RSAConference2019
San Francisco | March 4 – 8 | Moscone Center

BETTER.

SESSION ID: BAC-W03

Demystifying Quantum Computers

Radia Perlman
Fellow
Dell EMC

Charlie Kaufman
Security Architect
Dell EMC

Outline of talk

- What is a quantum computer?
- Why do we care about quantum computers?
- An intuition behind one quantum algorithm
- An intuition behind one quantum-safe public key algorithm
- Issues with building quantum computers

What's a quantum computer?

How quantum computers are predicted to work…

- Based on quantum mechanics
- Quantum mechanics is weird
- Unfortunately, quantum mechanicms seems to be true, and we need to live in this universe

RSAConference2019

First: what’s a classical computer?

A classical computer

- Stores information in “bits”
- Each bit stores 0 or 1
- n bits can be in one of 2^n states
- Various logic gates on bits, such as AND, OR, NOT, XOR
  - Take input(s) and yield output(s)

In contrast, a quantum computer…

A quantum computer

- Stores information in “qubits”
- Three weird things about qubits (more on next few slides)
  - Superposition
  - Measurement
  - Entanglement

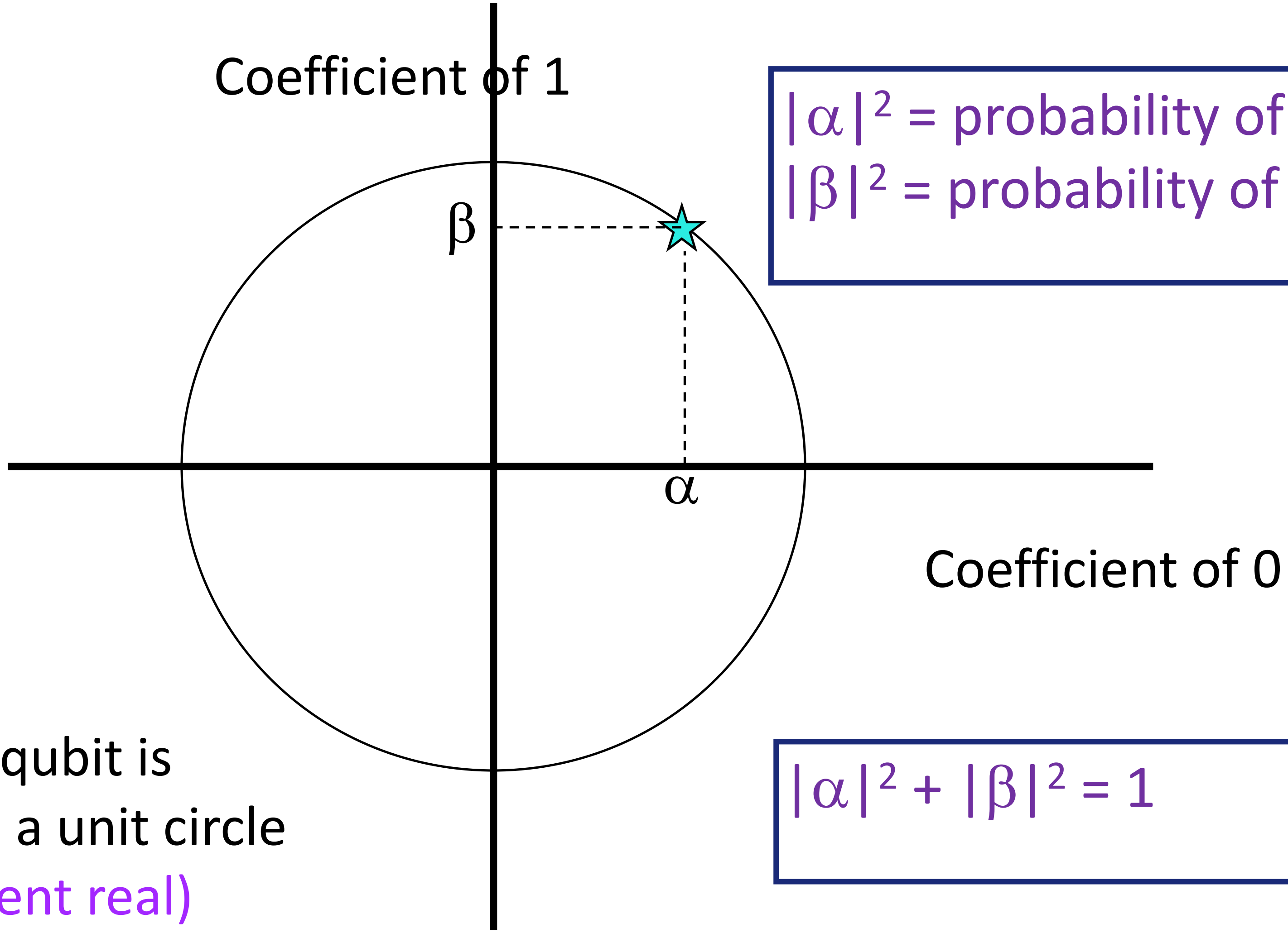
Superposition

- qubit can simultaneously store some amount of 0 and 1
- State of a qubit traditionally notated as
  $\alpha|0\rangle + \beta|1\rangle$
  - "Coefficients" α and β express the amount of 0 vs 1
  - Meaning, the likelihood, if you read it, whether you'll get 0 or 1
- Probability, if read the qubit, that it will read as 0 is $|\alpha|^2$
  - $|\alpha|^2 + |\beta|^2 = 1$

“Bra-ket” notation

$$\alpha|0\rangle + \beta|1\rangle$$

- I don’t like it
- Invented in 1939 by Paul Dirac
- But everyone uses it
- I’ll try to make it a bit more readable using color
- And if it doesn’t make it more readable, at least it will be prettier



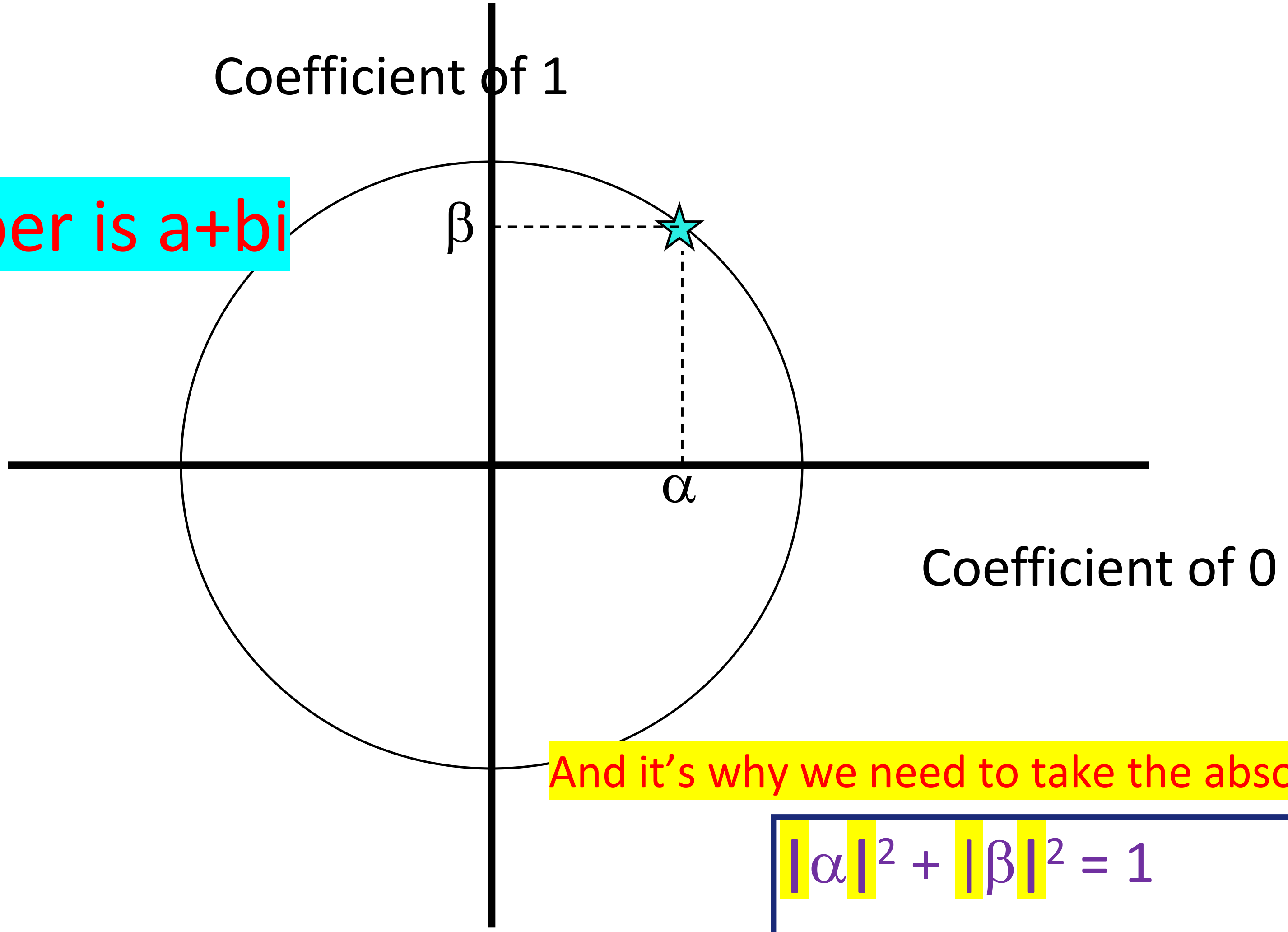
$|\alpha|^2$ = probability of reading 0
$|\beta|^2$ = probability of reading 1

$|\alpha|^2 + |\beta|^2 = 1$

State of a qubit is
a point on a unit circle
(if coefficient real)

Though coefficients are actually complex numbers, so it's a point on a 4-dimensional sphere

Complex number is a+bi

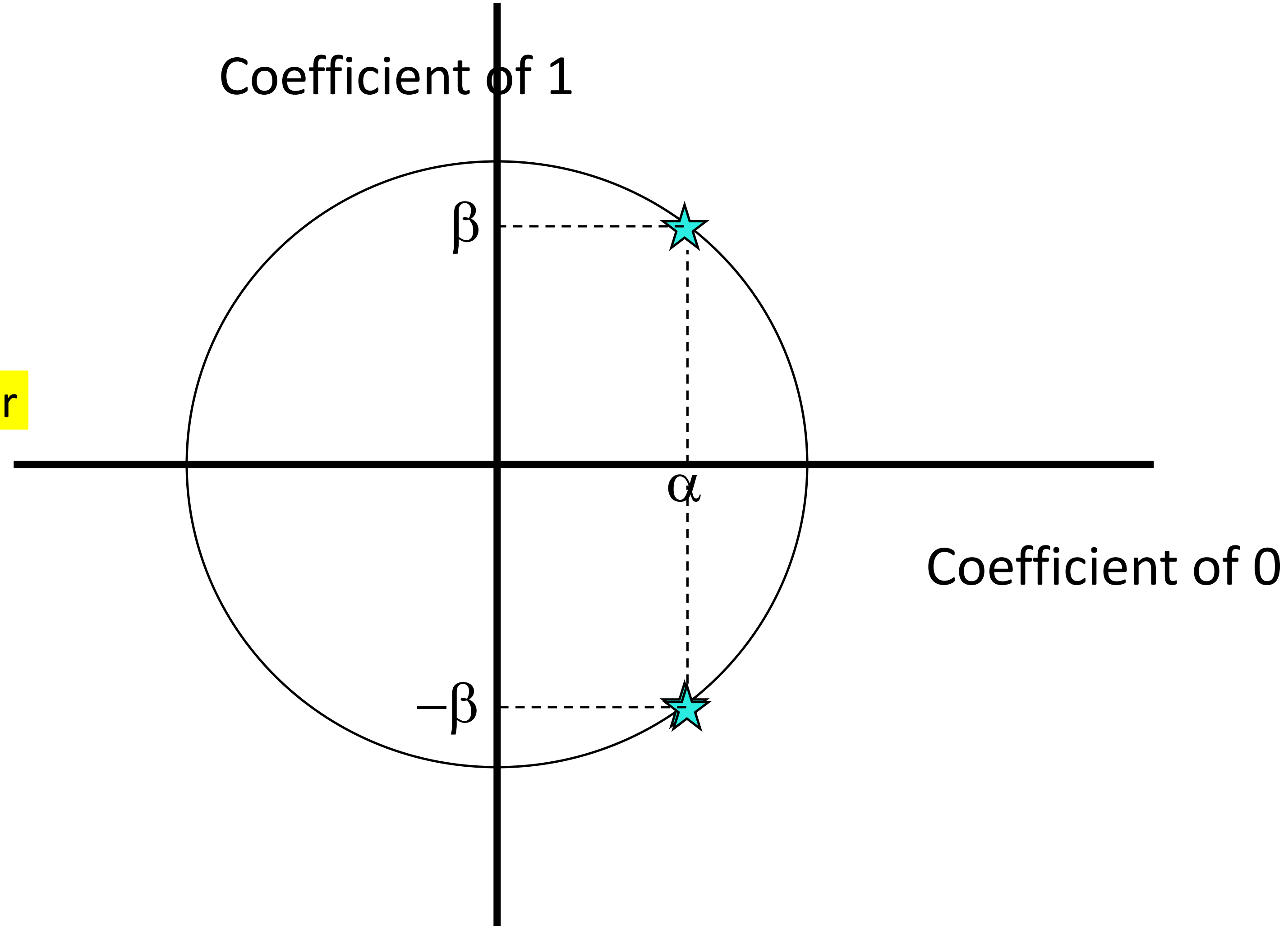


And it's why we need to take the absolute value

$$|\alpha|^2 + |\beta|^2 = 1$$

Phase: Absolute value is not the entire state

Phase doesn’t change measurement probability, but it is part of the state for computation



