two or more times per day, and there's no reason for us to believe that won't improve dramatically in the future
>
> — Joe Groff

In any case, in a mere two thousand years a physics experiment may be able to achieve what a conventional computer can do today

Physics Experiment?

Note the use of the term "physics experiment"



These are physics experiments, not computers

Physics Experiment? (ctd)

Claiming that it's a computer misrepresents what we're really working with

Takes advantage of the Heisenberg-Schrödinger Credulity Effect

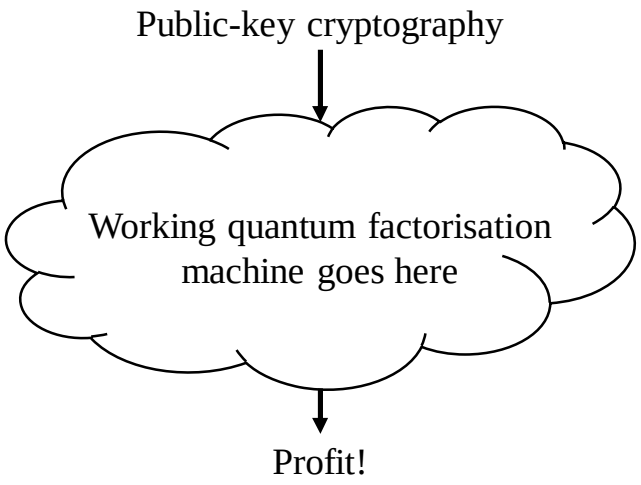
> The word "quantum" sucks people's brains out, and otherwise sensible people suffer from impaired reasoning
>
> — Jon Callas

- Should really be the Schrödinger-Heisenberg Credülity Effect
- We need more metal umlauts in crypto

Every time you see "quantum computer" mentally substitute "physics experiment", which is what's actually being discussed

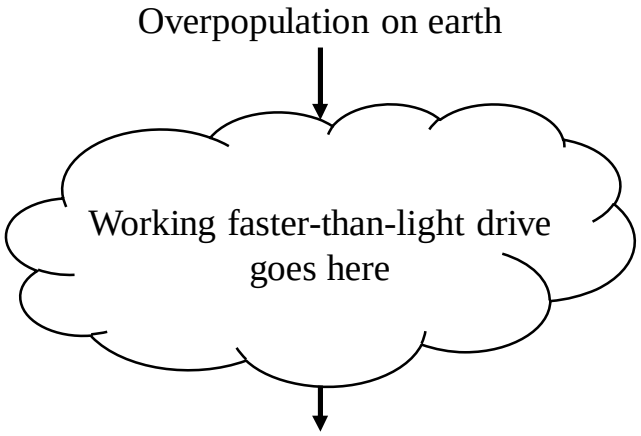
Physics Experiments

How does a physics experiment break crypto?



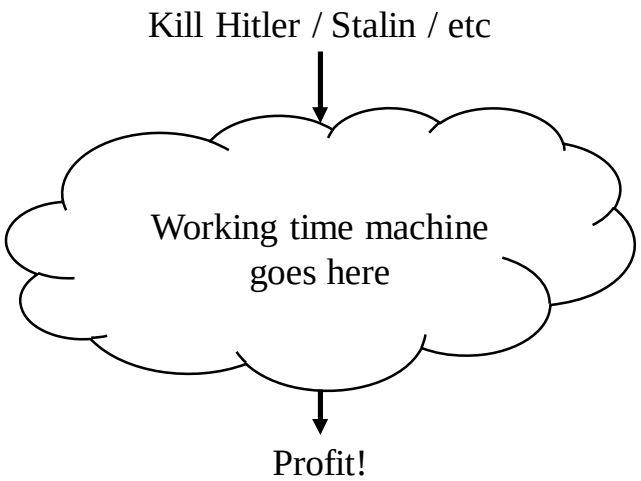
Physics Experiments (ctd)

This applies to any number of other things as well, e.g. colonising distant galaxies



Physics Experiments (ctd)

The possibilities are endless



Physics Experiments (ctd)

Quantum physics pioneer Wolfgang Pauli would have loved this stuff

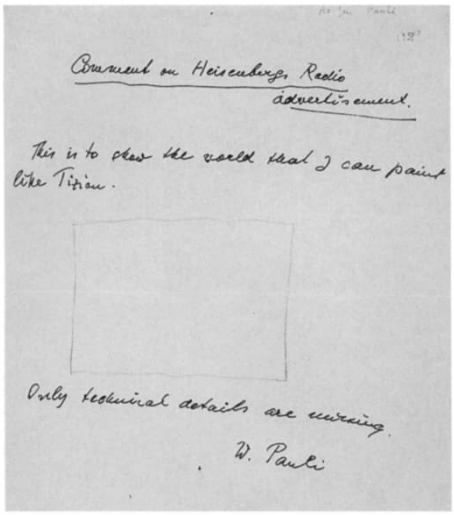
This is to show the world I can paint like Titian.

