

POST- QUANTUM

HOW YOUR PASSWORDS WILL BE
SAFE IN A QUANTUM FUTURE

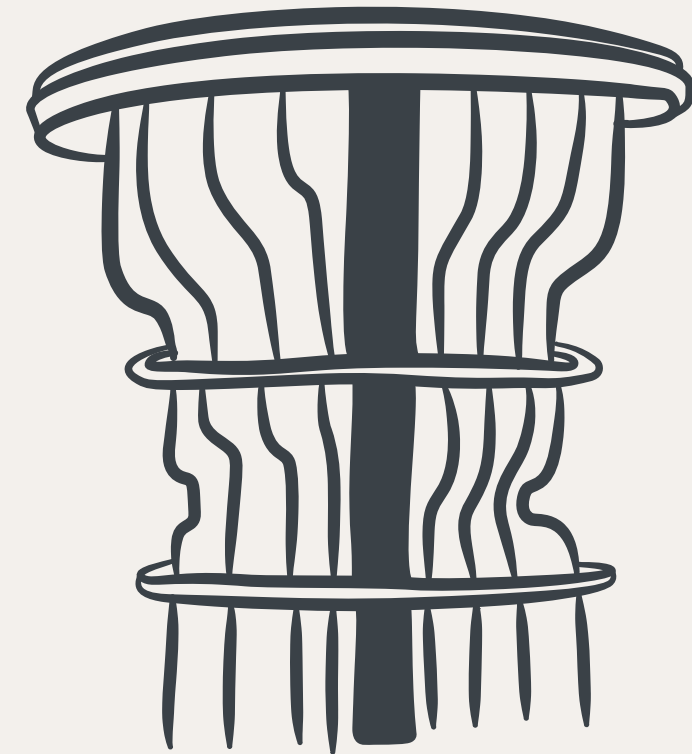
TODAY



“Classical”
Computer



is being
replaced by



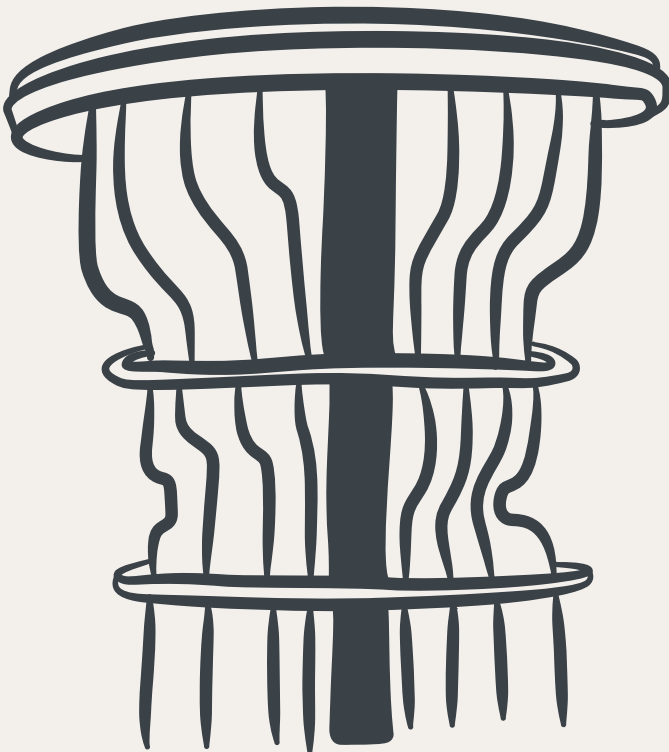
Quantum
Computer



“Classical”
Computer



is being
~~replaced~~
enhanced by



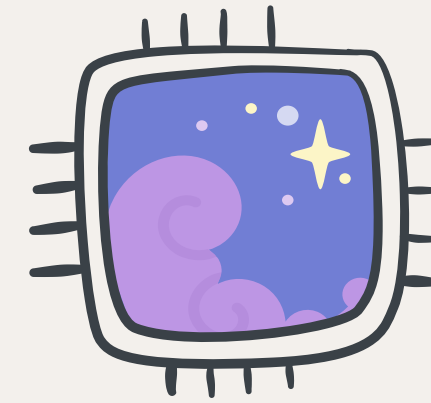
Quantum
Computer



“Classical”
Computer



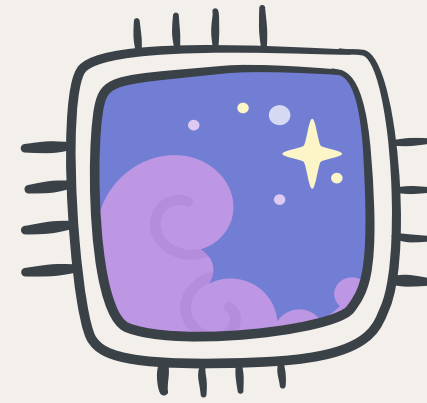
is being
~~replaced~~
enhanced by



Quantum
Processing Unit
(QPU)



“Classical”
Computer



Quantum Processing
Unit (QPU)

“Classical” bit

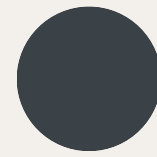


vs.