Two Things to do with qubits

- Measurement
  - Reads the value
  - But after measuring it, the qubit is solidly what you read (no more superposition)
- Gate
  - Does some sort of operation on a set of qubits
  - The output is a superposition proportional to each of the superposed inputs
  - The output overwrites the input qubits, so once the operation is completed the input values are no longer available

Measurement

- If you read the value of a qubit in state α|0> + β|1>, you will get either a 0 or a 1
- And then the qubit has lost its superposition
- It will either be solidly 0

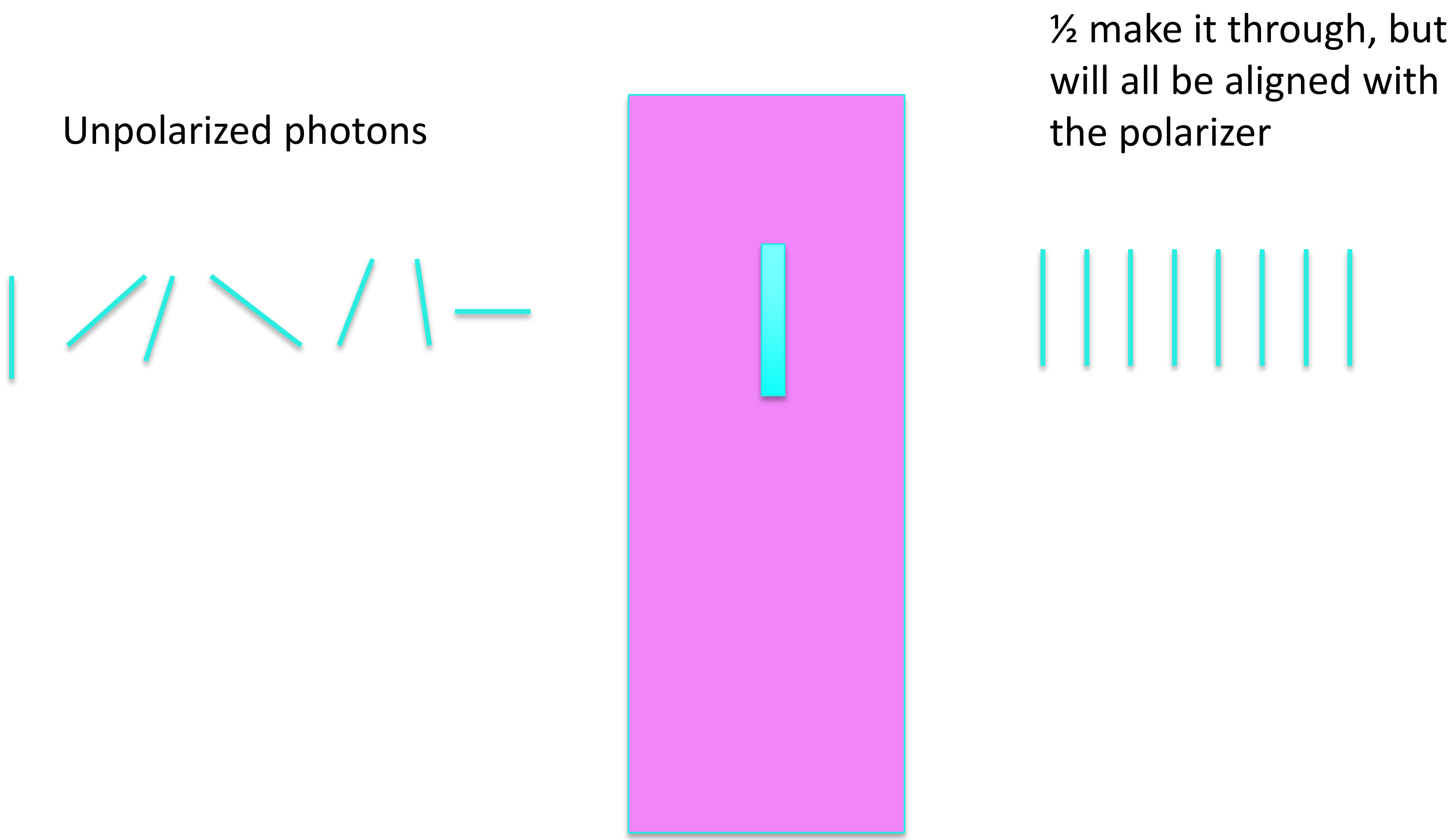
  1|0> + 0|1>
- Or solidly 1

  0|0> + 1|1>

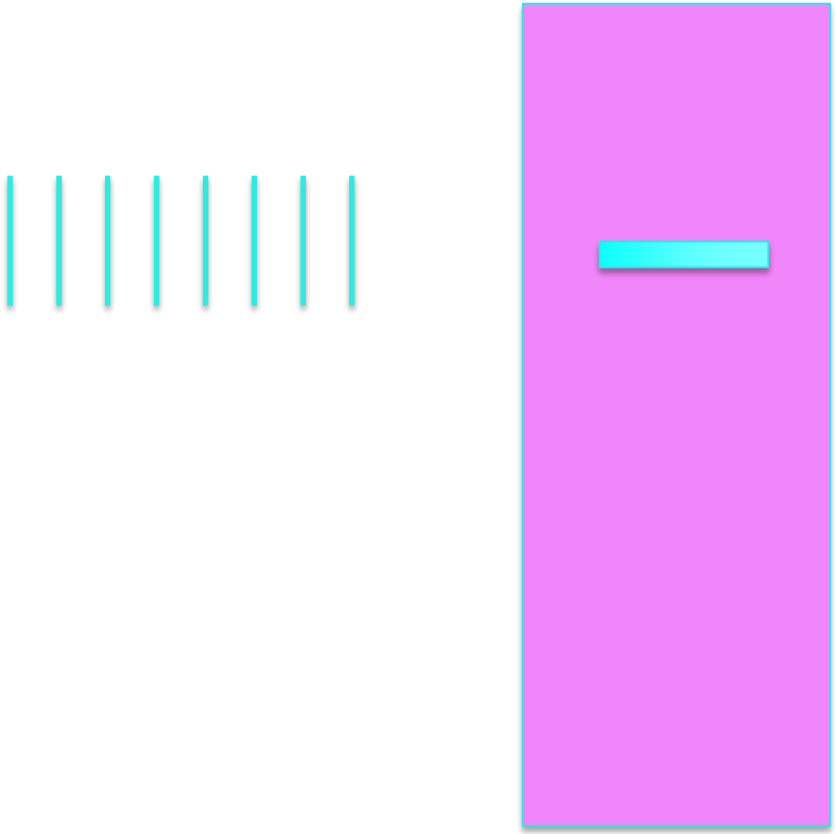
Example of Measurement Changing the Qubit

- Photons can be polarized vertically, horizontally, or anything in between
- If a vertically polarized photon hits a polarizer exactly aligned with it, it will go through
- If it hits a polarizer 90 degrees off, the photon will not go through
- If it's 45 degrees off, probability of ½
- The probability is based on the angle
- If the photon gets through, it's now "twisted" to align with the polarizer