Only technical details are missing.

could become

This is to show the world a quantum factorisation machine.

Only practical details are missing.



Physics Experiments (ctd)

Evidence for the Schrödinger-Heisenberg Credülity Effect

- When you say "Working time machine goes here" it's just being silly
- When you say "Working quantum factorisation machine goes here" it's dead serious

QED

Re-examining the Physics Experiment

Remember those factorisation records?

- They "factored" two carefully-chosen numbers with the results known in advance
- Sleight-of-hand numbers
  - (There isn't a real name for such a thing in cryptography because the attacker isn't supposed to be able to chose the answer in advance and then create the "experiment" to produce the required answer. This is my suggested candidate name).

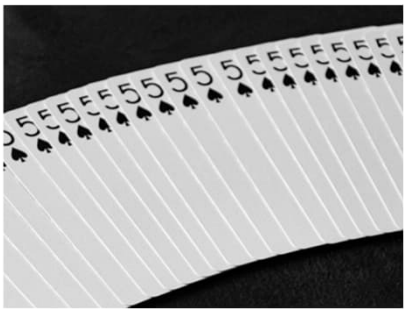
To date there has never been a physics-experiment factorisation of a non-sleight-of-hand number

- This method is the stock-in-trade of stage magicians

Re-examining the Physics Experiment (ctd)

Card trick:

1. "Pick a card any card"

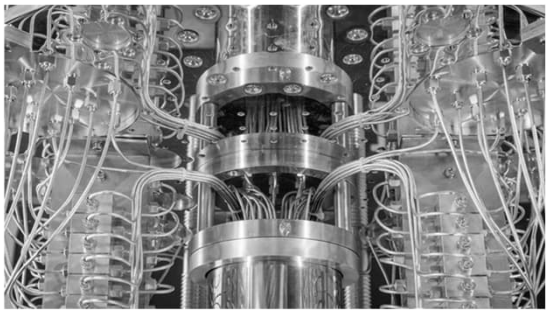


2. Lots of smoke and mirrors to distract the audience
3. "Is it the Five of Spades?"

Re-examining the Physics Experiment (ctd)

Quantum physics trick:

1. "Pick an integer greater than 14 and less than 16"



2. Lots of smoke and mirrors to distract the audience
3. "Is it 3 x 5?"

Re-examining the Physics Experiment (ctd)

Quantum cryptanalysis has only ever factored sleight-of-hand numbers

Extreme example: The D-Wave factorisation of an RSA 2048-bit integer

- The distance p-q between the factors in the samples was either 2 (a prime pair) or 6
- To factor this, take the square root and guess one single bit
- This can be done on one of these

Post Physics-experiment Cryptography

One option is Lattice-based cryptography

- Proposed 30 years ago

Never used because it wasn't very good

- Incredibly inefficient space-wise
  - Up to a factor of 1,000 times larger
- Vaguely interesting mathematically, sporadic papers published

It's probably physics-experiment proof

- Unless someone says otherwise in the future

We could perhaps use the time machine from a previous slide to look ahead and see if it's still OK

Post Physics-experiment Cryptography (ctd)

It's probably secure

- Unless someone says otherwise in the future
- Nearly half of all NIST PQC candidates have already been broken

Very little operational experience with it

- If the history of every other PKC is anything to go by, expect decades of vulnerabilities and attacks

Why are we Fixated on This?

This is Scribble

Scribble can bark five times

This makes him more capable than the world's most powerful factorisation physics experiment

Why are we Fixated on This? (ctd)

Nevertheless, our reaction to this data has been…

Why are we Fixated on This? (ctd)

To understand this, let's look at subprime mortgages

- House buyers / investors were practically given houses (Ninja mortgages)
- Mortgage brokers were earning large commissions
- Fannie Mae and Freddie Mac got plaudits for assisting low-income earners into housing
- Retail banks made money selling mortgages to investment banks, converting liability to cash assets
- Investment banks bought mortgage agreements from retail banks, bundled the mortgages into mortgage-backed securities (MBS) and sold them to investors

…continues…

Why are we Fixated on This? (ctd)

…continued…

- MBS investors made money from the payments from mortgage holders
  - This was a good scheme when creditworthy borrowers were involved
  - When those ran out, banks magicked AAA-rated mortgages from subprime mortgages via collateralised debt obligations and kept on issuing mortgages
- Insurance companies made money insuring the mortgages while magicking protection from problems via credit default swaps

…continues…

Why are we Fixated on This? (ctd)

…continued…

- Credit rating agencies were paid huge fees to bless the whole thing

Nobody in the entire food chain had the slightest motivation to push the emergency stop

- All the data was there
- No-one had any motivation to look at the data because they were too busy making money

Why are we Fixated on This? (ctd)

Pop quiz: Which one of these would you choose?

