Quantum Key Distribution Cheat Sheet

Cryptography uses a **cipher** (also called a cypher or a **code**) which is a type of algorithm that allows two parties to communicate in a way that **eavesdroppers** cannot understand

A **key** is a parameter that defines the output of a **cipher** algorithm

We assume the **cipher** is **public**ly known but the **key** is secret

The process of converting the secret message into a code is called **encryption**

The process of converting the code back to the secret message is called **decryption**

Right-Shift Caesar Cipher

Plain	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**
Cipher	X	Y	Z	A	B	C	D	E	F	G	H	I	J
Plain	**N**	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**
Cipher	K	L	M	N	O	P	Q	R	S	T	U	V	W

QUANTUM ↔NRXKQRJ

Called a **substitution** cipher because you *replace* the letters with other letters

The **key** for this right shift substitution cipher would be **3** (corresponding to the number of places we shift the letters by)

These ciphers are easy to encode and decode by hand, but that also makes them easy to crack

- Max 25 tries for Caesar cipher (assuming latin alphabet)
- Max N/2-1 tries (loose bound) for rail-fence cipher

Substitution and **transposition** are the two main operations used in many **ciphers** today – but they are combined to be much more complex

Rail-Fence Cipher

Q				T		
	U		N		U	
		A				M

QUANTUM ↔ QTUNUAM

Called a **transposition** cipher because you *rearrange* the letters

The **key** for this rail fence transposition cipher would be **3** (corresponding to the "height" of the rail fence)

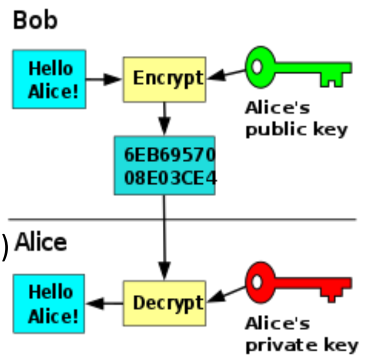
```
def cipher(key, message):
  n = key[0]
  e = key[1]
  m = message[100:]
  c = m**e % n
  return c
```

Modern **ciphers** are built to have **no possible algorithmic speedups**

- You can only crack them by **brute force** testing keys
  - This is the same as **searching an unstructured list** – classically **O(N)**
- Standard keys are 128 bits → $N = 2^{128} \approx 10^{34}$
  - Assuming 1.5ps gate time, it would take $\sim 10^{19}$ years to crack
    - The universe is only $\sim 10^{10}$ years old -- **uncrackable**
- **Grover's algorithm** can give us quadratic speedup: 128 bits → $2^{64} \approx 10^{19}$
  - With same assumptions, could crack a 128-bit key in ~**1 year**!!
  - **Solution**: double key length to 256 bits and we're back to $\sim 10^{19}$ years to crack

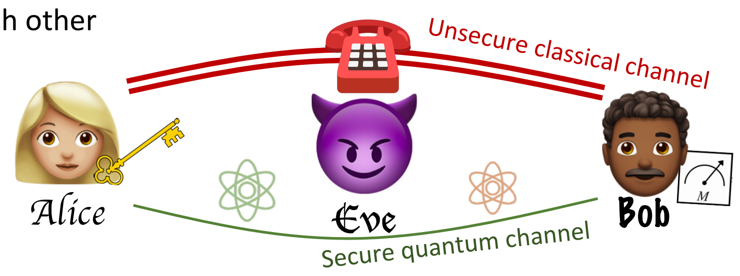
Types of modern cryptography

- **Private** (or **symmetric**) key
  - Uses a cipher
  - Requires **secure key distribution** – which is very hard to do
    - **QKD** solves key distribution problem!
  - Much faster than asymmetric key cryptography
- **Public** (or **asymmetric**) key
  - Uses mathematically related public/private key pairs (not a cipher)
    - Your **private key** is kept **secret** and used to **decrypt** messages intended for you
    - Your **public key** is sent out so that others can **encrypt** messages for you
  - Does not require secure key distribution
  - Susceptible to algorithmic speedups
    - **Quantum-insecure**: **Shor's algorithm** breaks all of the most widely-used public key encryption algorithms (RSA, elliptic curve)



https://en.wikipedia.org/wiki/Public-key_cryptography

Alice and Bob want to create a secret key so that they share secret messages with each other