Photons/Polarizers



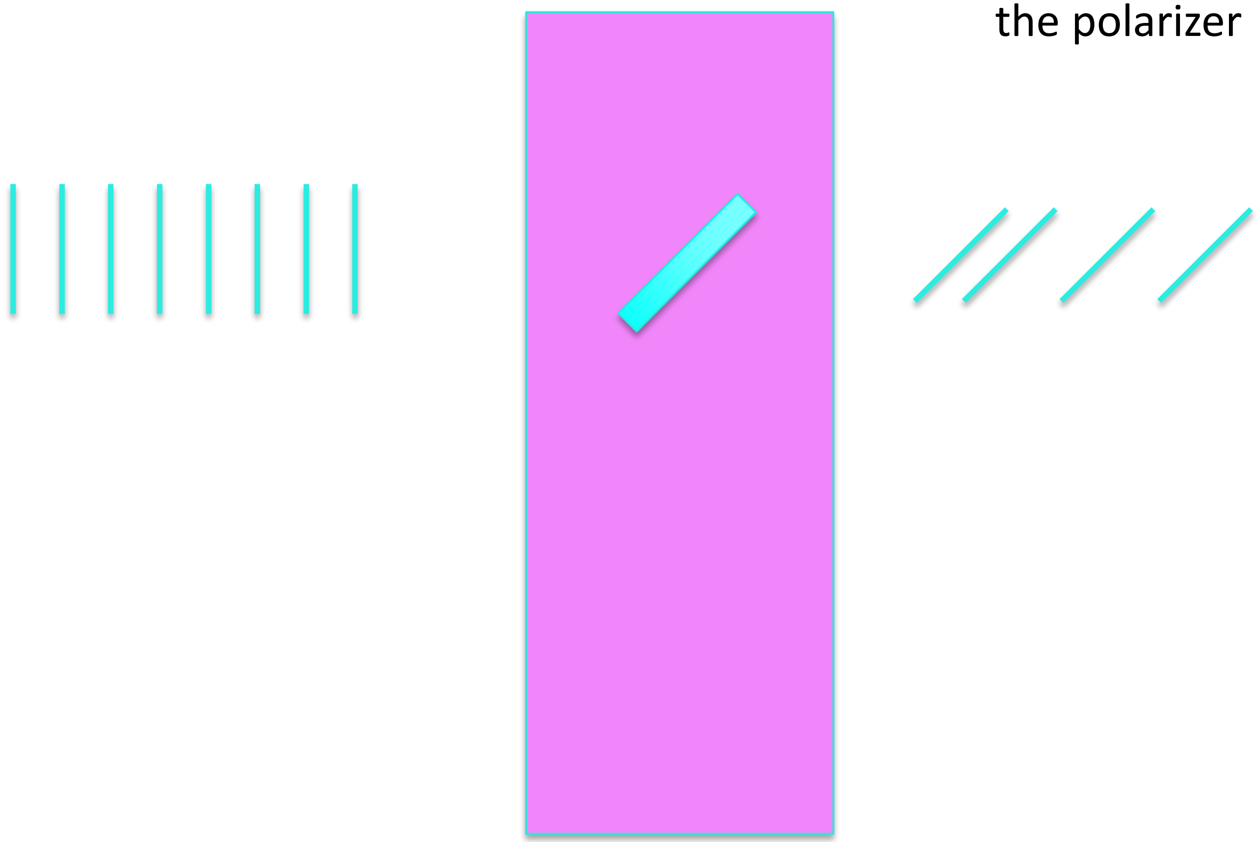
Photons/Polarizers



90 degree off…no photons make it through

Photons/Polarizers

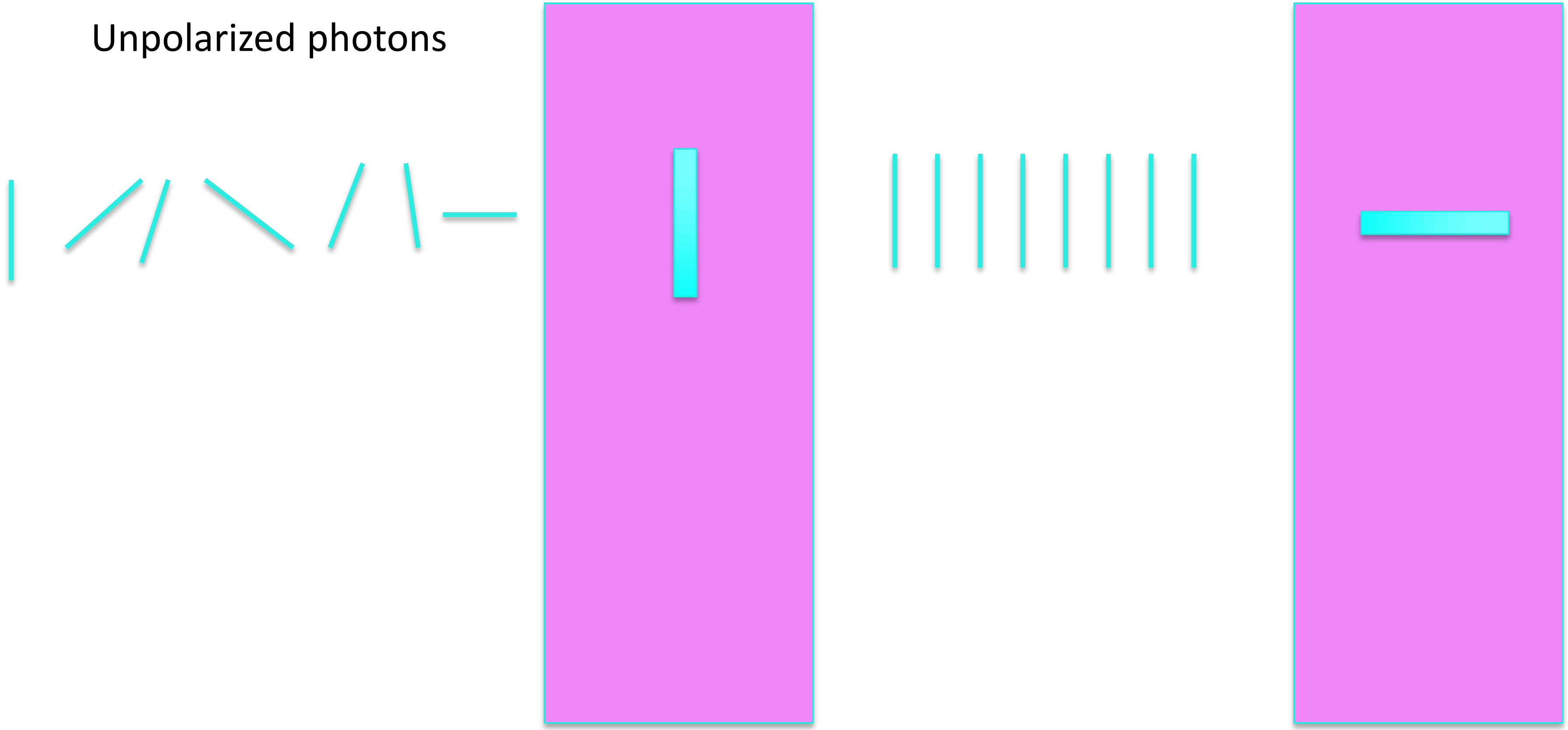
45 degree off…1/2 make it through, but now aligned with the polarizer



Experiment you can do

- Take two polarizing filters, and align them 90 degrees off
- No light goes through

2 Polarizers, 90 degrees off: no photos make it through

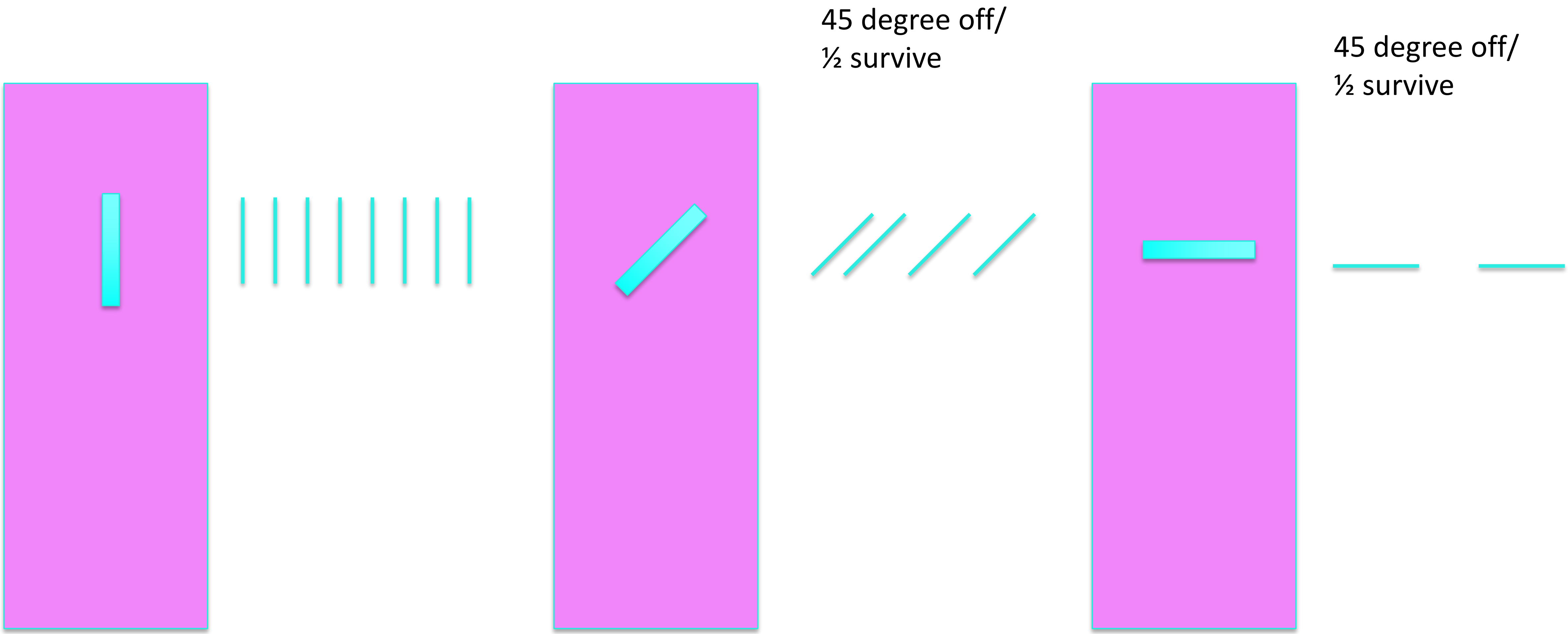


90 degree off…no photons make it through

Now add an extra polarizer

- Insert another polarizer in between, at 45 degrees
- Light goes through!

Put an extra polarizer in between: 1/8 make it through



Entanglement

Entanglement is truly weird!

- A set of entangled qubits no longer have independent states
- For instance, 3 qubits can hold a superposition of any subset of {000, 001, … 111}

Entanglement is deeply disturbing

- Let's say that there are 2 qubits which hold a superposition of 00 and 11, i.e., their state is

$$\alpha|00> + \beta|11>$$

- If you read the first one, you know the value of the 2nd one
- Even if, after entangling them, you move them far apart
- There are two ways this could possibly happen
  - Faster than light communication
  - Previous collusion between the two qubits
- Both are wrong!

RSA Conference 2019

How quantum computers matter to cryptography

Why we care

- Two known cryptography-relevant algorithms that can run on a quantum computer
  - Grover (1996) – square root speedup of classical brute force search (instead of 2^n, it's $2^{n/2}$)
  - Shor (1994) – polynomial time factoring and discrete logs

Known quantum algorithms

- Grover isn't really a problem. In theory it speeds up brute force search for hashes (e.g., SHA) and secret key cryptography (e.g., AES), from 2^n to $2^{n/2}$
  - All we need to do is double the size of the key (or hash)
  - But the algorithm is way cool!
- Shor's algorithm is devastating to current public key algorithms (RSA, ECC, Diffie-Hellman)
  - Really annoying, but the algorithm is also way cool!

Should we care?

- Need to replace current public key algorithms at least 10 years or so before a quantum computer may exist
  - (Sufficiently large quantum computers) may never exist, but to be safe, we have to assume they might
  - It will take years to convert
  - We want data encrypted with current algorithms to remain secret for years

Replacement public key algorithms

- Usually called “Post-quantum”
- I prefer the term “Quantum-safe”
- These run on classical computers

A quantum computer is NOT

- A simple extension of Moore’s Law
  - Quantum computers are not always faster than classical computers…they are different
- A non-deterministic Turing machine

What’s a nondeterministic Turing machine?

- Only a concept!
- Can be simulated (in exponential time) on a regular computer
- (if it existed) it could compute on all branches of a program simultaneously, and output the correct answer when one of the paths through the program finds the correct answer
- A quantum computer sounds similar…<u>but it’s not</u>

Worst Case Scenario

- Some well funded evil entity throws a trillion dollars at the problem, in secret, and creates a quantum computer that can break 2048-bit RSA, and surprises the world
- Suddenly everything on the Internet can be impersonated
- All cryptocurrency could be spent by the bad guys
- So…good guys must try at least as hard as bad guys to build them
- And there can be valuable spinoff technologies, as with the moon mission
- For instance, super-sensitive sensors