“Quantum” bit



“Classical” bit



Can be in one of two possible
states, 0 or 1

n “classical” bits



Can be in 1 of 2^n states

“Quantum” bit



can be in two possible states,
0 or 1, simultaneously

“Quantum” bit



can be in two possible states,
0 or 1, simultaneously

$$|0\rangle + |1\rangle$$

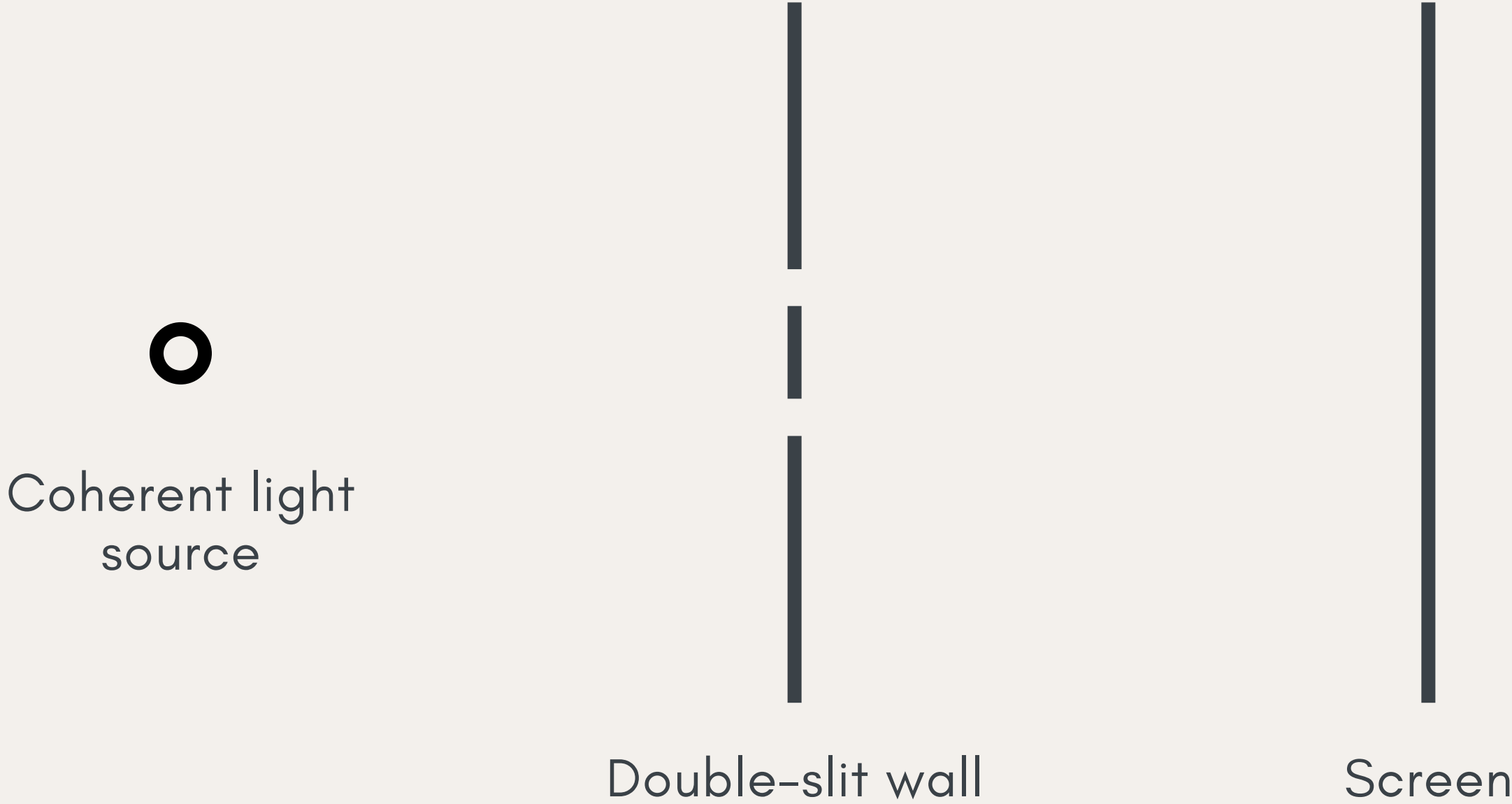
n “quantum” bits



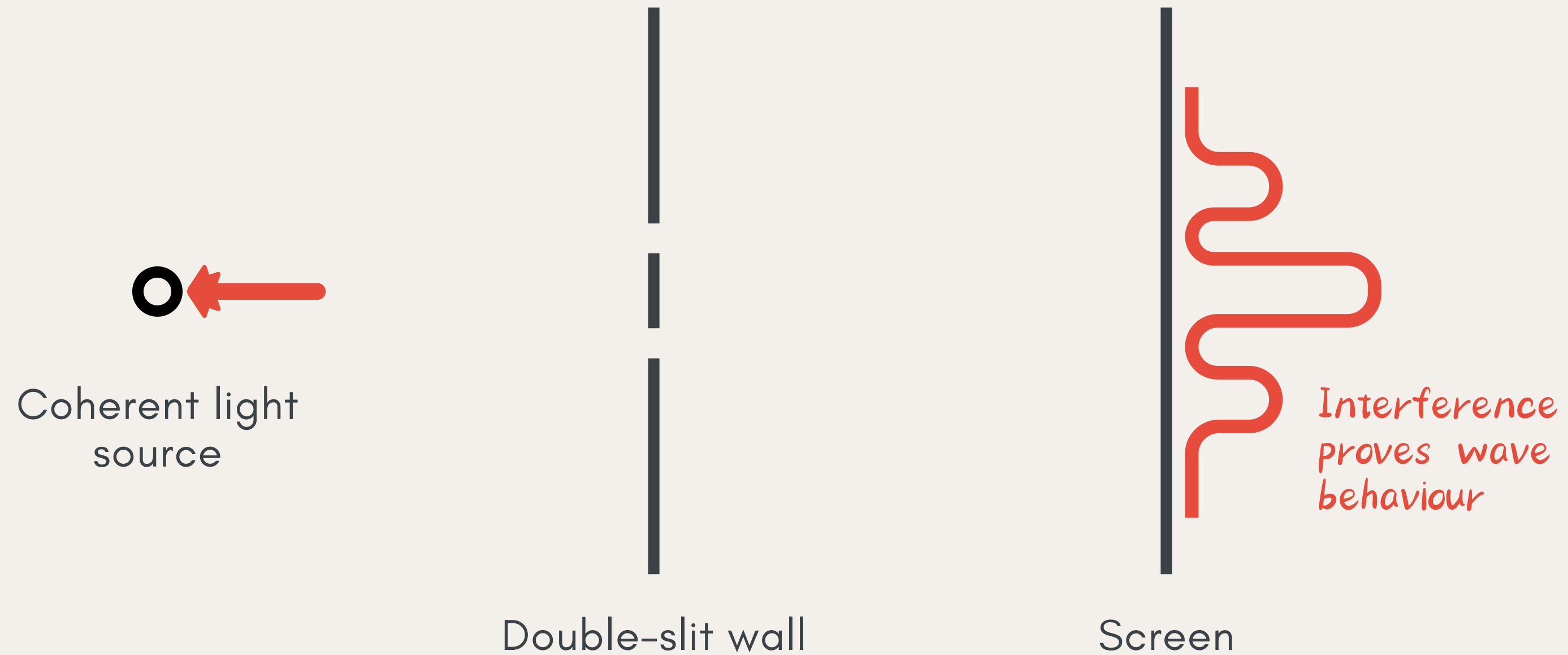
can be in 2^n states
simultaneously

$$|0\rangle + |1\rangle + |2\rangle + \dots + |2^n\rangle$$

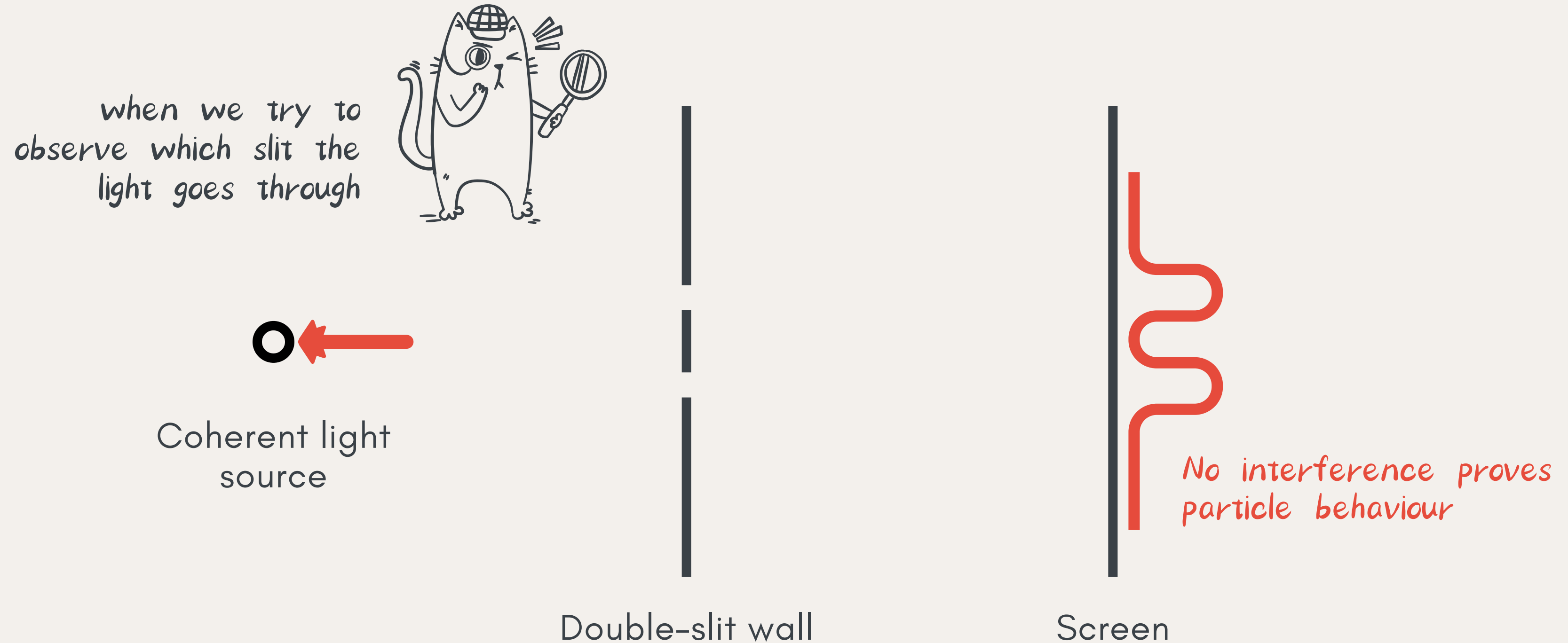
Part I: Quantum computing fundamentals – Superposition & interference



Part I: Quantum computing fundamentals – Superposition & interference

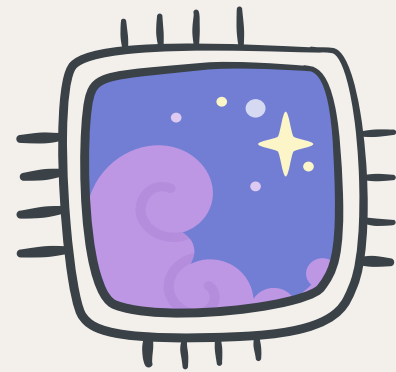


Part I: Quantum computing fundamentals – Superposition & interference



Part I: Quantum computing fundamentals – Qubit superposition & interference

$$x + 2 = 3$$



Quantum
circuit

$$x = |0\rangle + |1\rangle + |2\rangle + |3\rangle + \dots \longrightarrow |0, false\rangle + |1, true\rangle + |2, false\rangle + \dots$$

$$|0, false\rangle + |1, true\rangle + |2, false\rangle + \dots$$

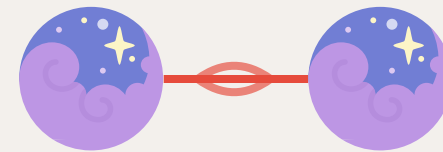
Part I: Quantum computing fundamentals — Qubit superposition & interference



state collapses to a single
value when being measured

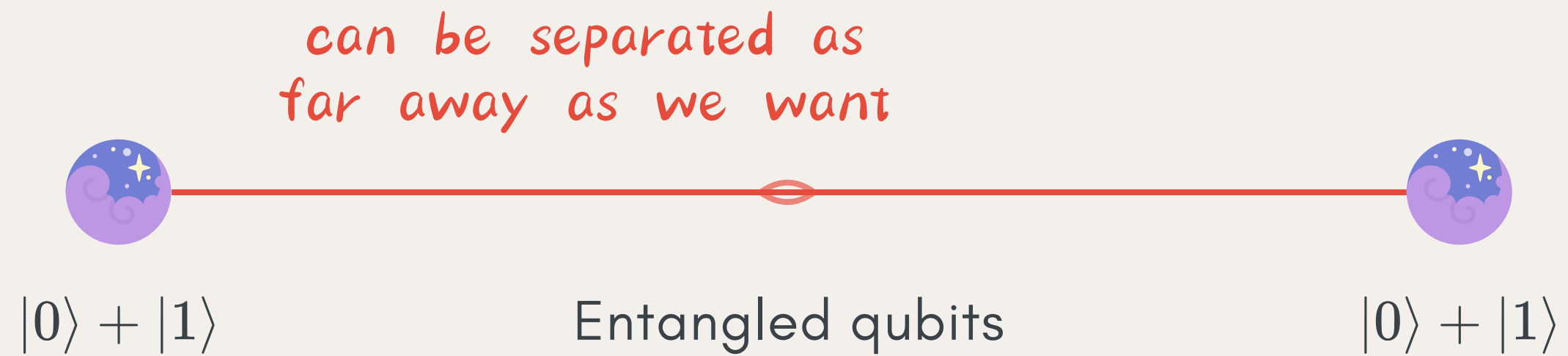
$$\langle 0, true \rangle + \langle 1, false \rangle + |2, false\rangle + \dots$$

Part I: Quantum computing fundamentals — Entanglement



Entangled qubits

Part I: Quantum computing fundamentals – Entanglement





when we observe one, the measurement results of both are correlated



1



0



“Spooky action at
a distance”

— Albert Einstein

Part II: Shor's algorithm breaking RSA encryption

Part II: Shor's algorithm breaking RSA encryption



Modern
RSA encryption



Modern
RSA encryption

$$N = a \times b$$

Really large
number

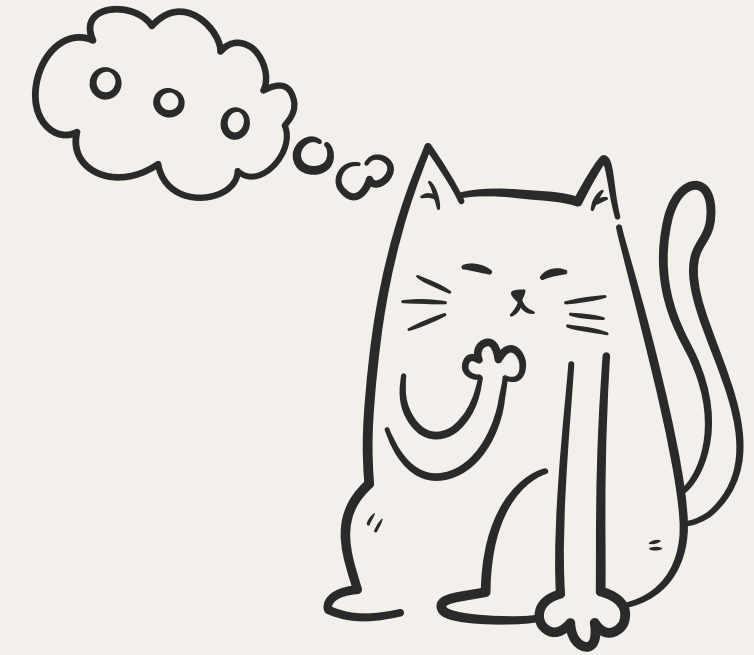
You need its factors to
decrypt the message



Modern
RSA encryption

$$N = a \times b$$

Really large
number



You need its factors to
decrypt the message

even with supercomputer,
guessing by brute force would
take over 300 trillion years



