# Quantum Key Distribution

Julie Butler

# Classical Key Distribution

# No Keys

- Sending an unencrypted message over the internet is possible though not advised

# One Key

- A message can be encrypted with just one key but this key will likely be sent over a public channel
  - Anyone can see the message when it is sent

# Two Keys

- One key is public (sent over a public channel), one key is private (generated by each user)
  - Methods vary
- A message is encrypted and decrypted using both keys
  - Much harder to hack but some problems with the rise of quantum computers

# Review of Hadamard Gates

# How to Create the Superposition States?

$$H|\uparrow\rangle = |+\rangle \quad H|+\rangle = |\uparrow$$
$$H|\downarrow\rangle = |-\rangle \quad H|-\rangle = |\downarrow$$

# How Could This Be Used To Send Binary Data?

- **To Qiskit**
- What if no ones looks at the message?
- What if someone looks at the message?

# It's Not Quite This Simple

- Knowing someone looked at the data is not good enough (someone saw your secrets)
- We need a method where we will know if someone looked at the data BUT also can not get any of the secrets out of the data

# Quantum Key Distribution Process

# Step 1: Person 1 Creates a Random String of Binary Digits

- The person creating the message creates a random string of binary digits, the longer the string the better.
- Person 1 does know what this string is (**this is important**)
- Example: 10010

# Step 2: Person 1 Randomly Changes the State of Each of the Digits

- Person 1 then applies 1 of two filters to each bit, the choice of filter for each bit is random
  - Filter 1 ($F_1$, the Nothing Filter): Nothing if $|\uparrow\rangle$, Nothing if $|\downarrow\rangle$
  - Filter 2 ($F_2$, the Hadamard Filter): $H|\uparrow\rangle = |+\rangle$ if $|\uparrow\rangle$ and $H|\downarrow\rangle = |-\rangle$ if $|\downarrow\rangle$
- Person 1 needs to know the order the filters are applied (**this is important**)
- Example: $F_1F_2F_1F_2F_2$

# Step 3: Person 1 Sends the Encoded Message

- Person 1 now sends the encoded message to Person 2. This can be over a public channel
- Encoded Message: 10010 and $F_1F_2F_1F_2F_2$ results in $|\downarrow + \uparrow - +\rangle$

# Step 4: Person 2 Receives the Message and Measures Each Digit

- Person 2 must choose to apply $F_1$ or $F_2$ to each digit. After applying each filter, Person 2 measures each qubit. Note that each filter is its own inverse
- Example: Assume Person 2 chooses $F_2F_2F_1F_2F_1$ so measures $|- \uparrow\uparrow\downarrow +\rangle$ so there are four possible measurements Person 2 could make

## Step 5: Person 1 and Person 2 Share the Filters the Used on Each Digit

- Over a public channel, Person 1 and Person 2 share the order they applied the filters.
- If they match filter for filter then all is good. If they do not match, then they need to decide if they should restart (if the string is small or there are not many matches) or just keep the digits where they had the same filters (if the string is large or there are many matches)
- Person 1 and Person 2 are now reasonably confident they have the same binary string, this is now the secret key.
- Example: Person 1 shares $F_1F_2F_1F_2F_2$ and Person 2 shares $F_2F_2F_1F_2F_1$ so maybe they decide to just keep the middle entries so 001 is now the secret string

# Step 6: Person 1 and Person 2 Share a Fixed Amount of the Secret Keys

- Finally, Person 1 and Person 2 share some fixed amount of the secret key, if those digits match then they can be reasonably sure they do have the same secret key. The shared digits are removed from the secret key.
- Example: Person 1 and Person 2 share the first digit. Both share 0 and are satisfied. Their secret string is now 01.

# Drawbacks of the Algorithm

# What if there is a Hacker?