An example quantum gate

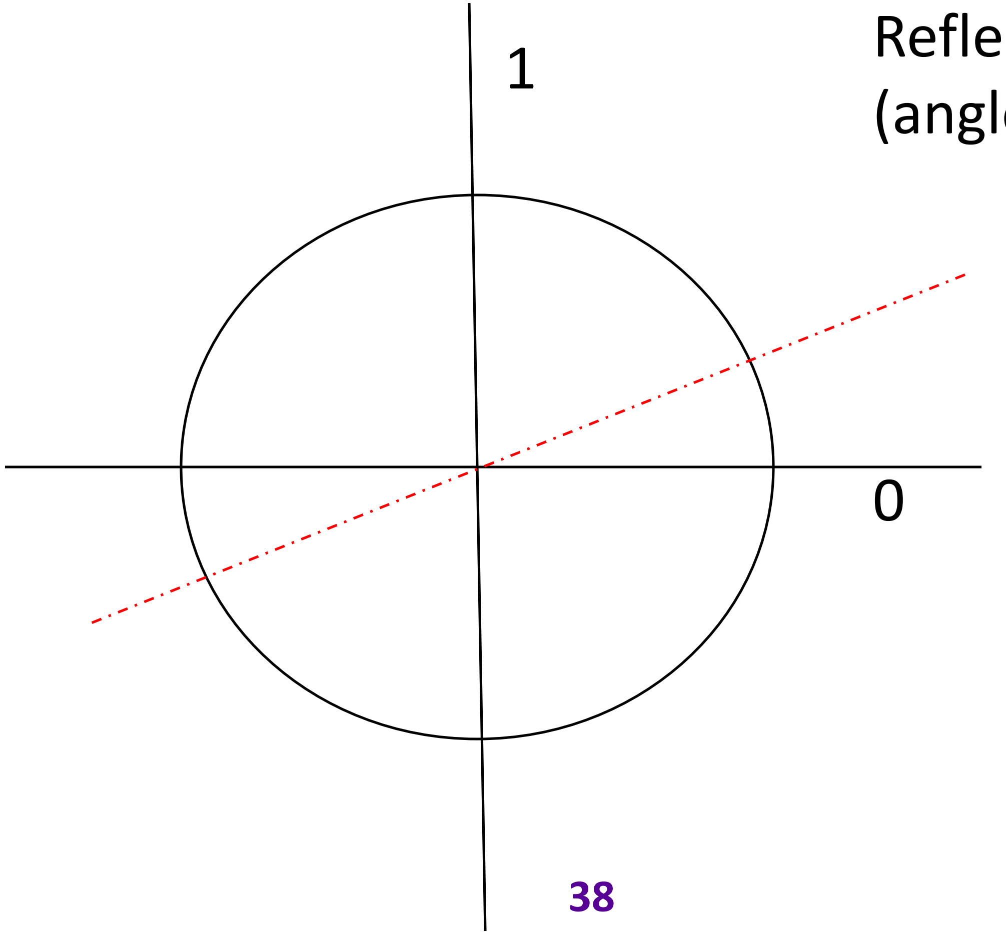
Hadamard Gate

- Sets a qubit that is either 0 or 1, to an equal superposition of 0 or 1

Hadamard Truth Table

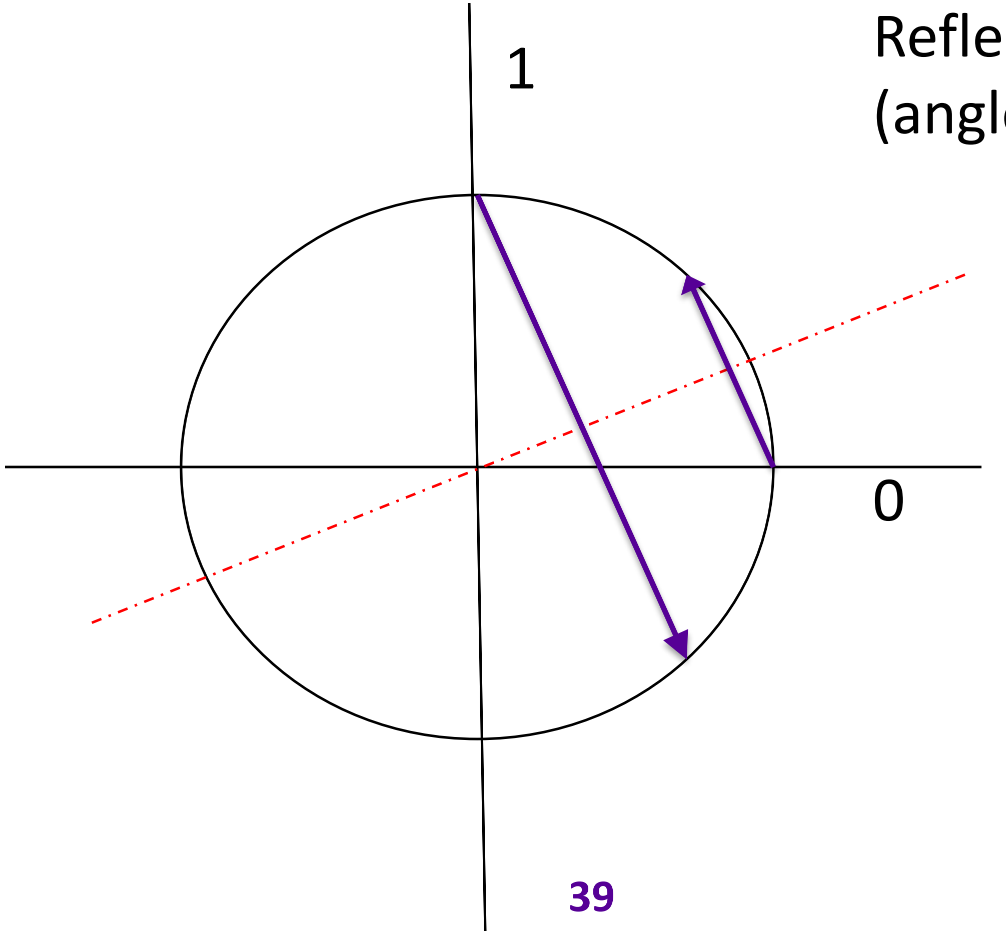
- $|0> \rightarrow \frac{1}{\sqrt{2}}\,|\,0> + \frac{1}{\sqrt{2}}\,|\,1>$
- $|1> \rightarrow \frac{1}{\sqrt{2}}\,|\,0> - \frac{1}{\sqrt{2}}\,|\,1>$
- Note that both of those have same probabilities of reading 0 or 1, even though the phase of |1> is different

Picturing Hadamard Gate



Reflect around red line
(angle 22.5 degrees from 0)

Picturing Hadamard Gate



Reflect around red line
(angle 22.5 degrees from 0)

RSAConference2019

Multi-qubit states

Entanglement

- A set of entangled qubits has a state
- If a group of qubits is entangled, impossible to describe collective state by talking about the states of individual qubits
- With 3 entangled qubits, the state will be a superposition of
  - 000, 001, 010, 011, 100, 101, 110, 111
- State of group of 3 entangled qubits:

$\alpha|000> + \beta|001> + \gamma|010> + \delta|011> + \varepsilon|100> + \zeta|101> + \eta|110> + \theta|111>$

$$|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 + |\varepsilon|^2 + |\zeta|^2 + |\eta|^2 + |\theta|^2 = 1$$

Multi-qubit states

- If a set of n qubits is NOT entangled, the state can be expressed compactly with 2n coefficients
  - For each of n qubits, coefficient of 0 and coefficient of 1
- If they are entangled, it takes 2^n coefficients
  - For each of the 2^n states, coefficient of that state
- Example: 10 qubits:
  - Unentangled requires 20 coefficients
  - Entangled requires 1024 coefficients

Entanglement makes quantum really powerful

- An entangled set of n qubits holds a superposition of 2^n different values
  - All at the same time, unlike classical…
- The quantum computer computes on all 2^n in parallel
- Without entanglement, a quantum computer would be no more powerful than a classical computer

Operating on entangled qubits

- There are really just one or two qubit gates
- So, what happens when you operate on a subset of an entangled set of qubits?
- The other qubits don't change, but remain entangled

For example, NOT operation on the first qubit of 3 entangled qubits

$\alpha\ |000> + \beta\ |111>$

- Apply NOT to the first qubit
- You’ll get

$\alpha\ |100> + \beta\ |011>$

Typical Quantum Program Starts with…

- Suppose you want to operate in parallel on all possible 2^n classical values of n qubits
  - Initialize all n qubits to 0 (measure them and invert if they are 1)
  - Apply Hadamard to each of them
  - Now all 2^n classical values are equally probable
    - probability of each value is $1/2^n$
    - coefficient of each one is $1/2^{n/2}$)

The Computing Model

What makes a quantum computer powerful

- A circuit that operates on n qubits, is operating on all (up to 2^n) superposed values simultaneously
- Though not quite as powerful as that…
  - Quantum gates (that can be built) operate on one or two qubits
  - A logical n-qubit gate would be built out of lots of one or two qubit gates
  - The “running time” of a circuit which is logically an n-qubit gate is adjusted to account for how many actual gates would be needed to construct the circuit

Limitations

- You can't measure the qubits without losing superposition
- No Cloning Theorem
  - You can't copy a qubit

    $\alpha|0> + \beta|1>$
  - You could XOR its value into a qubit initialized to 0
  - BUT: you wouldn't wind up with what you want, which is two independent qubits, each in state

    $\alpha|0> + \beta|1>$
  - Instead you'd wind up with two entangled qubits in state

    $\alpha|00> + \beta|11>$

RSAConference2019

Grover’s Algorithm

To get a feel for a quantum algorithm

- A peek into Grover’s algorithm
- Shor’s algorithm is also really cool, but not enough time in this talk for both, and Shor’s involves really understanding Fourier transforms, which many people either never understood, or have forgotten

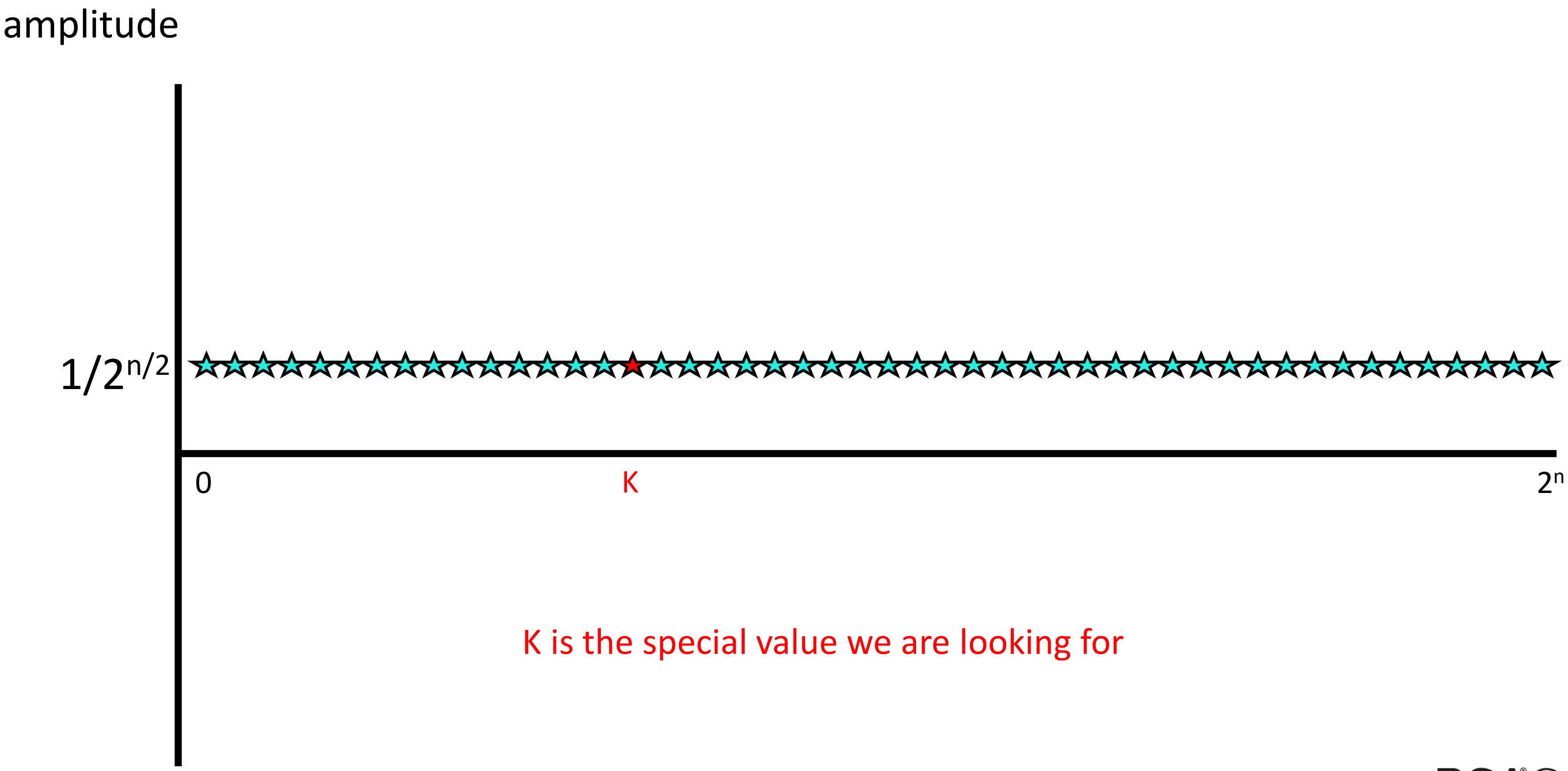
Grover's Algorithm

- Problem it's solving: brute force search, for instance
  - Which n-bit key K takes plaintext X to ciphertext Y?
- Brute force search on a classical computer would take 2^n tries
- Assume
  - exactly one correct answer (e.g., one value K that takes plaintext X to ciphertext Y)
  - We have function f (input x), outputs "yes" or "no"
    - E.g., does this n-bit input used as an AES key map X to Y?
- Note: We're not doing the optimal version of Grover's…we're doing what we think is easiest for intuition

Very High Overview of Grover

- Want to find the n-bit key K (that takes plaintext X to ciphertext Y)
- First step: for each of n qubits
  - Initialize it to 0, then apply Hadamard
- Now each of the 2^n possible values are equally likely