Modern
RSA encryption

$$N = a \times b$$

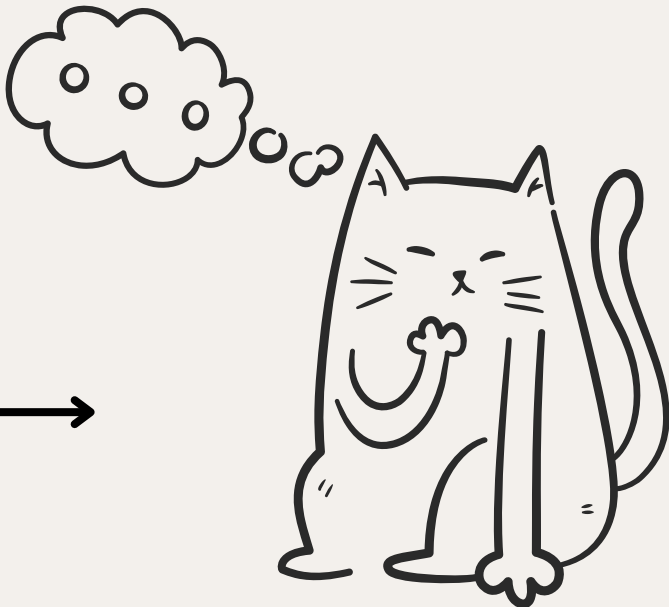
Really large
number

You need its factors to
decrypt the message

Part II: Shor's algorithm breaking RSA encryption



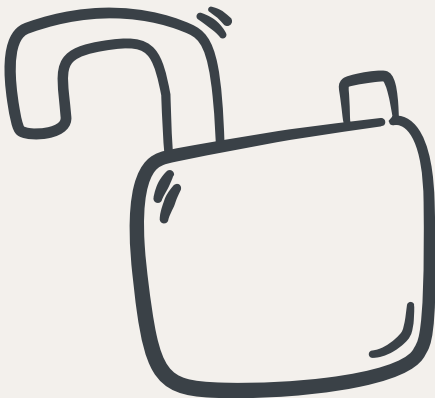
Modern
RSA encryption



Making a
guess



300
trillion
years

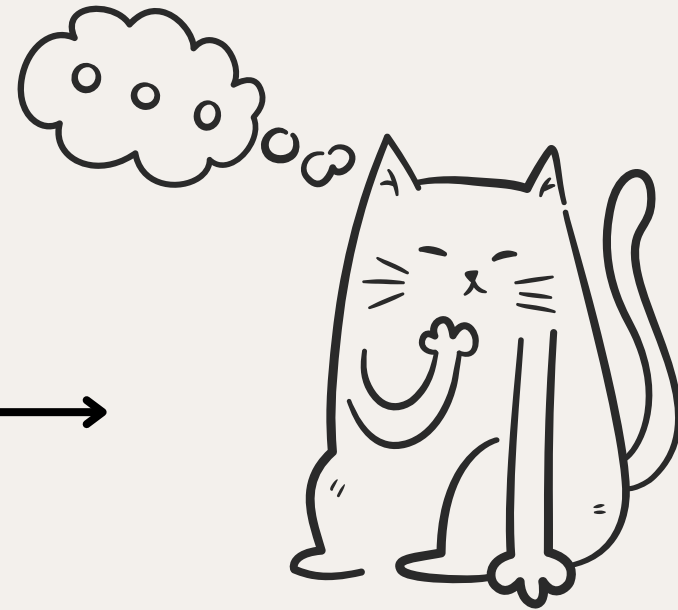


Break
encryption

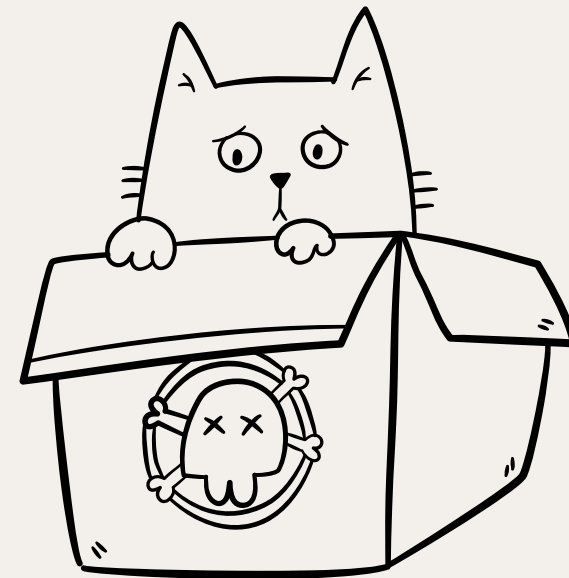
Part II: Shor's algorithm breaking RSA encryption



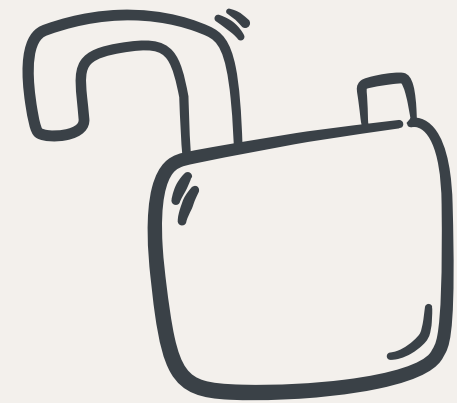
Modern
RSA encryption



Making a
guess



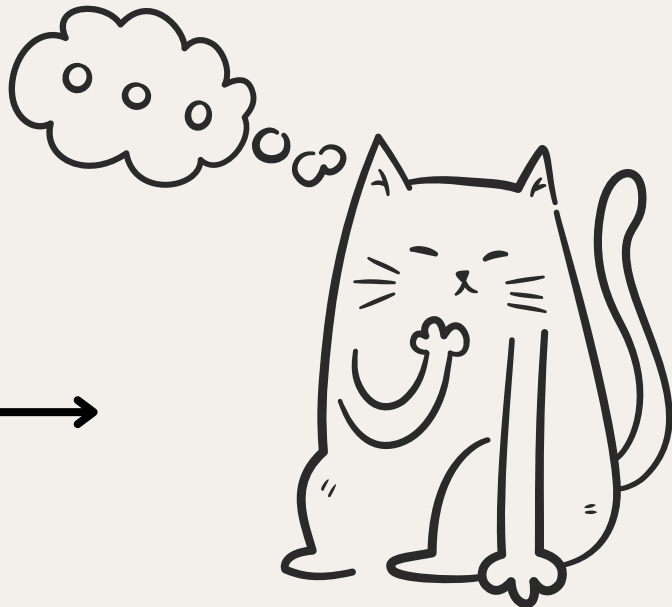
"magic box"
turns bad
guess into
good one



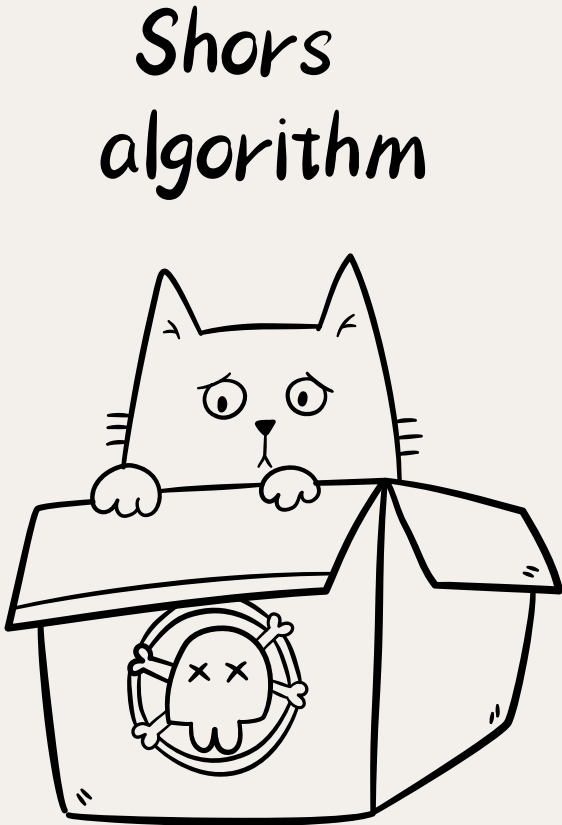
Break
encryption



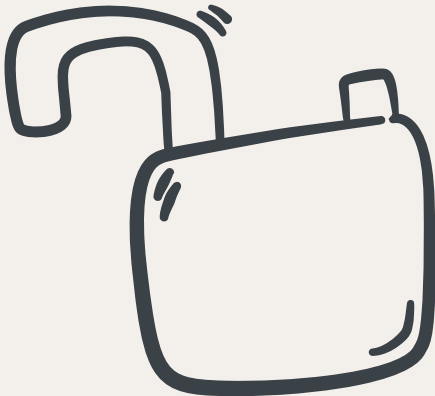
Modern
RSA encryption



Making a
guess



"magic box"
turns bad
guess into
good one



Break
encryption

$$A \times A \times A \times \dots \times A = m \times B + 1$$

$$A^p = m \times B + 1$$

$$g^p = m \times N + 1$$

$$(g^{p/2} + 1) \times (g^{p/2} - 1) = m \times N$$

$$g^p = m \times N + 1$$

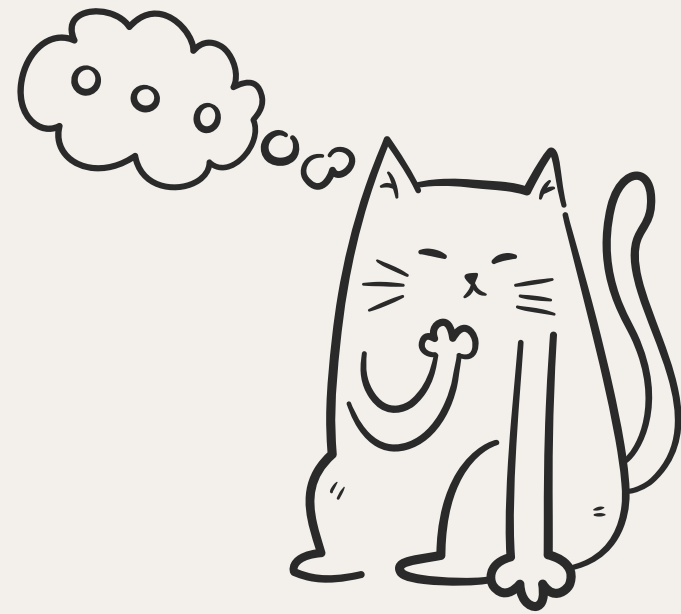
$$(g^{p/2} + 1) \times (g^{p/2} - 1) = m \times N$$

something

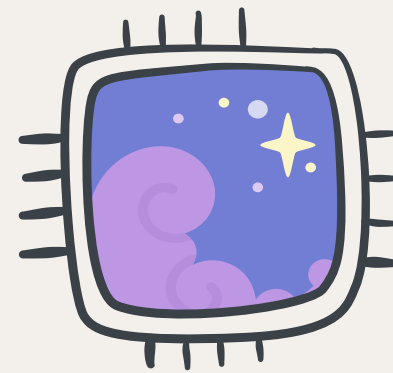
something else

shares factors
with N

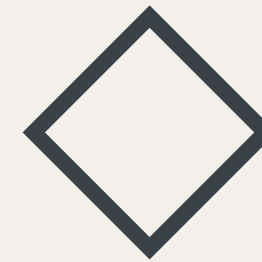
Part II: Shor's algorithm breaking RSA encryption



Making a
crappy guess

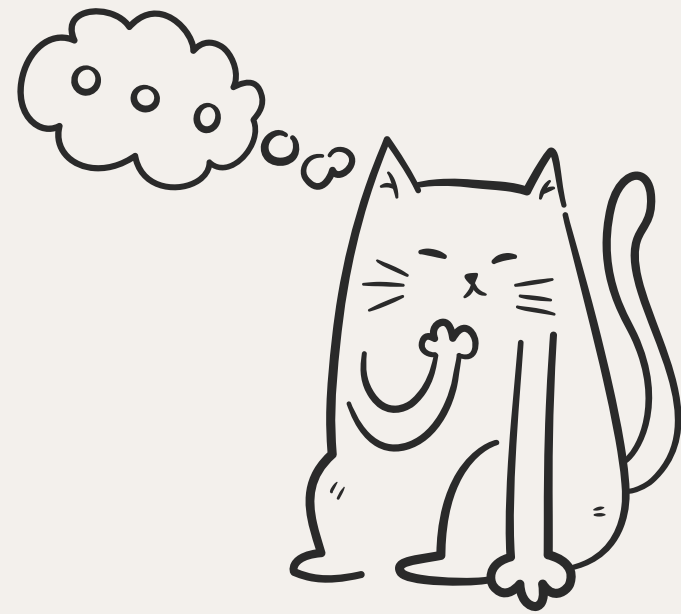


Shor's
algorithm
turns bad
guess into
good one

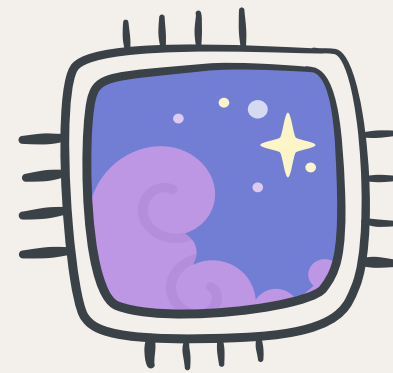


Check for
problems

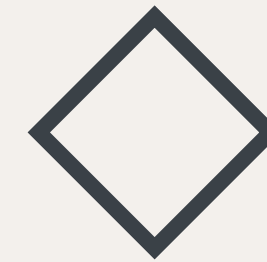
Part II: Shor's algorithm breaking RSA encryption