- They don't want Eve (man-in-the-middle) to overhear their messages
- They can use the Quantum Key Distribution (**QKD**) protocol **BB84** to achieve a verifiably secure key distribution

BB84

1. **SELECT ENCODING:** Alice randomly selects a basis (Z or X) to encode each bit

Message	0	1	0	1	1	1	0	1
Encoding basis	X	Z	X	X	Z	Z	Z	X

2. **SELECT MEASUREMENT:** Bob randomly selects a basis (Z or X) to measure each bit.

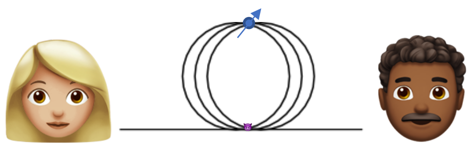
Measure basis	Z	Z	X	Z	X	X	Z	X

3. **Q. ENCODE:** Alice creates the quantum states, encoded in the elected bases.

	0	1
Z-basis	$\vert 0\rangle$	$\vert 1\rangle$
X-basis	$\vert +\rangle$	$\vert -\rangle$

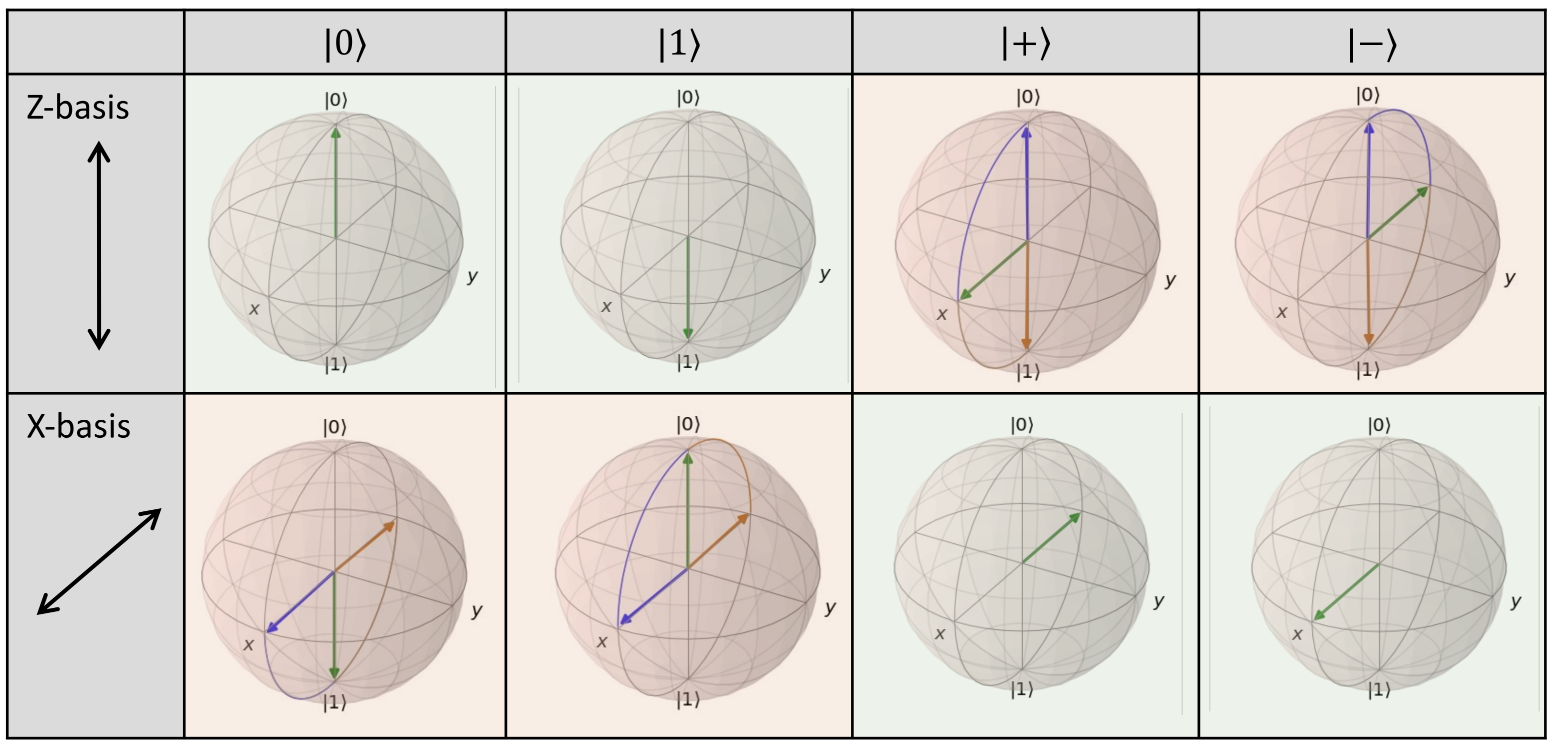
Encoded states	$\vert +\rangle$	$\vert 1\rangle$	$\vert +\rangle$	$\vert -\rangle$	$\vert 1\rangle$	$\vert 1\rangle$	$\vert 0\rangle$	$\vert -\rangle$