Initial amplitudes of each of the 2^n values



K is the special value we are looking for

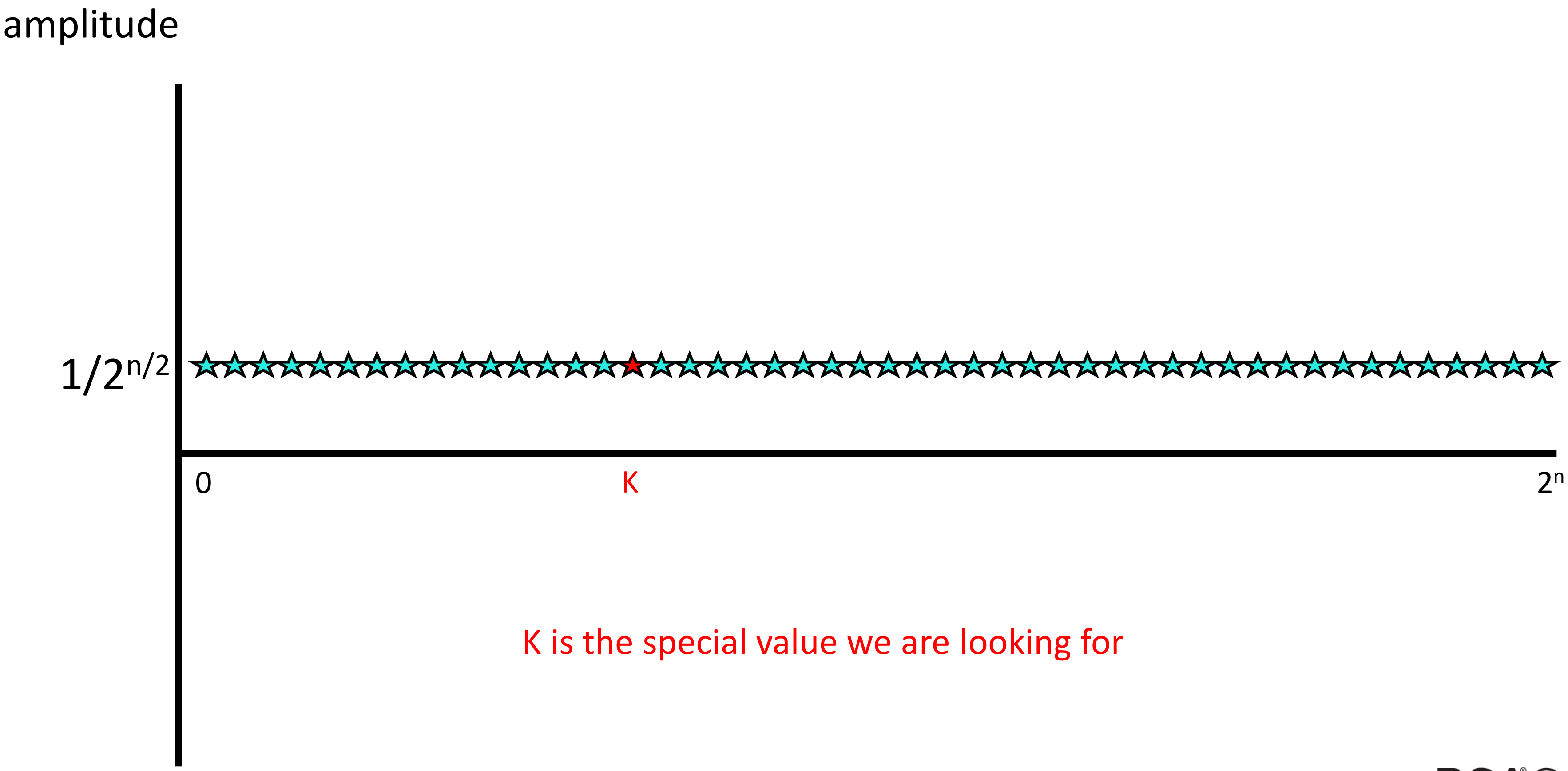
Very High Overview of Grover

- Have an operation on the n qubits that raises the probability of reading K by a little bit
  - Do this operation approximately $2^{n/2}$ times
- Then measure the qubits, and it will be highly probable you'll read the value K
- Note: You do not know K in advance
- The magic is that the quantum circuit can boost the probability of getting the value K when you read the qubits

Boosting the amplitude of K

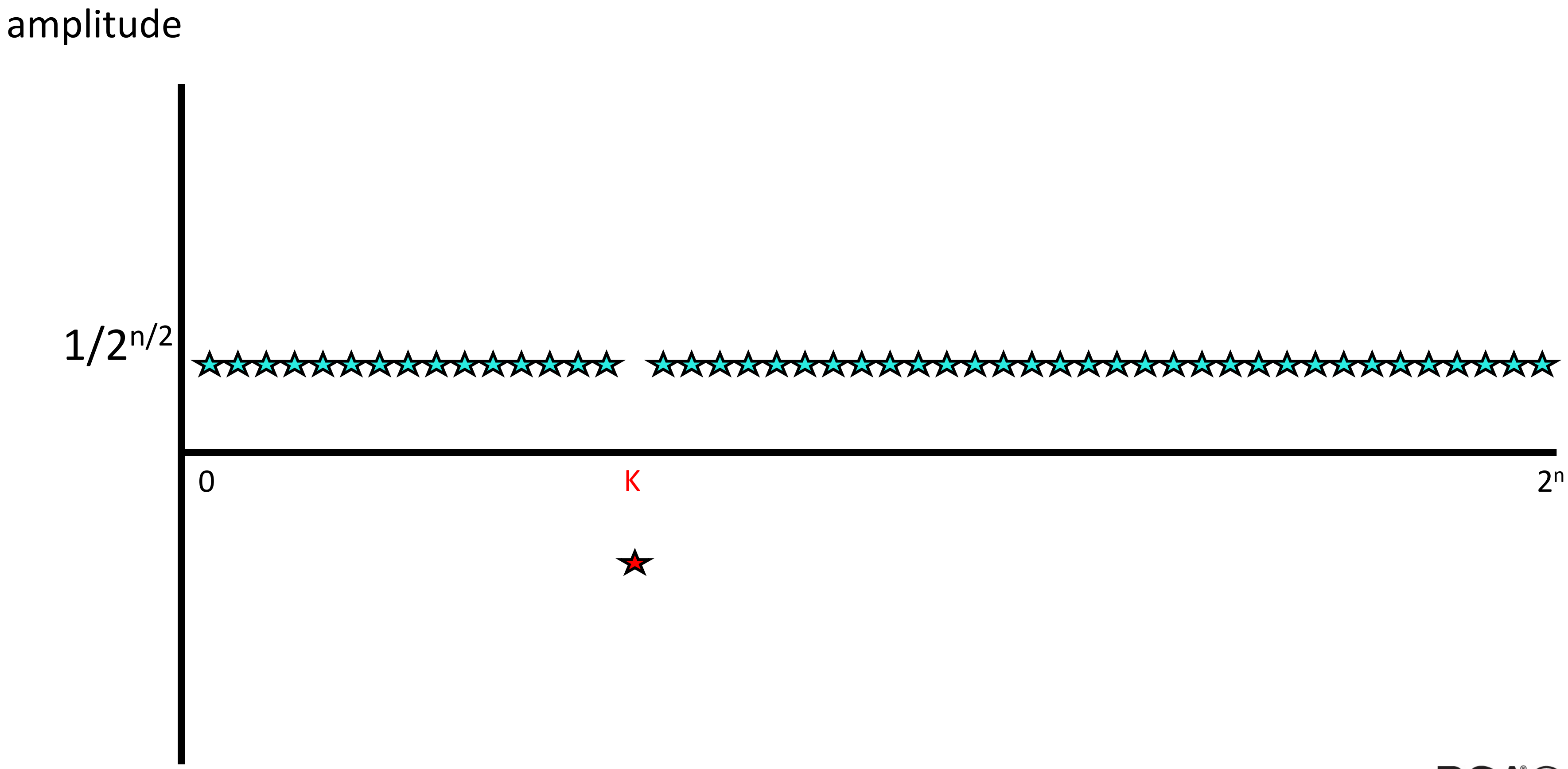
- Doing these two steps boosts the probability of reading K by a little bit
  - Invert amplitude (coefficient) of K (multiply it by -1)
  - Reflect all amplitudes around the mean of all the amplitudes
- Do these two steps about $2^{n/2}$ times

Initial amplitudes of each of the 2^n values



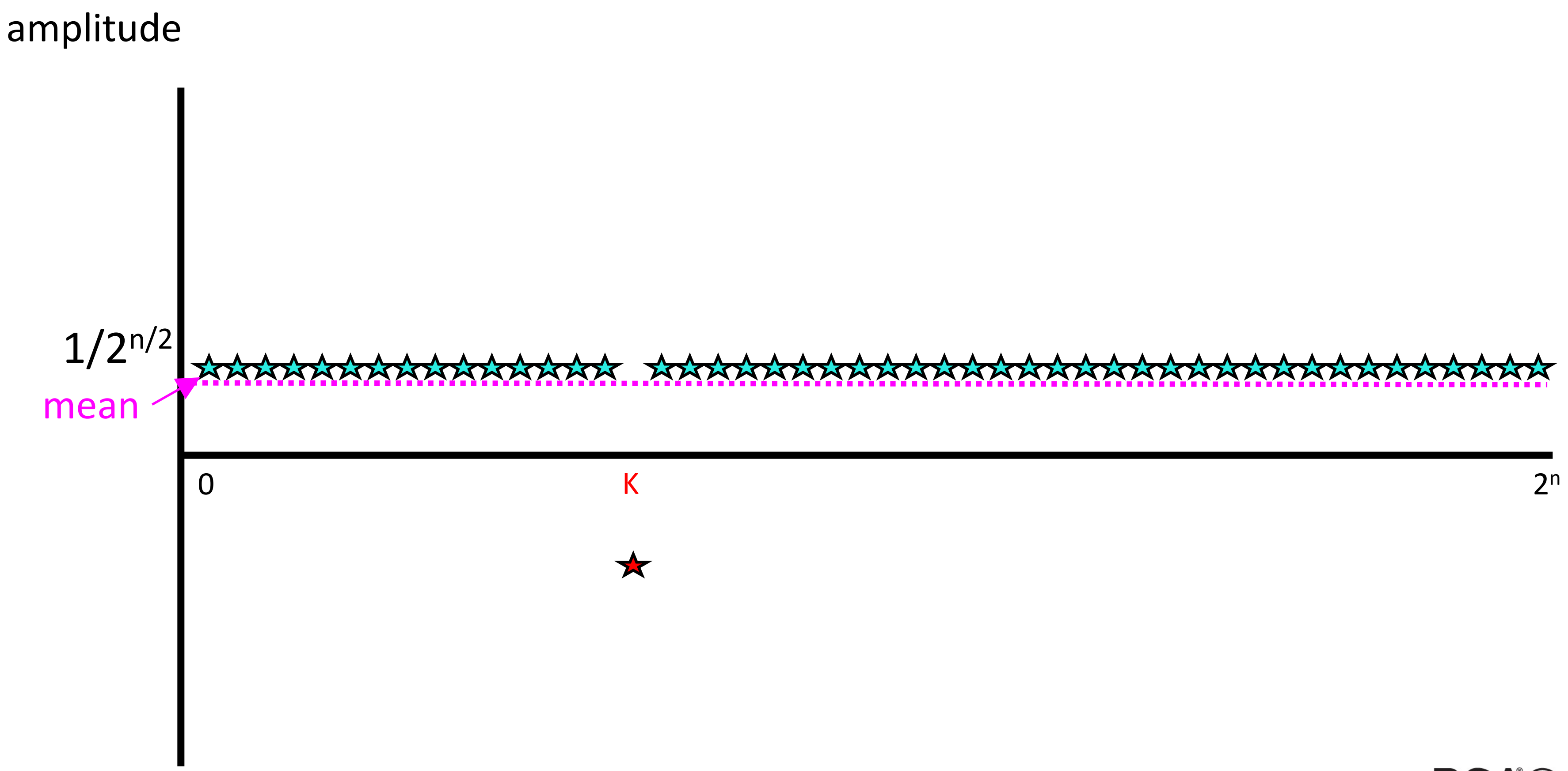
K is the special value we are looking for