Making a
crappy guess

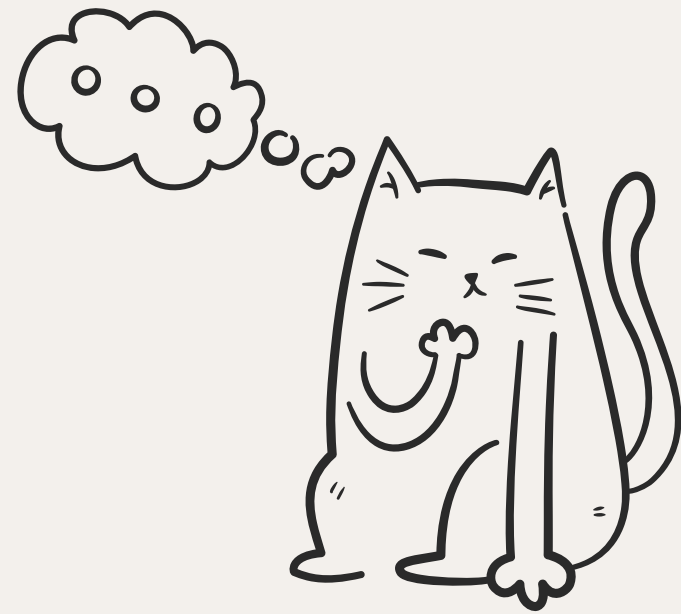


Shor's
algorithm
turns bad
guess into
good one

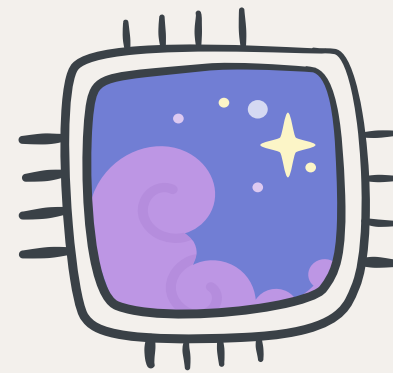


Check for
problems
multiple of N ?

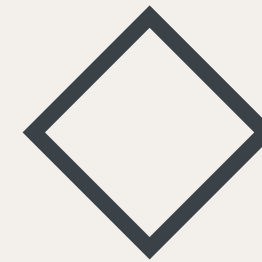
Part II: Shor's algorithm breaking RSA encryption



Making a
crappy guess



Shor's
algorithm
turns bad
guess into
good one

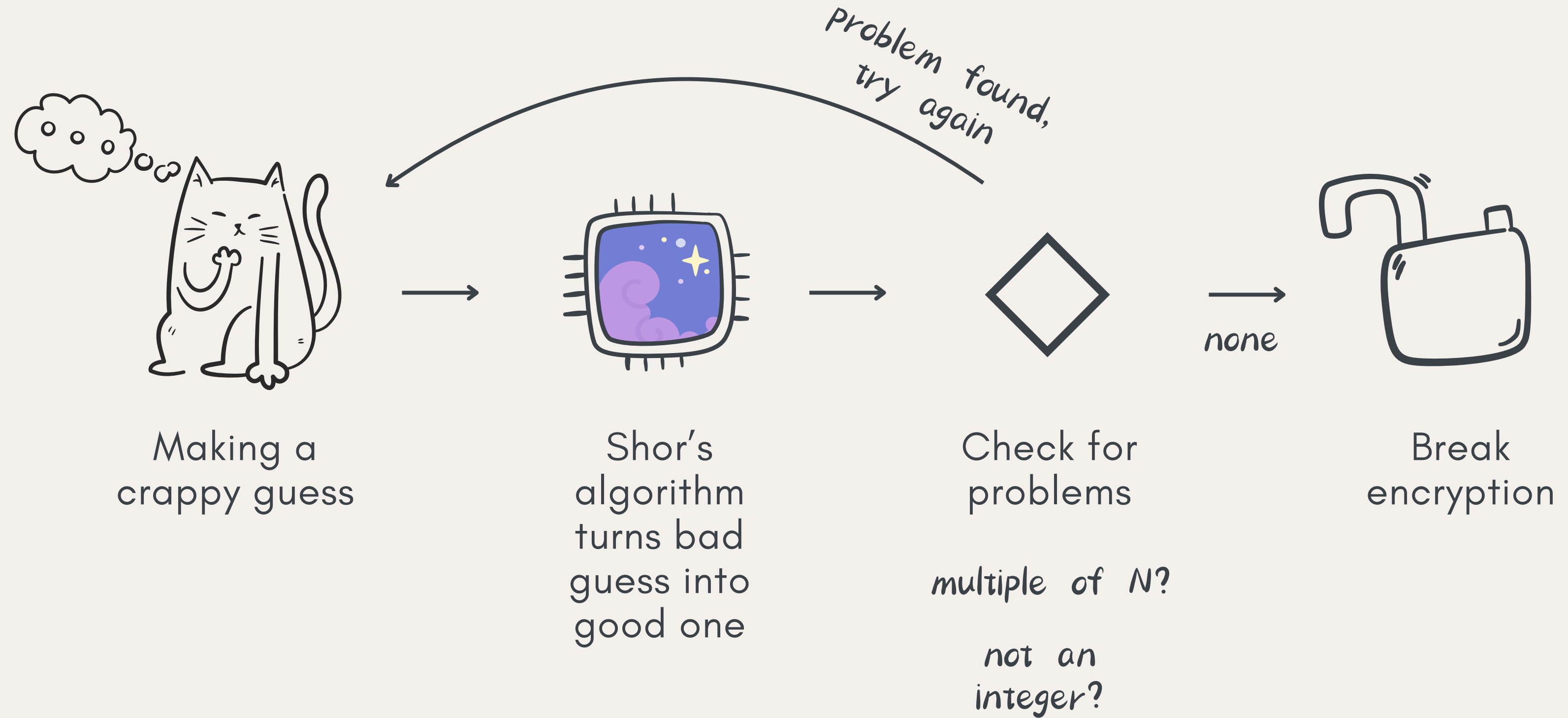


Check for
problems

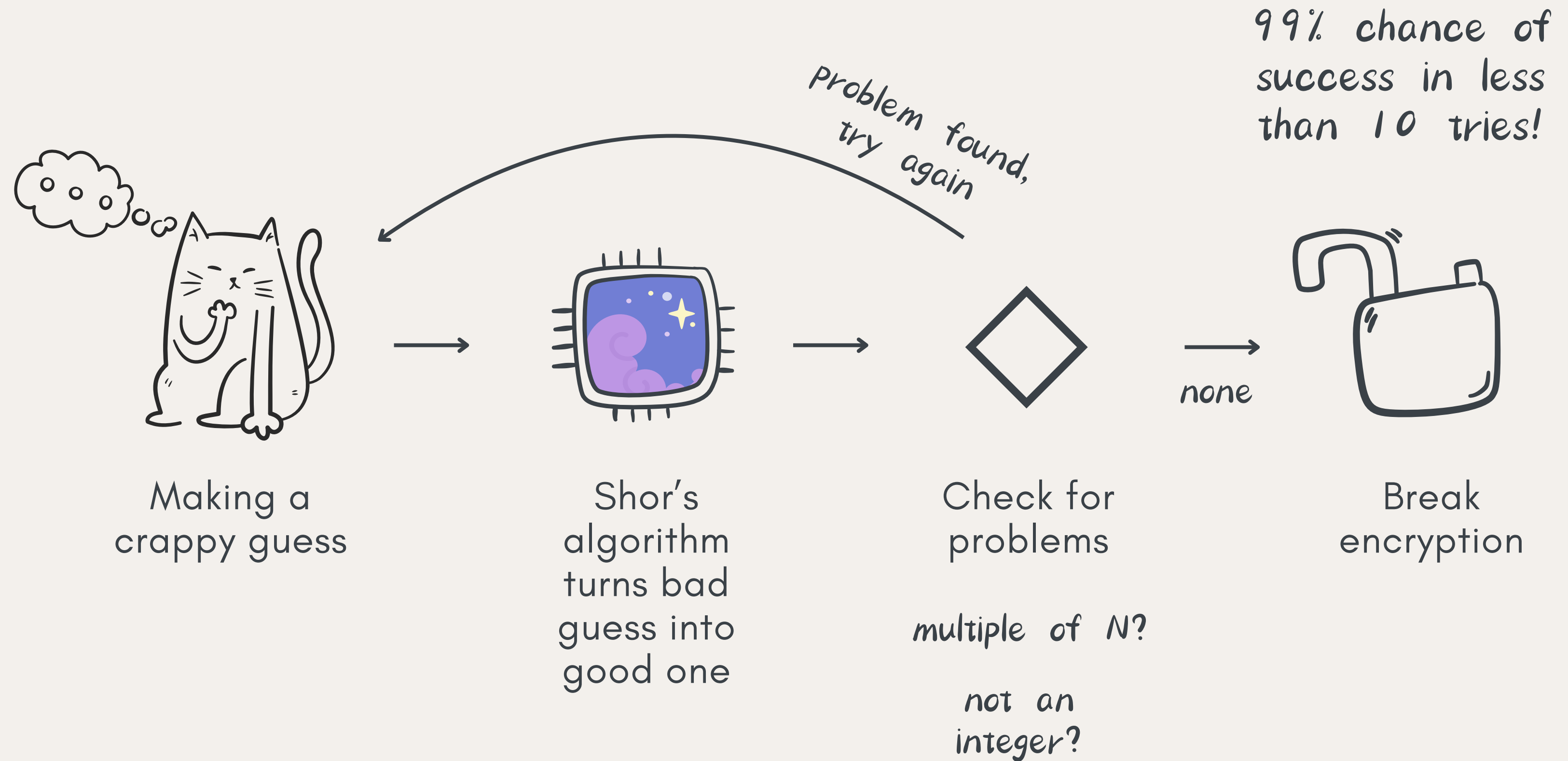
multiple of N ?