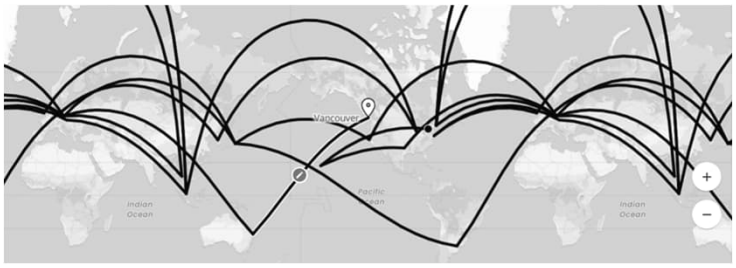
Academics

A. Publish yet another paper on group key management that no-one reads
B. Publish a paper on a cool new post-physics-experiment algorithm

Why are we Fixated on This? (ctd)

Standards groups

A. Standardise away at yet another TLS extension that no-one apart from the sponsoring company cares about
B. Fly from one exotic location to another and argue over which post-physics-experiment algorithm is the most cromulent



- Recent IETF meetings were held in Bangkok, Brisbane, Buenos Aires, Dublin, Madrid, Montreal, Prague, Seoul, Vienna, Yokohama
- It's a great job if you can get it

Why are we Fixated on This? (ctd)

Developers

A. Audit existing code for problems
B. Implement a new post-physics-experiment algorithm that a standards group is still arguing over

Journalists

A. Write about this week's PHP vulnerability
B. Announce quantum supremacy or the quantocalypse for the 37th time in a row

Why are we Fixated on This? (ctd)

Hands up all those who chose ‘B’ on each one

- Nobody wants ‘A’, the status quo, because ‘B’ is much more fun

As with subprime mortgages, nobody involved has any incentive to stop the merry-go-round

- If the merry-go-round stops, everyone has to go back to doing the boring stuff

Why is This a Problem?

Fixating on unrealistic attacks draws significant resources away from solving the real problems that we’re facing

- The endless churn and added complexity then creates *more* problems

Given the relatively unproven nature of lattice-based crypto, we may need to churn again in the future

Why is This a Problem? (ctd)

Actually we'll need to churn anyway no matter how lattice-based crypto turns out

> Future adoption of these algorithms is likely inevitable even if a quantum computer is never built […] opening the door to decades of new research in cryptanalysis
>
> — "The State of the Art in Integer Factoring and Breaking Public-Key Cryptography", Boudot et al.

Software security designers and standards people thrive on churn

Why is This a Problem? (ctd)

Something you'll never hear in any security protocol / standards group discussion ever:

> OK, we're all done now

Even standards groups that have been explicitly shut down just continue by other means

- Formal: PKIX carries on as LAMPS
- Semi-formal: PGP (openpgp) just keeps going and going and going and going
- Informal: SSH (secsh) carries on as OpenSSH inventions, `https://cvsweb.openbsd.org/src/usr.bin/ssh/PROTOCOL`

Why is This a Problem? (ctd)

Getting back to the stock market analogy…

You can make money when the market is going up or going down. You can't make money when prices are constant

- The whole stock market system is designed to have churn
- Churn means brokers make money

In crypto, churn means…

- Academics can publish papers
- Implementers have something to hack away at
- Vendors have something new to sell to customers

Churn is good for everyone except those primarily concerned about security

Why is This a Problem? (ctd)

Source: Alastair Miller, Aura

Churn is complexity serialised

- Standard complexity is everything up-front
- Churn adds more pieces of complexity every few months

This turns the already bad-enough complexity problem into the even worse Red-Queen complexity problem

Why is This a Problem? (ctd)

The TLS protocol alone has

- 60 RFCs
  - No, that's not an error, *sixty* RFCs
- 32 further RFC drafts in progress

That's just under *two thousand pages* of standards documents

- This is what it would look like if printed

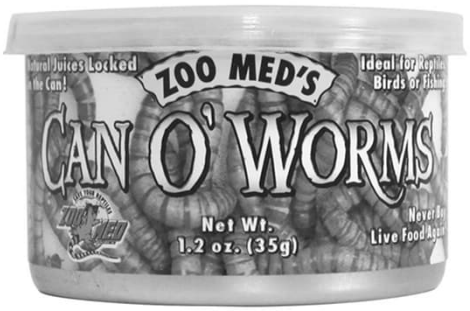
Does anyone seriously think there aren't reams of vulnerabilities hidden in this enormous complexity?