Invert amplitude of K

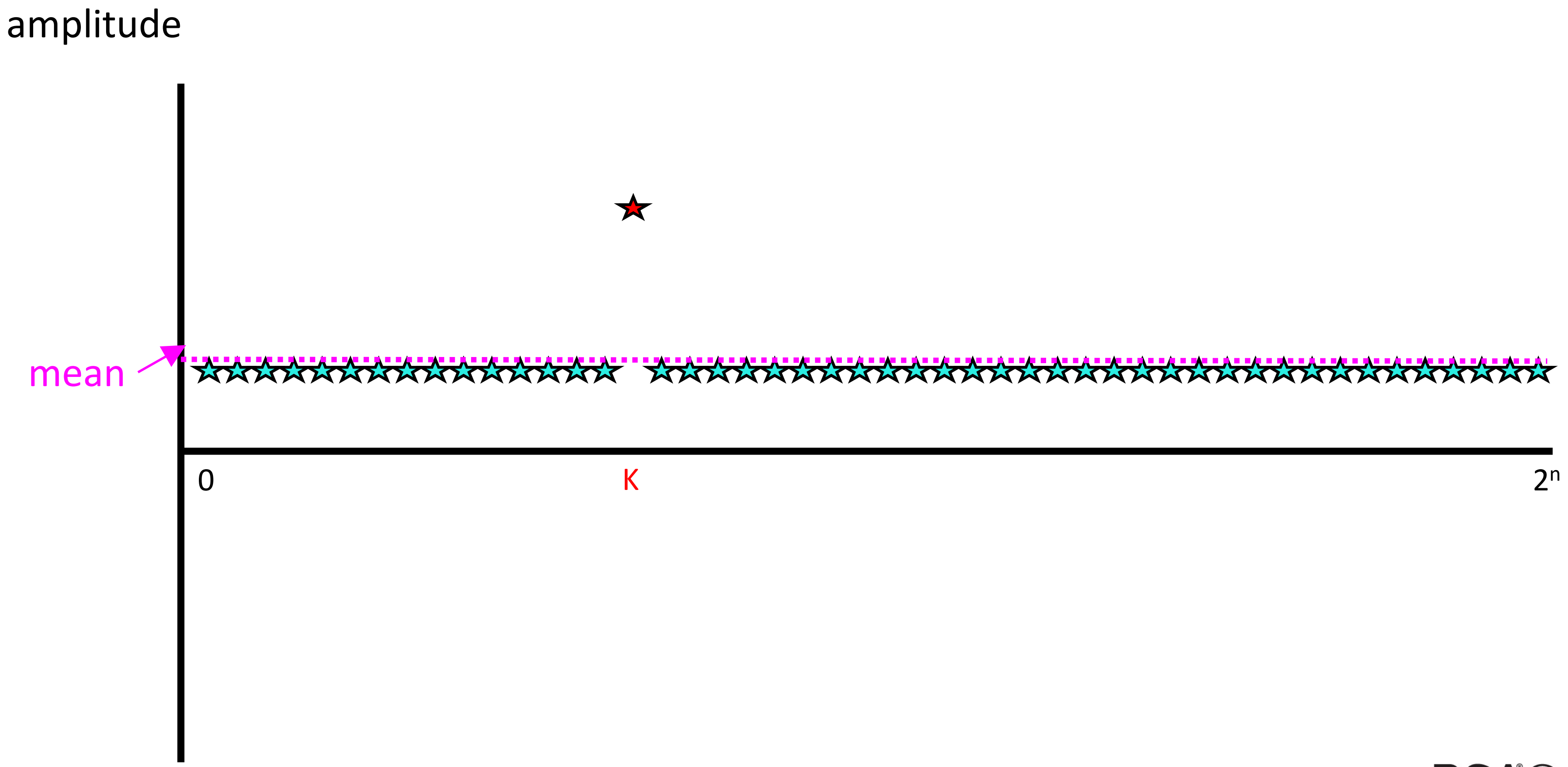


Next step: Flip all amplitudes around the mean

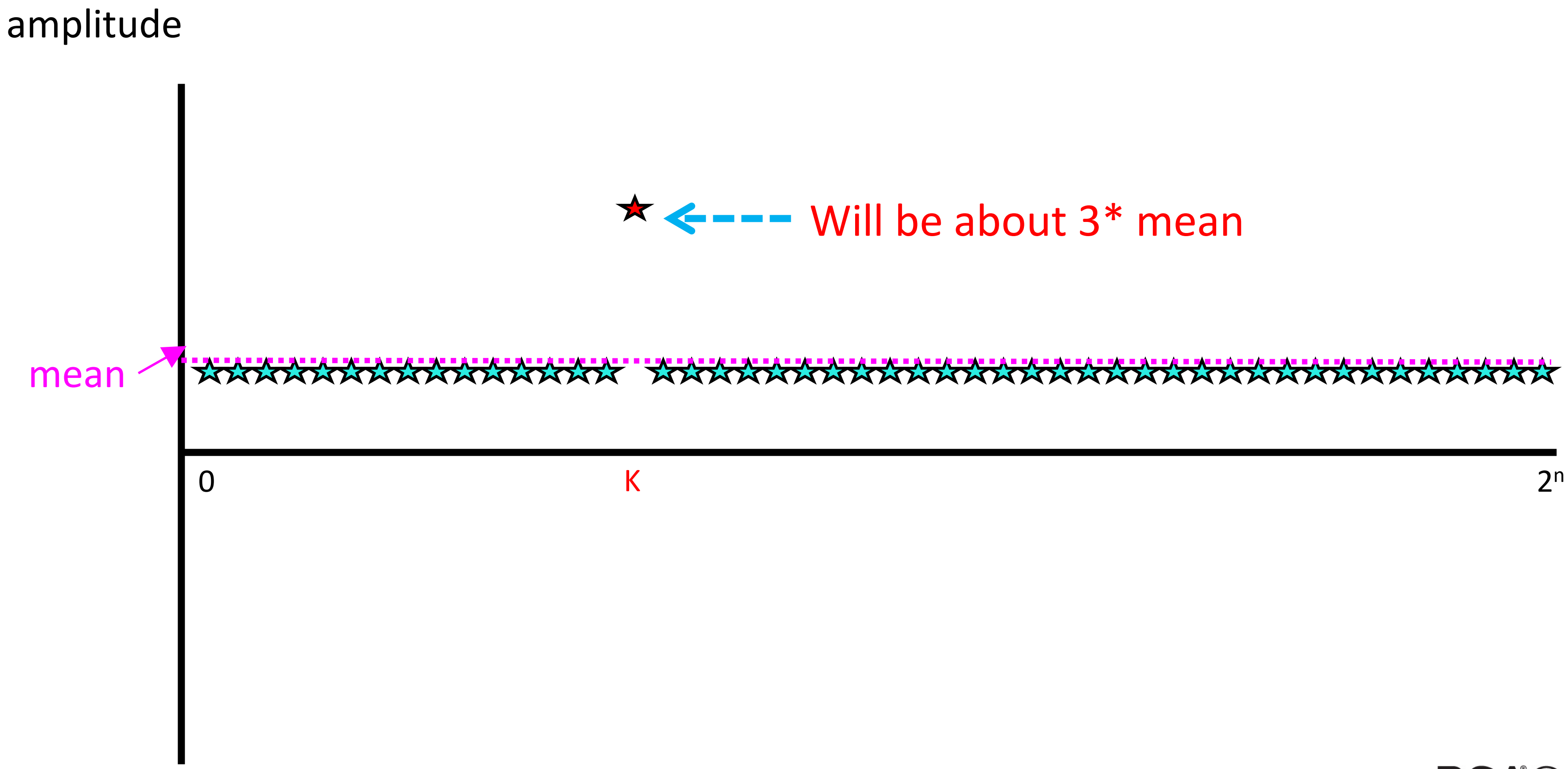
What is the mean?