*not an
integer?*

Part II: Shor's algorithm breaking RSA encryption



Part II: Shor's algorithm breaking RSA encryption



Part II: Shor's algorithm breaking RSA encryption

[HTTPS://WWW.YOUTUBE.COM/WATCH?V=LVTQBM5DQ4Q](https://www.youtube.com/watch?v=LVTQBM5DQ4Q)

$$g^p = m \times N + 1$$

$$g^p = 1 \pmod{N}$$

$$g^p = 1 \pmod{N}$$

$$g^x = r \pmod{N}$$

e.g. for $x=21$,
we might get a
remainder $r=3$

$$g^p = 1 \pmod{N}$$

$$g^x = r \pmod{N}$$

$$g^{x+p} = r \pmod{N}$$

*e.g. for $x=21$,
we might get a
remainder $r=3$*

$$g^p = 1 \pmod{N}$$

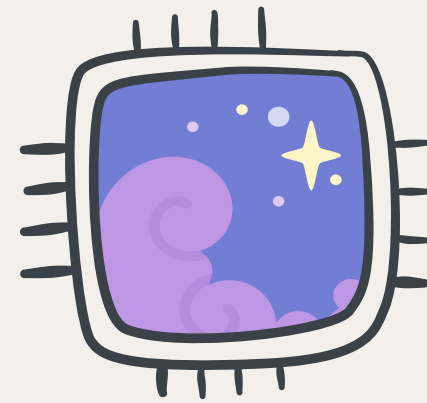
$$g^x = r \pmod{N}$$

$$g^{x+p} = r \pmod{N}$$

$$g^{x+2p} = r \pmod{N}$$

*e.g. for $x=21$,
we might get a
remainder $r=3$*

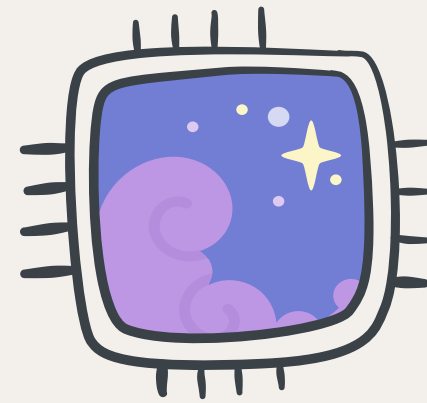
$$g^x = r \pmod{N}$$



Quantum
circuit



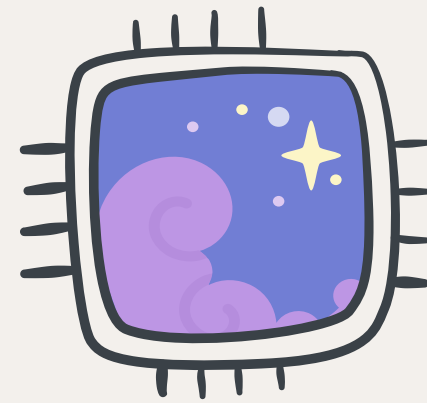
$$g^x = r \pmod{N}$$



Quantum
circuit

$$x = |0\rangle + |1\rangle + |2\rangle + \dots \longrightarrow$$

$$g^x = r \pmod{N}$$



Quantum
circuit

$$x = |0\rangle + |1\rangle + |2\rangle + \dots \longrightarrow |0, +17\rangle + |1, +3\rangle + |2, +92\rangle + \dots$$



$$|1, +3\rangle + |11, +3\rangle + |21, +3\rangle + \dots$$



*only measure the
remainder value*

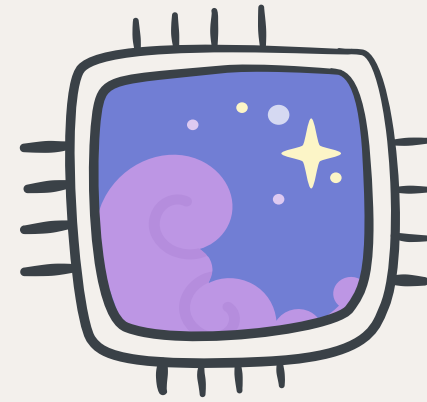
$$|1, +3\rangle + |11, +3\rangle + |21, +3\rangle + \dots$$



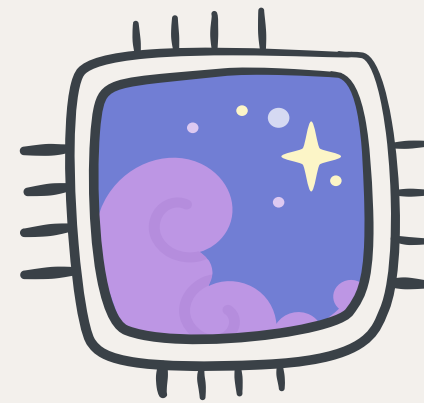
only measure the
remainder value

$$|1\rangle + |11\rangle + |21\rangle + |31\rangle + |41\rangle + \dots$$

the resulting superposition will
have a frequency of 1 over p



Quantum Fourier
Transform

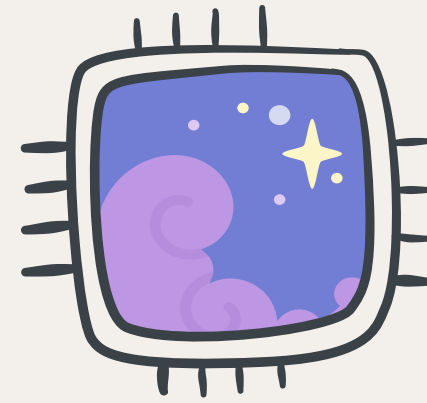


Quantum Fourier
Transform

$$|1\rangle + |11\rangle + |21\rangle + |31\rangle + \dots \longrightarrow \frac{1}{10} \quad \text{the result lets us infer the frequency of the superposition}$$



"Classical"
Computer



QPU: modulo
calculation + QFT

Part II: Shor’s algorithm breaking RSA encryption – Current challenges with QPUs

Problem 1: Number of qubits

E.g. to run Shor's algorithm on RSA-2048, you would need around 20 million “noisy” qubits to have sufficient memory

[HTTPS://WWW.QUANTUM-EXPECTATIONS.COM/ANSWERS/MOST-QUBITS-QUANTUM-COMPUTER](https://www.quantum-expectations.com/answers/most-qubits-quantum-computer)

[HTTPS://EN.WIKIPEDIA.ORG/WIKI/LIST_OF_QUANTUM_PROCESSORS](https://en.wikipedia.org/wiki/List_of_quantum_processors)

Problem 1: Number of qubits

E.g. to run Shor's algorithm on RSA-2048, you would need around ~~25 million~~ "noisy" qubits to have sufficient memory
1 million

[HTTPS://WWW.QUANTUM-EXPECTATIONS.COM/ANSWERS/MOST-QUBITS-QUANTUM-COMPUTER](https://www.quantum-expectations.com/answers/most-qubits-quantum-computer)

[HTTPS://EN.WIKIPEDIA.ORG/WIKI/LIST_OF_QUANTUM_PROCESSORS](https://en.wikipedia.org/wiki/List_of_quantum_processors)

Problem 1: Number of qubits

E.g. to run Shor's algorithm on RSA-2048, you would need around ~~20 million~~ "noisy" qubits to have sufficient memory
~~100k~~ 100k

[HTTPS://WWW.QUANTUM-EXPECTATIONS.COM/ANSWERS/MOST-QUBITS-QUANTUM-COMPUTER](https://www.quantum-expectations.com/answers/most-qubits-quantum-computer)

[HTTPS://EN.WIKIPEDIA.ORG/WIKI/LIST_OF_QUANTUM_PROCESSORS](https://en.wikipedia.org/wiki/List_of_quantum_processors)

Problem 1: Number of qubits

E.g. to run Shor's algorithm on RSA-2048, you would need around ~~20 million~~ "noisy" qubits to have sufficient memory
~~100k~~ 100k

Problem 2: Error correction

It is really hard to prevent any unintentional outside influence that would cause the quantum state irreversibly to collapse

Proactive solution

We shouldn't wait until the problem is “real” before solving it, proactively working on a solution is good

Proactive solution

We shouldn't wait until the problem is “real” before solving it, proactively working on a solution is good

Prevent “store now, decrypt later”

SNDL attacks are one of the biggest threats to RSA at the moment, where attackers collect encrypted data now to decrypt once RSA has been broken



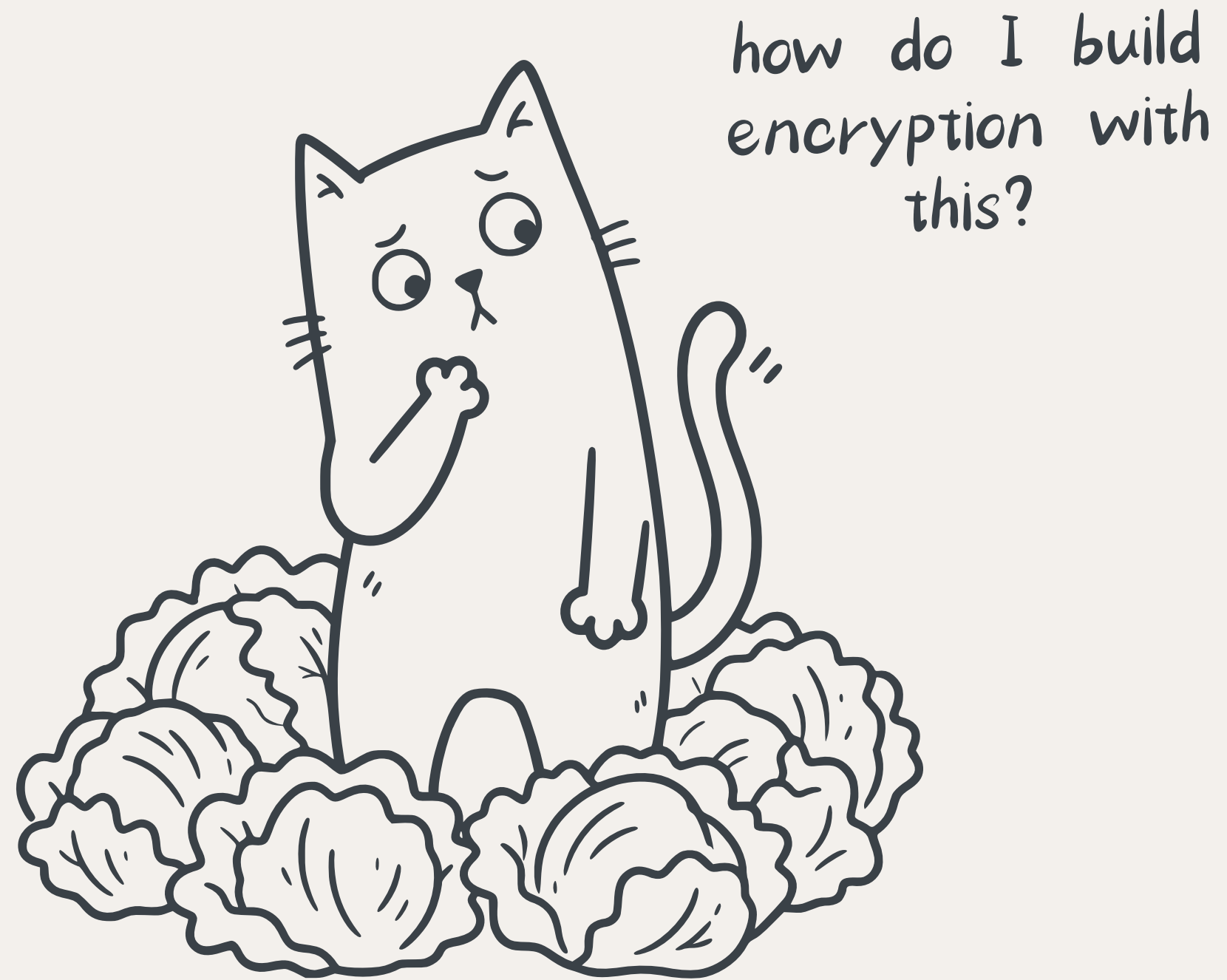
NIST – National Institute for Standards and Technology

NIST – National Institute for Standards and Technology

- Lattice based
- Hash based
- Code based
- Multivariate based

NIST – National Institute for Standards and Technology

- Lattice based ML-KEM (Kyber), ML-DSA (Dilithium), FN-DSA (Falcon)
- Hash based SLH-DSA (SPHINCS+)
- Code based HQC
- Multivariate based MAYO



Part III: Post quantum cryptography — Lattice based cryptography

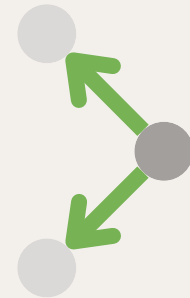


[HTTPS://WWW.YOUTUBE.COM/WATCH?V=QDDOOYDB748](https://www.youtube.com/watch?v=QDDOOYDB748)