Why is This a Problem? (ctd)

Complexity is the enemy of security

- The more complexity you have, the more scope there is for vulnerabilities

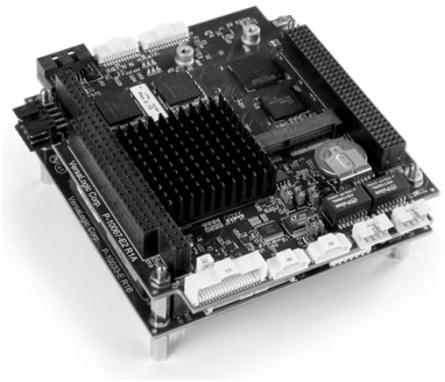
Constant churn adds more complexity and unexpected emergent properties



Why is This a Problem? (ctd)

Some of the most secure systems I've audited were created by (non-security-geek) embedded systems engineers

- Bare-bones TCP stack with no options
- TLS with one single cipher suite and no options
- Certificate management via `memcpy()`



There's simply nothing there to attack

> Best block, no be there
> — Nariyoshi Miyagi

Conclusion

Something similar to quantum cryptanalysis has happened in theoretical physics with string theory

- Non-falsifiable
  - Can't generate any testable predictions
- Drew significant resources away from other physics research for at least two decades

String theory has, however, been spectacularly successful on one front — public relations
— Peter Woit, Columbia University

Conclusion (ctd)

Quantum cryptanalysis is the string theory of security

- String theory has never generated a single testable prediction
- Quantum cryptanalysis has never factored a single non-sleight-of-hand number

Quantum Cryptanalysis

Magical thinking says it's a serious threat

Empirical data says its bollocks

Woof, woof, woof, woof, woof!

Ignoring bad ideas doesn't make them go away; they will still eat up funding. […] Killing ideas is a necessary part of science. Think of it as a community service

— Sabine Hossenfelder, "Lost in Math"

Notes

Some notes for people reading the slides, the talk itself contains more details that aren't explicitly written down in the slides…

- Schwerer Gustav means "Heavy Gustav", named after Gustav von Krupp, the gun being a Krupp product.
- The aircraft that were used with the gun were Fieseler Fi 165 "Storch" (stork) spotter aircraft, notable for being able to take off and land in places nothing else could, for example on a rocky mountaintop if you wanted to rescue an Italian dictator being held there, and fly at treetop height below the stall speed of the aircraft attacking them. They could in theory carry a small bomb load and thus also in theory could have "got the boom from A to B", although in practice you'd use almost anything else for the job.

Notes (ctd)

- Röchling shells were what today would be called bunker-buster shells, fin-stabilised discarding-sabot subcalibre munitions with a length measured in metres that could penetrate ten metres of solid rock and several metres of reinforced concrete but could still be fired from conventional towed artillery like 21cm howitzers. So you could do the job with off-the-shelf equipment and didn't need a supergun at all.
- OWASP stands for "Open Source Foundation for Application Security", like ACM their naming has changed a bit since it was initially founded. Another version is "Open Worldwide Application Security Project". Their security top ten, published since 2003, is used in many standards and organisations including MITRE, PCI-DSS, DISA, and the FTC.

Notes (ctd)

- For a good overview of the subprime mortgage crisis and how everyone was so involved in it that no-one wanted to hit the emergency stop, see "Financial Fiasco", Johan Norberg, Cato Institute, 2009. For string theory, see "Not Even Wrong", Peter Woit, Basic Books, 2006.
- The term "stunt cryptography" is from Thomas Ptacek, `https://news.ycombinator.com/item?id=31831049`, via Martin Albrecht and Kenny Paterson, "Analysing Cryptography in the Wild".
- If you thought the title of this talk was too much then you definitely don't want to read physicist Chris Ferrie's book "Quantum Bullshit", in particular chapter 7, "Quantum f**king technomagic", which explains quantum computing.

Notes (ctd)

- Details on the special tricks used to factor 15 and 21, and what the compiled Shor's algorithm is, are in "Pretending to factor large numbers on a quantum computer", John Smolin, Graeme Smith, and Alex Vargo.
- A longer discussion of sleight-of-hand and stunt factorisations is in an upcoming paper.
- The figure for broken PQC algorithms is from Dan Bernstein, "Quantifying risks in cryptographic selection processes". It's an older paper so things have possibly got even worse by now.
- The card deck depicted is called a force deck, used to force subjects to pick a specific card. It's usually encountered in the form of a Svengali deck where the magician can show you a deck apparently containing all different cards but force you to pick from all-identical cards.

Notes (ctd)

- The observation about the D-Wave “factorisation” is from Markku-Juhani O.Saarinen, `https://x.com/mjos_crypto/status/1893989617575092240`
- The Joe Groff quote is from `https://f.duriansoftware.com/@joe/113188727301593689`.
- Scribble is very well trained and virtually never barks so his owner had to play with him with a ball for awhile to get him to bark.
  It was a special performance just for the slides, because he understands the importance of evidence-based science.