Reflect all amplitudes around mean

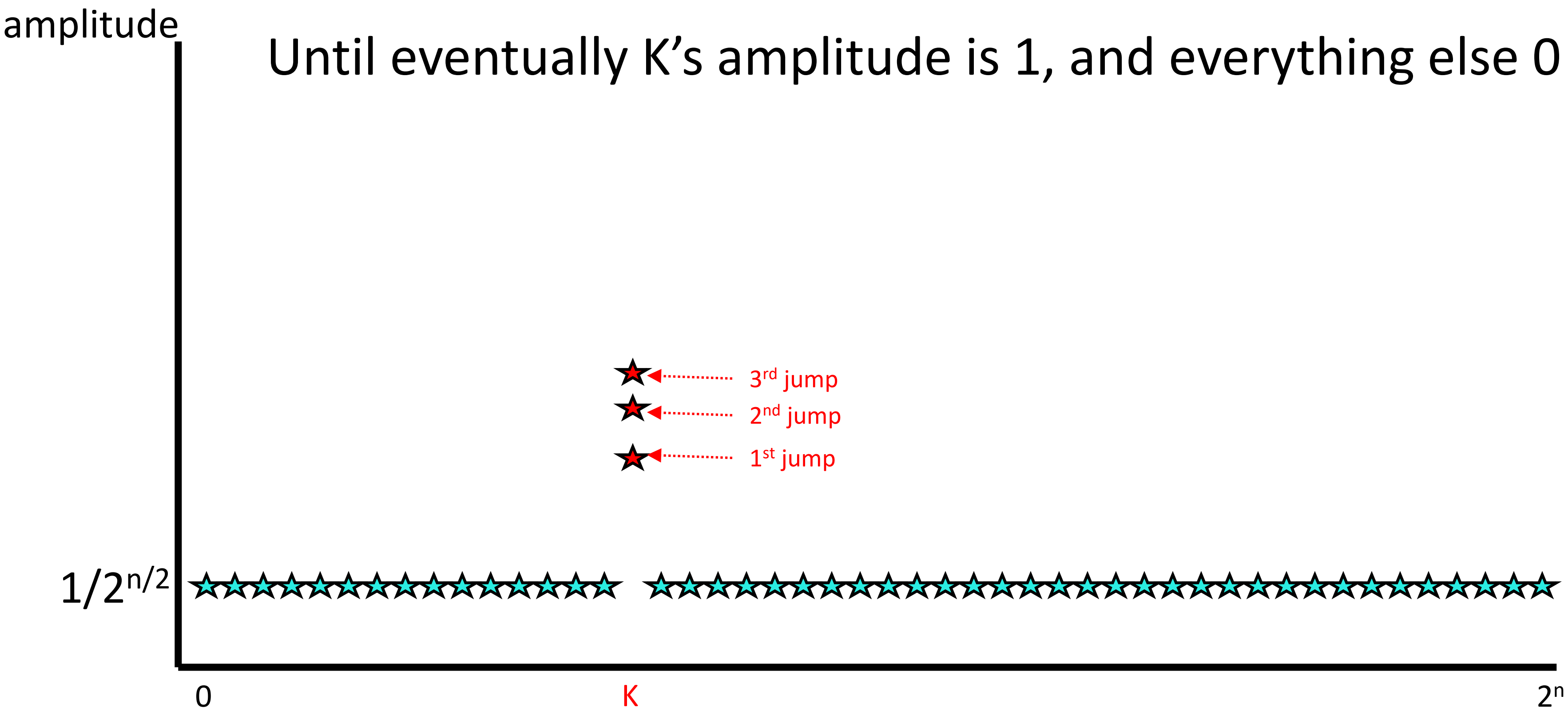


Reflect all amplitudes around mean

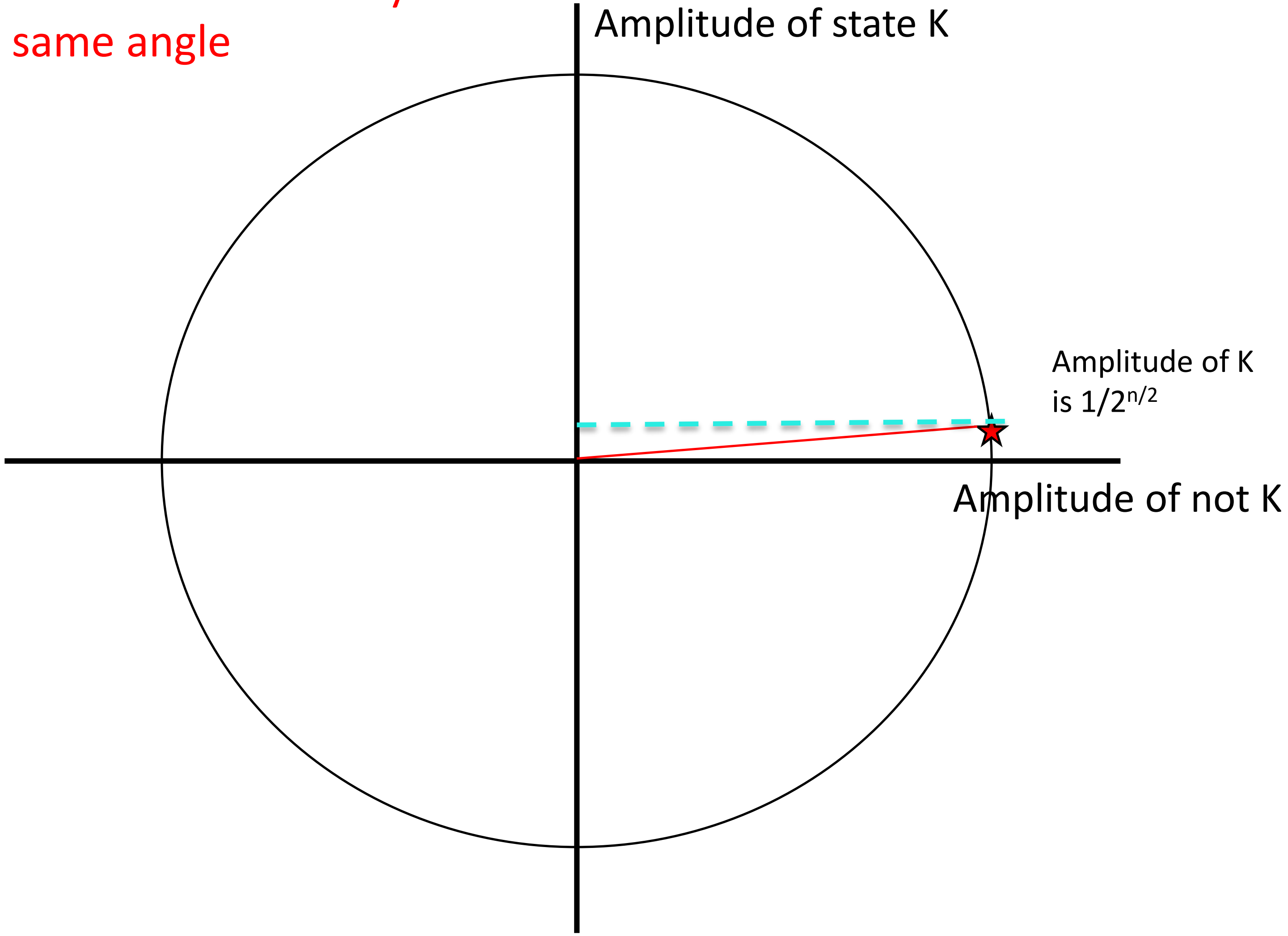


Each iteration increases K's amplitude

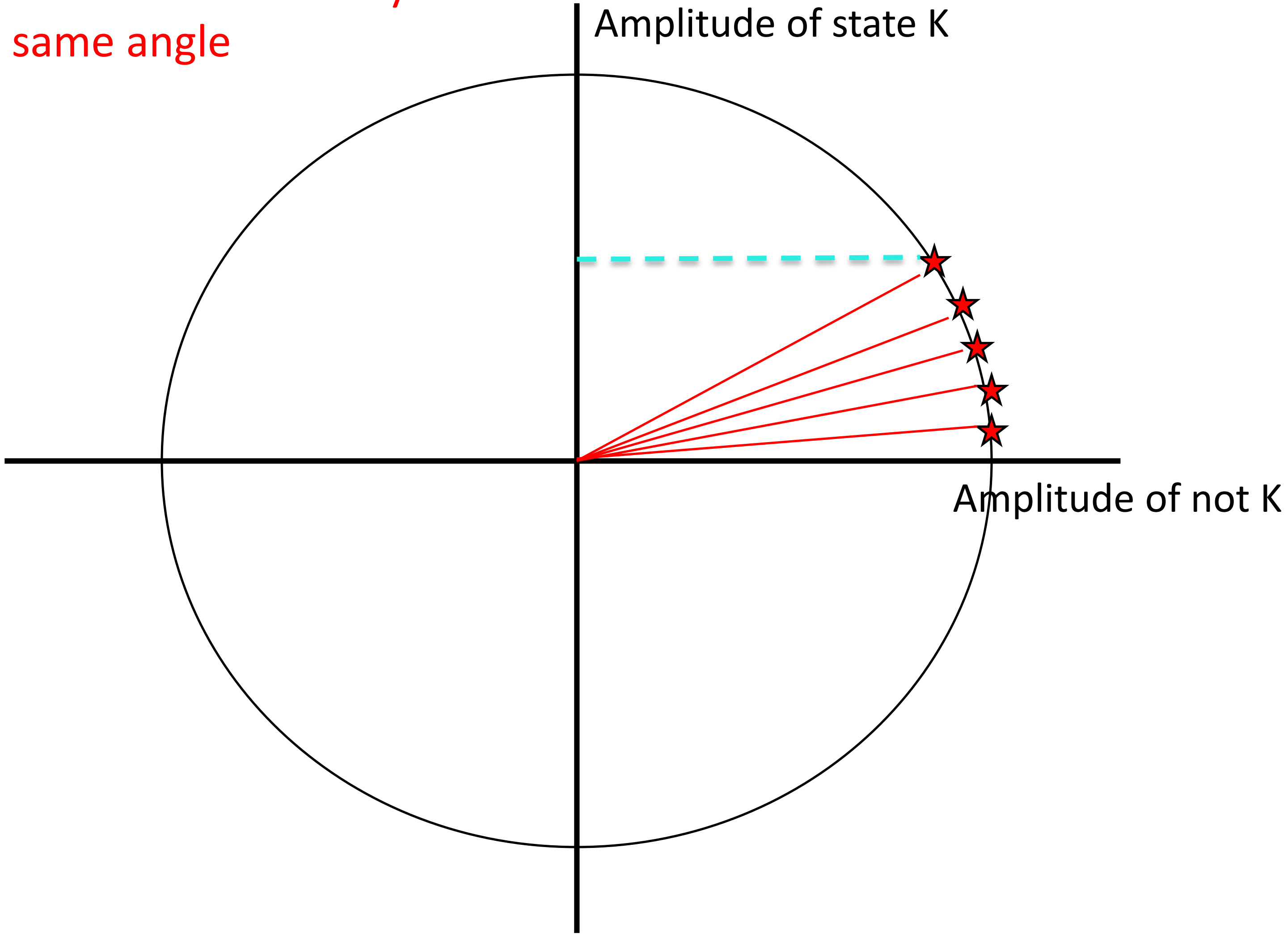
Until eventually K's amplitude is 1, and everything else 0



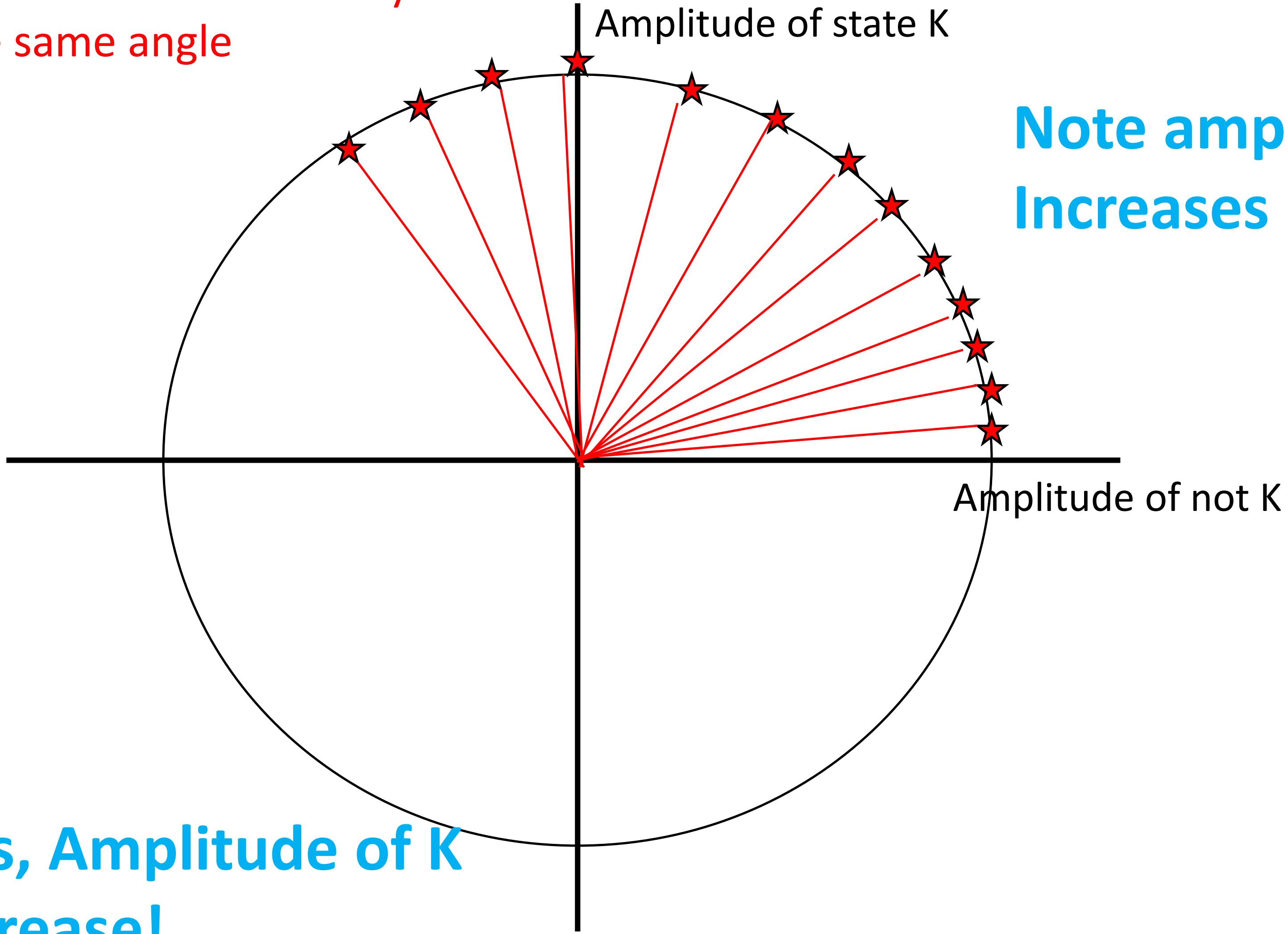
Each iteration rotates by the same angle



Each iteration rotates by the same angle



Each iteration rotates by the same angle



Note amplitude of K Increases most at start

after 90 degrees, Amplitude of K will start to decrease!

How do we do these operations on a quantum computer?

- We know how to initialize to “all 2^n values equally likely”
- How do we multiply the amplitude of K by -1?

How to multiply amplitude of K by -1

- Use an ancilla (an extra qubit), initialized to 0
- Use function f ("is x the right value?"), on n-bit input x:
  - If f(x)=no, don't change ancilla
  - If f(x)=yes, perform NOT on ancilla
- Note that f is operating simultaneously on all 2^n superposed values of the n qubits

Before performing f

x	ancilla
0000	0
0001	0
0010	0
0011	0
0100	0
0101	0
0110	0
0111	0
1000	0
1001	0
1010	0
1011	0
1100	0
1101	0
1110	0
1111	0

After performing f, simultaneously on all 16 values of the 5 qubits

x	ancilla
0000	0
0001	0
0010	0
0011	0
0100	0
0101	1
0110	0
0111	0
1000	0
1001	0
1010	0
1011	0
1100	0
1101	0
1110	0
1111	0

Before performing f

x	ancilla
0000	0
0001	0
0010	0
0011	0
0100	0
0101	0
0110	0
0111	0
1000	0
1001	0
1010	0
1011	0
1100	0
1101	0
1110	0
1111	0

After performing f, simultaneously on all 16 values of the 5 qubits

x	ancilla
0000	0
0001	0
0010	0
0011	0
0100	0
0101	1
0110	0
0111	0
1000	0
1001	0
1010	0
1011	0
1100	0
1101	0
1110	0
1111	0

5 entangled qubits, indicating 0010 is NOT the answer

Before performing f

x	ancilla
0000	0
0001	0
0010	0
0011	0
0100	0
0101	0
0110	0
0111	0
1000	0
1001	0
1010	0
1011	0
1100	0
1101	0
1110	0
1111	0

After performing f, simultaneously on all 16 values of the 5 qubits

x	ancilla
0000	0
0001	0
0010	0
0011	0
0100	0
0101	1
0110	0
0111	0
1000	0
1001	0
1010	0
1011	0
1100	0
1101	0
1110	0
1111	0

5 entangled qubits, indicating 0101 IS the answer

Before performing f

x	ancilla
0000	0
0001	0
0010	0
0011	0
0100	0
0101	0
0110	0
0111	0
1000	0
1001	0
1010	0
1011	0
1100	0
1101	0
1110	0
1111	0

After performing f

x	ancilla
0000	0
0001	0
0010	0
0011	0
0100	0
0101	1
0110	0
0111	0
1000	0
1001	0
1010	0
1011	0
1100	0
1101	0
1110	0
1111	0

But reading any of these is equally likely.

We want to boost probability of reading K

We now have n+1 entangled qubits

- If you read the n+1 qubits, you'd get a value and "yes/no" for that value
- At first, almost certainly you'd get some number and "nope, that's not the answer"
- So we want to boost the probability we'll read (K | yes!)

To invert amplitude of K

- Perform “Z gate” on ancilla

  |0> → |0>

  |1> → -|1>

- Note: We’ve multiplied amplitude of K|1 by -1!
  - Without changing the phase of any of the other superposed states

After Z gate (invert amplitude of amplitude if qubit=1)

x	ancilla	amplitude
0000	0	1/4
0001	0	1/4
0010	0	1/4
0011	0	1/4
0100	0	1/4
0101	1	- 1/4
0110	0	1/4
0111	0	1/4
1000	0	1/4
1001	0	1/4
1010	0	1/4
1011	0	1/4
1100	0	1/4
1101	0	1/4
1110	0	1/4
1111	0	1/4

Then do f again, to zero out the ancilla

x	ancilla	amplitude
0000	0	1/4
0001	0	1/4
0010	0	1/4
0011	0	1/4
0100	0	1/4
0101	0	- 1/4
0110	0	1/4
0111	0	1/4
1000	0	1/4
1001	0	1/4
1010	0	1/4
1011	0	1/4
1100	0	1/4
1101	0	1/4
1110	0	1/4
1111	0	1/4

The other Grover operation

- Reflect around the mean
- Similar in spirit to “multiply amplitude of k by -1”
- But a bit more complicated to explain

Why Grover isn't devastating

- At best square root of number of keys (so double the size of key)
- And parallelism doesn't help as much as with classical
  - With classical, can use a million computers and divide work to be a million times faster
  - With Grover, the "million" comes off the original number, so you only save square root of a million (thousand times faster)

RSAConference2019

New Public Key algorithms

Making a scheme

- Find a math problem that’s probably hard
- Turn it into a crypto scheme
- Do optimizations to make it practical
- Current schemes based on various math problems, e.g.,
  - Hashes
  - Error Correcting Codes
  - Lattices
  - Multivariate (quadratic) equations

There won’t be a single “best” algorithm

- There will be different tradeoffs in terms of
  - Computation required for key generation, encryption, signatures, etc.
  - Size of keys
  - Size of signatures
- And usually, each scheme is just good for encryption or for signing