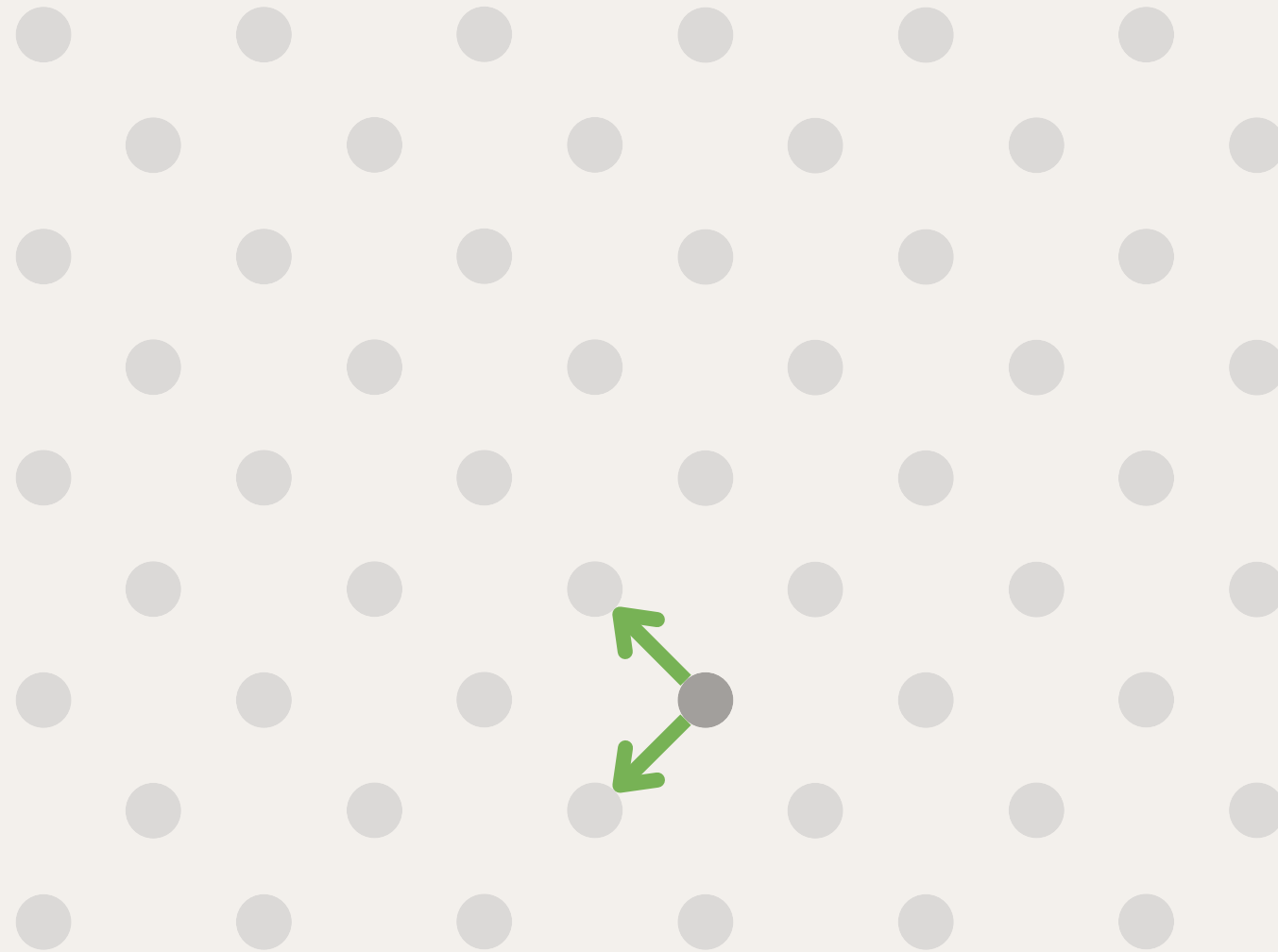
Part III: Post quantum cryptography — Lattice based cryptography