4. **Q. SEND:** Alice sends Bob the encoded states, via the quantum channel.



Optical fiber

5. **Q. MEASURE:** Bob measures all the quantum states in his pre-selected measurement bases.



Measure message	0	1	0	1	1	0	0	1

6. **C. ANNOUNCE BASIS:** Alice announces which basis she used to encode each bit via the classical channel

Basis	X	Z	X	X	Z	Z	Z	X

7. **C. REVEAL SOME BITS:** Alice reveals some of the bits she sent

Bits	0	1	0	1

8. **ANALYSIS:** Bob performs analysis to determine if the message was intercepted by Eve.

A basis	X	Z	X	X	Z	Z	Z	X
B basis	Z	Z	X	Z	X	X	Z	X
Match?	No	Yes	Yes	No	No	No	Yes	Yes

Part A: Check which bits were measured correctly

Bit #	0	1	2	3	4	5	6	7
Bob bits	?	1	0	?	?	?	0	1
Alice bits	0	1	0	1				

Part B: Compare Bob's measurement with Alice's reported bits

If Alice and Bob's bits match

Match?	?	Yes	Yes	?

It's a match! So our key is secure

Bit indices	6	7

Part C: If it's a match, Bob sends the indices of our matched bases to Alice

Key	0	1

Part D: Alice and Bob both construct their secret key using the bits at the matched indices

This key is a **symmetric** key, and Alice and Bob will use the same key for both **encryption** and **decryption**

If Alice and Bob's bits do not match

Eve was here

The quantum channel has been **breached**!

Do not send the key! **Go back to step 1** and use a new quantum channel

Q: How secure is QKD?

A: As secure as it is unlikely Eve chooses the correct basis every time

- There are two possible measurement bases, so Eve has a 50% chance of choosing the correct one for each bit
- If we check N bits, we have a $P(NOT\ detect) = 0.5^N$

Q: How many bits do we need to check to have less than *one in a million* chance of NOT detecting eve?

A: 20 bits

- $P(NOT\ detect) = 10^{-6} = 0.5^N$
- $\rightarrow N = \frac{6(\log(2)+\log(5))}{\log(2)}$

Q: But how do we actually **encrypt** our data? We just have a key but no cipher?

A: Any symmetric key cipher, examples:

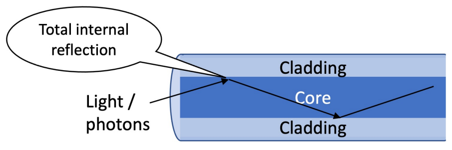
- Advanced encryption standard (AES)
- ChaCha20

Q: How do we send qubits?

A: Send polarization-encoded photons down an optical fiber

- States: $|\nearrow\rangle, |\nwarrow\rangle, |\rightarrow\rangle, |\uparrow\rangle$
- Bases: $\times, +$

optical fiber



guides light long distances with low loss