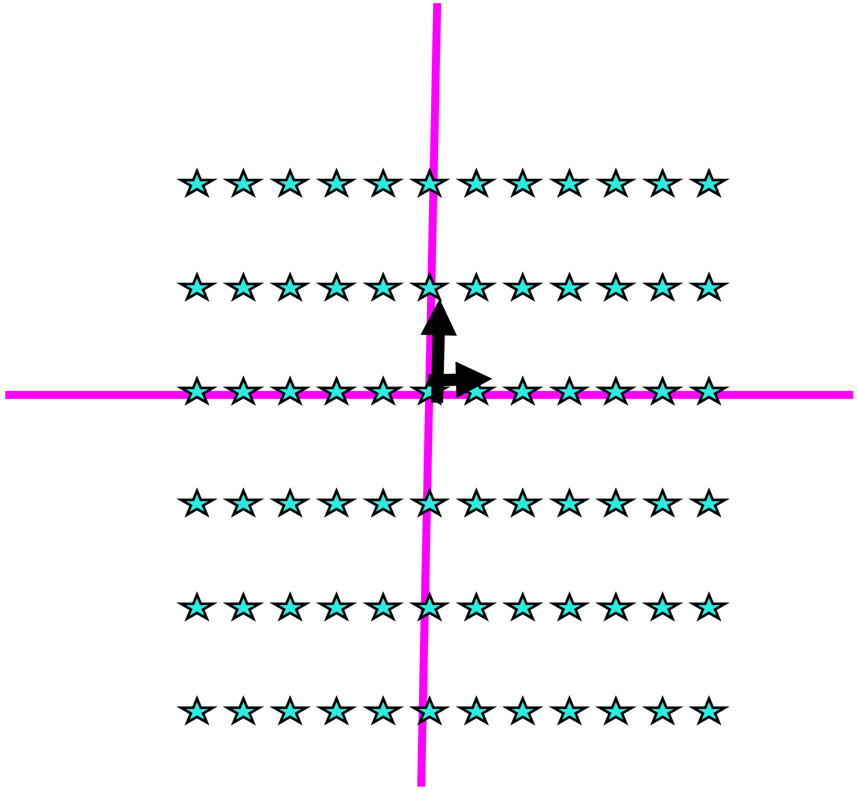
RSAConference2019

Intuition behind one scheme

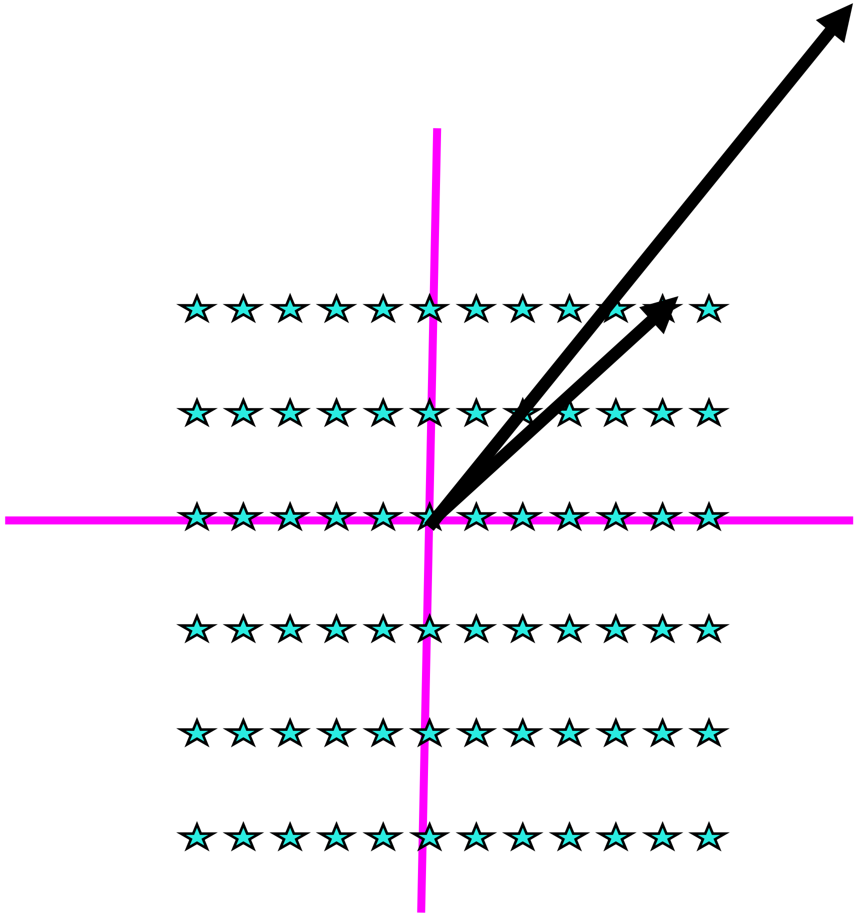
An intuitive Lattice-Based Encryption Scheme

- A lattice is a set of points generated from a “basis” of vectors
- Where any integer linear combination of the basis vectors is a point in the lattice

Lattice with basis (0,3), (1,0)



Alternative basis (5,6), (12,15)



An assumed-to-be-hard lattice problem

- Find the nearest lattice point, given a point in n-dimensional space
- Difficult if you only know a “bad basis”
- But easy if you know a “good” (short) basis
- It would be easy in 2 dimensions, but we’re talking about hundreds of dimensions

How Alice creates a (public, private) key pair

- She generates a lattice (n linearly independent small n-dimensional vectors)
  - That's her private key
- She creates a bad basis (linear combinations of her basis vectors)
  - The bad basis is her public key

How Bob can agree on a secret with Alice

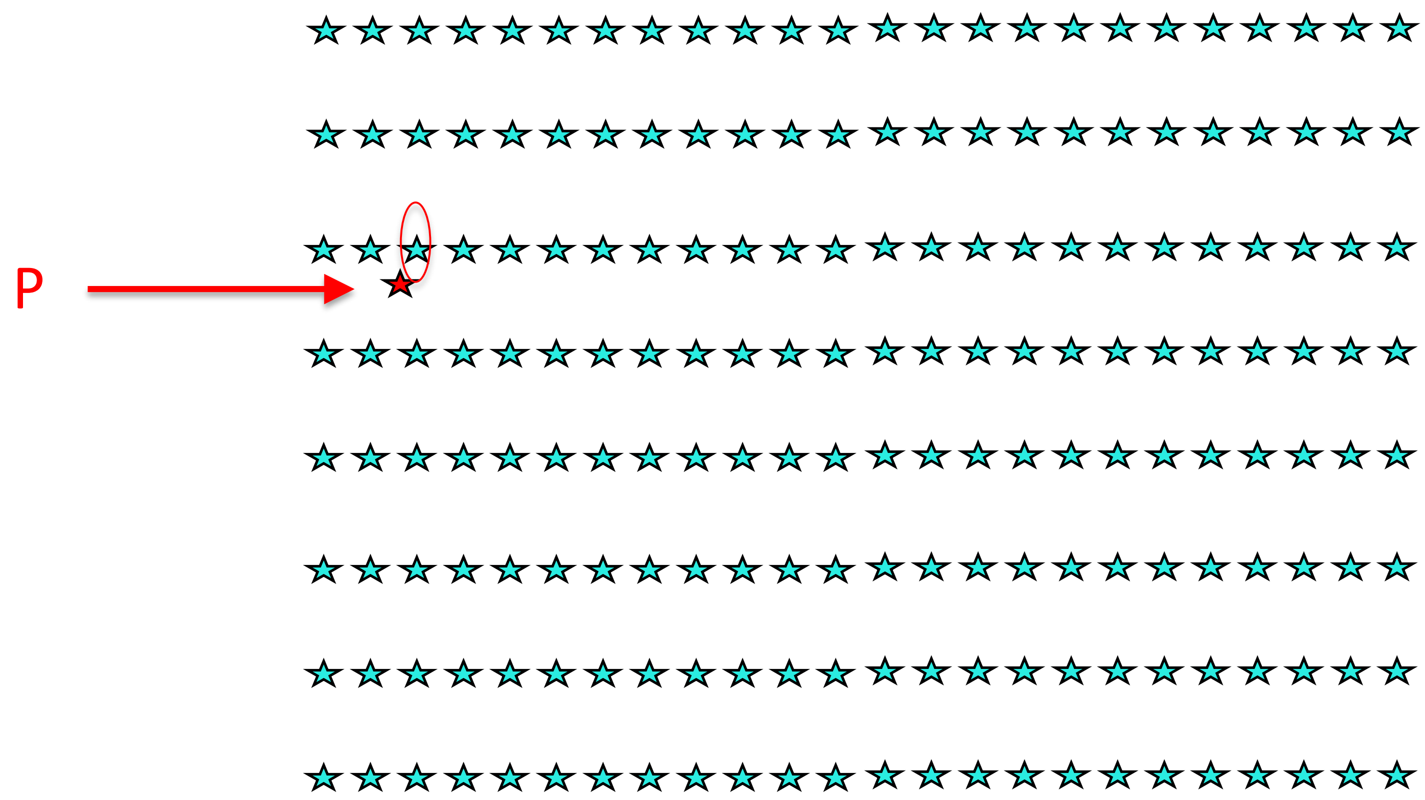
- Bob is going to choose an n-dimensional vector with small coefficients
- Let’s call this “E”
- This will be an offset from a lattice point
- The secret Bob and Alice will share is h(E)

How Bob sends E to Alice

- Using Alice's public key (bad basis), he chooses a random lattice point X
- He adds E to X to get a (non-lattice) point P
- Alice, using her good basis, can find X
- Then she subtracts P from X to get E

Bob chooses random lattice point X

Adds small E, sends that (non-lattice pt) P to Alice



Given P, Alice finds X,
Computes P-X = E
Secret is h(E)

RSAConference2019

Challenges in building quantum computers

Qubits are Temperamental

- Often implemented as properties of subatomic particles (e.g., electron spin, photon polarity, electron energy level)
- Very difficult to accurately measure
- Gates require that they interact in a controlled fashion
- State is destroyed if they interact with their environment, so require
  - Near absolute zero temperature, or
  - Near-perfect vacuums
- Spontaneously decay with half lives counted in microseconds

Quantum Computers are inherently slow and energy intensive

- Computation consumes energy which heats the qubits
- Fast computation heats them faster than we can cool matter at 10 milli-Kelvins
- To a first approximation, with current known technology, quantum computers will be a million times slower and a trillion times less energy efficient than classical computers (quantum error correction responsible for a lot of the overhead)
- Predictions for when a quantum computer capable of breaking 2048-bit RSA range from "never" to 2030
  - with energy requirement= a nuclear power plant)
- Of course there might be unforeseen breakthroughs

Challenges

- Making qubits that are as reliable as possible
- Making gates that are as reliable as possible
- Doing error-correction

Quantum Error Correction

- As with digital circuits where extra bits can correct for errors in some bits
- The more errors you need to correct, the more extra bits you need
- Some large number of qubits (e.g., 10-1000) can be used to correct for errors in a few of them
- If error rate in physical qubits low enough to run the error correction algorithm, in theory you can build long lived extremely reliable "logical qubits"

What kinds of problems are suggested for quantum computers?

- Breaking our public key algorithms, of course
- Things with quadratic speedups (e.g., Grover’s).
  - But there would need to be a dramatic improvement in price/performance of quantum computers if they would be more cost-effective than using classical computers and parallel computation
  - So really need exponential speedup

What kinds of problems are suggested for quantum computers?

- Big data
  - A single qubit can be in an infinite number of states; a set of entangled qubits exponentially more so
  - Sounds like a great way to store a lot of information in a small amount of space
  - However…since measurement destroys the superposition, it's not all that useful
- Optimization problems
  - In practice, the "optimal" solution is not all that important. What's needed is a "pretty good" solution
    - And it's not obvious that a quantum solution for "pretty good" is any better than a heuristically chosen "pretty good" solution on a classical computer
  - And for "optimal", there aren't currently known optimization problems that a quantum computer would be better at finding the optimal solution, especially when cost factored in

What kinds of problems are suggested for quantum computers?

- Other potential problems
  - Modeling/Predicting Chemical Reactions
  - Neural Networks

RSA Conference 2019

Summary: Take-aways (“Apply”)

Take-aways

- Hopefully you've gained a feel for what a quantum computer is and what it can do
- The only known "killer app" is breaking current public key algorithms
- Most of us will not be involved in trying to build quantum computers
- But it's good to support such research
- But, we all will need to be thinking about converting from RSA to quantum-safe public key algorithms
- Which we can trust NIST, and cryptographers, to develop and standardize
- But replacement may not be that simple, because new algorithms might have different characteristics (e.g., huge key sizes), which might require rethinking implementations and protocols