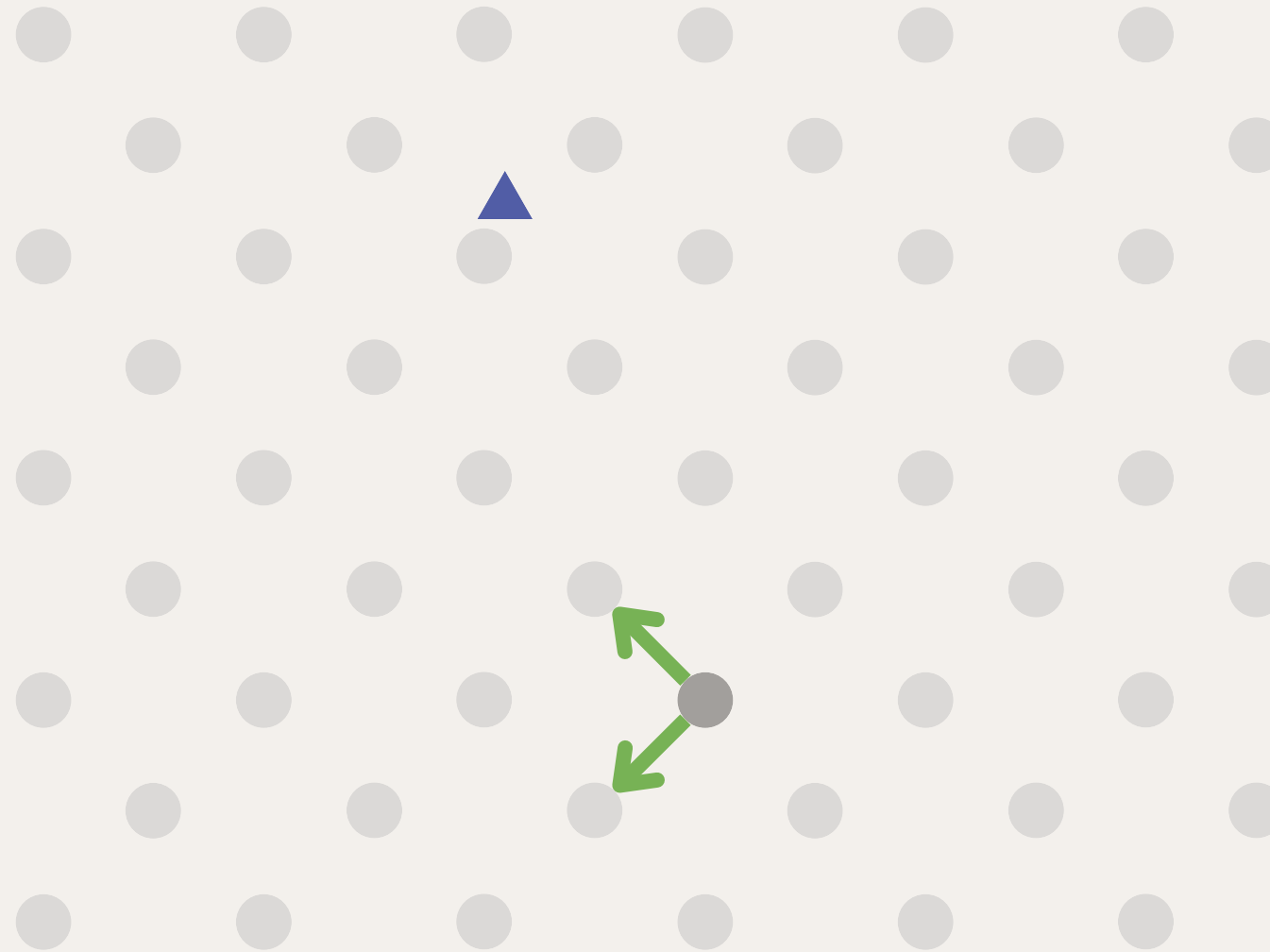


Part III: Post quantum cryptography — Lattice based cryptography

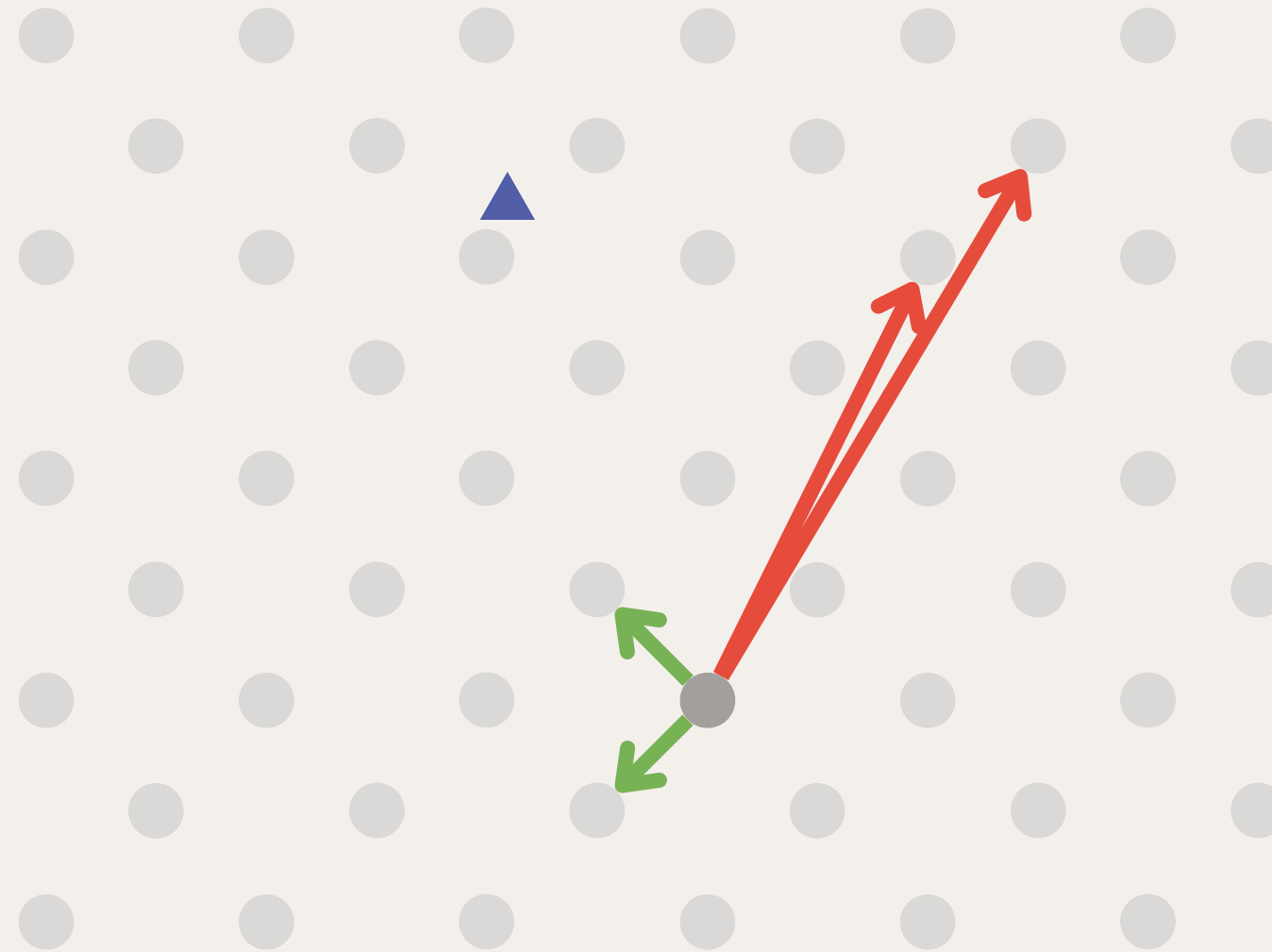


Part III: Post quantum cryptography – Lattice based cryptography

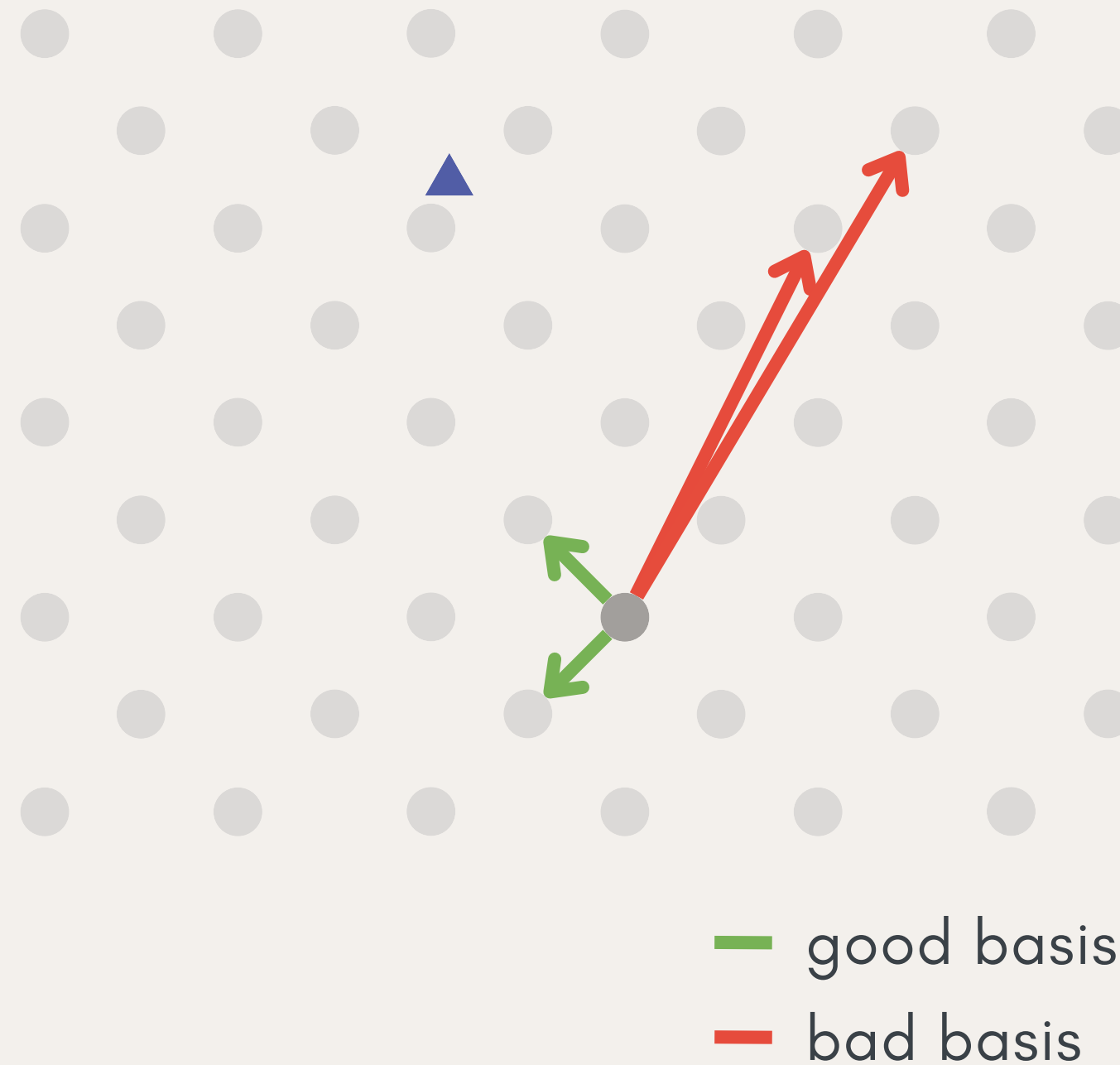




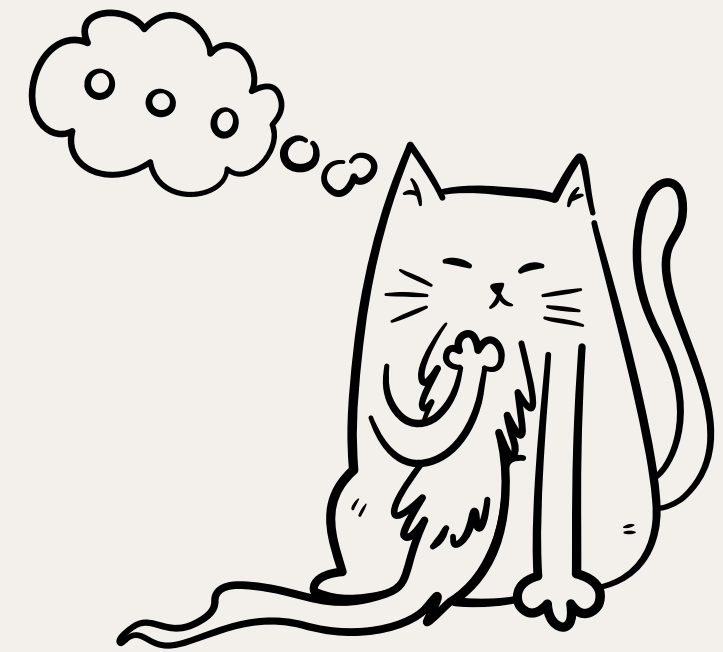
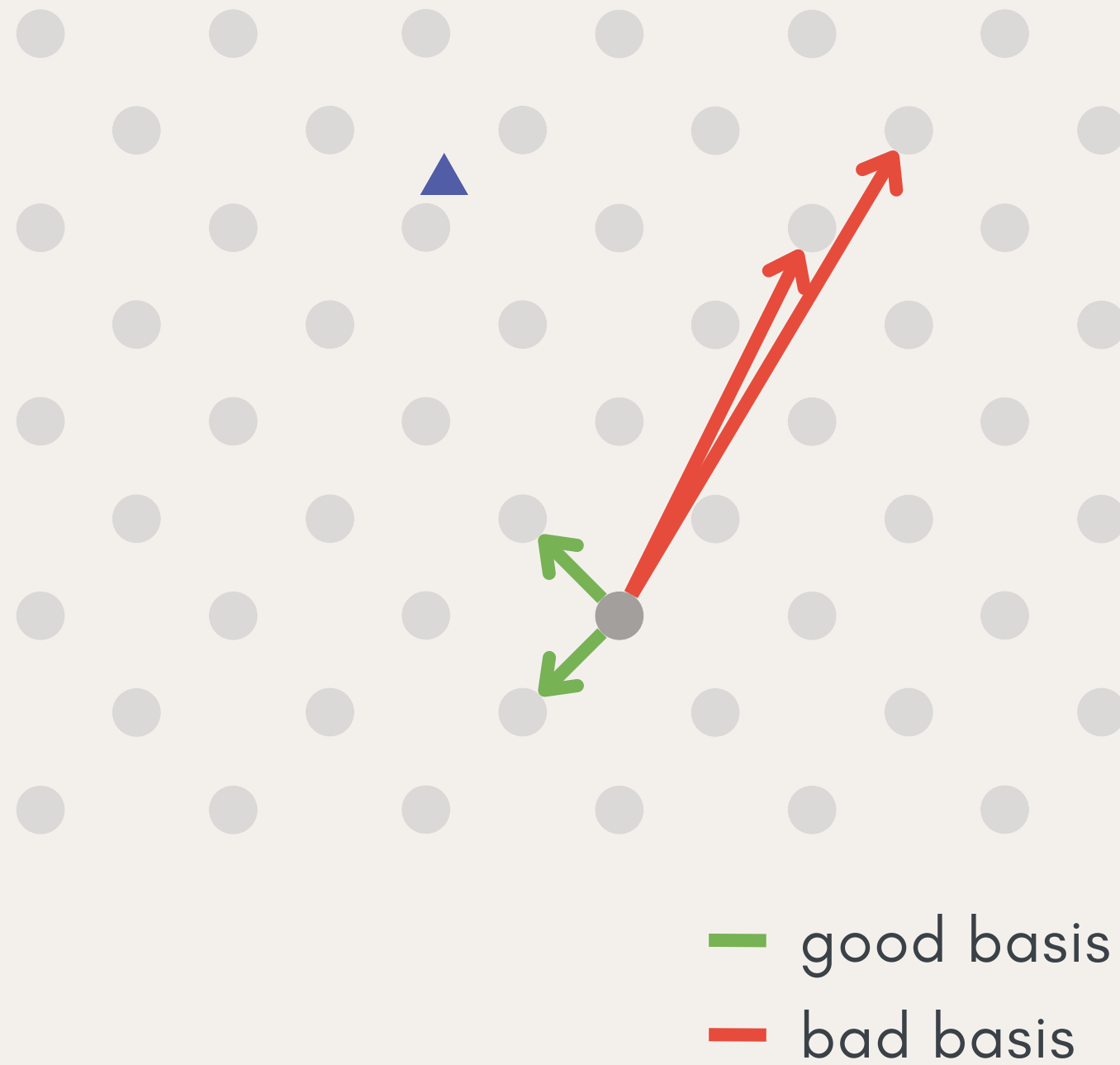
what is the closest
point on the lattice to
the target?



what is the closest
point on the lattice to
the target?



what is the closest
point on the lattice to
the target?



finding the closest point
with vectors of the bad
basis is really hard

Learning with errors

Let's introduce deliberate linear errors to our equations we share as our public key, to make it significantly harder to solve them programmatically

Private key

10

82

50

5

Private key

10

82

50

5

Public key

$$77x + 7y + 28z + 23w = 2859$$

$$21x + 19y + 30z + 48w = 3508$$

$$4x + 24y + 33z + 38w = 3848$$

$$8x + 20y + 84z + 61w = 6225$$

$$6x + 53y + 1z + 86w = 4886$$

$$42x + 86y + 31z + 8w = 9062$$

Private key

10

82

50

5

Public key

$$77x + 7y + 28z + 23w = 2859$$

$$21x + 19y + 30z + 48w = 3508$$

$$4x + 24y + 33z + 38w = 3848$$

$$8x + 20y + 84z + 61w = 6225$$

$$6x + 53y + 1z + 86w = 4886$$

$$42x + 86y + 31z + 8w = 9062$$

Public key

$$\begin{aligned}77x + 7y + 28z + 23w &= 2859 \\21x + 19y + 30z + 48w &= 3508 \\4x + 24y + 33z + 38w &= 3848 \\8x + 20y + 84z + 61w &= 6225 \\6x + 53y + 1z + 86w &= 4886 \\42x + 86y + 31z + 8w &= 9062\end{aligned}$$

Public key

$$77x + 7y + 28z + 23w = 11 \pmod{89}$$

$$21x + 19y + 30z + 48w = 37 \pmod{89}$$

$$4x + 24y + 33z + 38w = 21 \pmod{89}$$

$$8x + 20y + 84z + 61w = 84 \pmod{89}$$

$$6x + 53y + 1z + 86w = 80 \pmod{89}$$

$$42x + 86y + 31z + 8w = 73 \pmod{89}$$

Public key

$$\begin{aligned}77x + 7y + 28z + 23w &= 11 & + & -3 \\21x + 19y + 30z + 48w &= 37 & + & 2 \\4x + 24y + 33z + 38w &= 21 & + & -1 \\8x + 20y + 84z + 61w &= 84 & + & 0 \\6x + 53y + 1z + 86w &= 80 & + & 4 \\42x + 86y + 31z + 8w &= 73 & + & -1\end{aligned}$$

Public key

$$\begin{aligned}77x + 7y + 28z + 23w &= 8 \\21x + 19y + 30z + 48w &= 39 \\4x + 24y + 33z + 38w &= 20 \\8x + 20y + 84z + 61w &= 84 \\6x + 53y + 1z + 86w &= 84 \\42x + 86y + 31z + 8w &= 72\end{aligned}$$

Public key

$$21x + 19y + 30z + 48w = 39$$

$$4x + 24y + 33z + 38w = 20$$

$$5x + 24y + 79z + 27w = 49$$

Public key

$$21x + 19y + 30z + 48w = 39$$

$$4x + 24y + 33z + 38w = 20$$

$$5x + 24y + 79z + 27w = 49$$

$$30x + 67y + 53z + 24w = 19$$

Public key

$$21x + 19y + 30z + 48w = 39$$

$$4x + 24y + 33z + 38w = 20$$

$$5x + 24y + 79z + 27w = 49$$

$$30x + 67y + 53z + 24w = 19$$

$$30x + 67y + 53z + 24w = 19 + 44$$

$$30x + 67y + 53z + 24w = 19 + 0$$

the modulo allows us to
construct encrypted
“1” and “0”

Decrypting the message

$$30x + 67y + 53z + 24w = 63 \pmod{89}$$

$$30x + 67y + 53z + 24w = 19 \pmod{89}$$

Decrypting the message

$$30x + 67y + 53z + 24w = 63 \pmod{89}$$

$$30x + 67y + 53z + 24w = 19 \pmod{89}$$

$$30 \times 10 + 67 \times 82 + 53 \times 50 + 24 \times 5 = 20 \pmod{89}$$

knowing the real answer, we can
calculate the error, which should be
either close to 0 or to 44

Decrypting the message

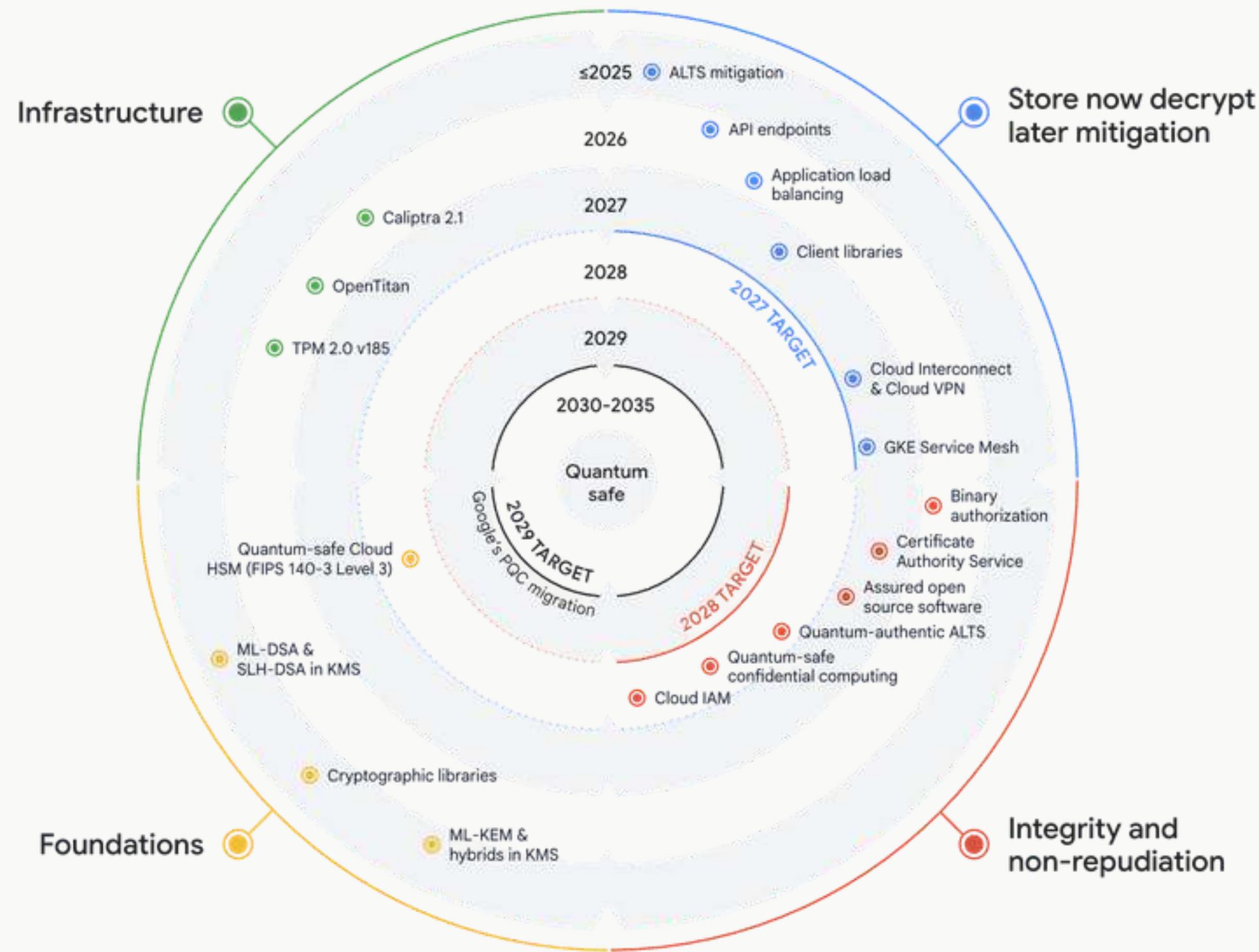
$$30x + 67y + 53z + 24w = 63 \pmod{89} + 43$$

$$30x + 67y + 53z + 24w = 19 \pmod{89} + -1$$

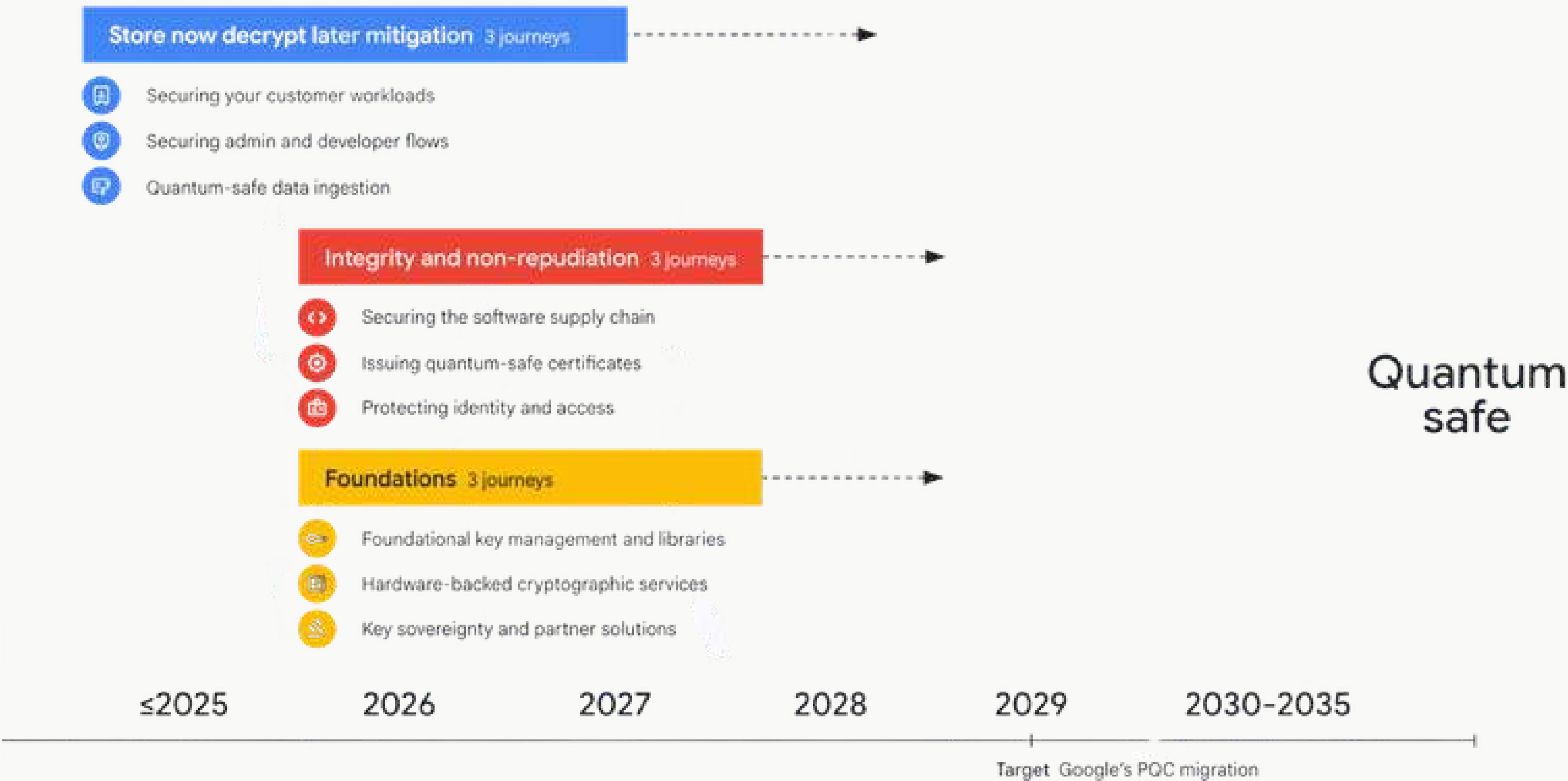
$$30 \times 10 + 67 \times 82 + 53 \times 50 + 24 \times 5 = 20 \pmod{89}$$

knowing the real answer, we can
calculate the error, which should be
either close to 0 or to 44

Part III: Post quantum cryptography – Timelines to a quantum-safe future



Part III: Post quantum cryptography – Timelines to a quantum-safe future



Conclusion

How Quantum Computers Break Encryption | Shor's Algorithm Explained

<https://www.youtube.com/watch?v=lvTqbM5Dq4Q>

Post Quantum Cryptography - Computerphile

https://www.youtube.com/watch?v=_MoRcYLN-7U

Lattice-based cryptography: The tricky math of dots

<https://www.youtube.com/watch?v=QDdOoYdb748>

Learning with errors: Encrypting with unsolvable equations

<https://www.youtube.com/watch?v=K026C5YaB3A>

The Idea behind Lattice-Based Cryptography

<https://nicklaskoertge.medium.com/the-idea-behind-lattice-based-cryptography-5e623fa2532b>

Lattice-based cryptography explained

<https://qramm.org/learn/lattice-based-cryptography.html>

PQC in Plaintext: Google Cloud's post-quantum cryptography roadmap

Google — <https://tinyurl.com/2948dn54>

Post-Quantum Cryptography

AWS — <https://aws.amazon.com/security/post-quantum-cryptography/>

AWS History and Timeline regarding AWS Key Management Service

https://hidekazu-konishi.com/entry/aws_history_and_timeline_aws_kms.html

Accelerating the quantum-safe timeline

Microsoft / Azure — <https://tinyurl.com/5n8secap>

Thank you!



LINK TO THE SLIDES:

[https://www.julianburr.de/
devoxx-belgium-2026-slides.pdf](https://www.julianburr.de/devoxx-belgium-2026-slides.pdf)

<https://www.linkedin.com/in/julianburr/>

<https://bsky.app/profile/julianburr.de>