

How it works



Hedy Lamarr 1944

- Hedy Lamarr was a trendy Austrian-American Film actress and inductee into the National Inventor's Hall of Fame for her work on radio **frequency-hopping spread spectrum** radio-guided torpedoes used in WW II. Her invention received [U.S. Patent 2,292,387](#). This method of **obfuscating** radio messages entails transmitting small parts of a message on different radio channels. The transmitter and receiver agree on a list of channels that will be used to send and receive each part of the message. When the first bit of the message is transmitted on the first channel in the list, the transmitter and receiver switch their radios to the next channel on the list. Each bit of the message is sent over a different radio channel until the entire message has been sent. The message cannot be easily intercepted if the list of channels is kept secret from a hacker. This activity generates the channel list from a **private key** shared between the sender and receiver.

What will you do?

- Practice making a channel Python list:
 - Each group member should open '*pract_2.py*' in the editor and run the program. Enter a short key when prompted.
 - Run the program again with a longer key.
 - What is the difference between the outputs of the two programs? Does a short key or a long key generate more channels? Which channel list would be more secure?
 - Look at the code for '*mk_key_2.py*'. Can you figure out how the code works? It utilizes the **ASCII** number for each character in the key and ensures the channel number is never greater than the 84 possible micro:bit channels.
- Texting a message using frequency hopping spread spectrum:
 - Ensure all group members use the same assigned group number.
 - The **receiver**:
 - Advance to '*recv_2.py*', change the group to their assigned number, and run the program *before* the sender has run theirs.
 - The **sender**
 - Advance to the '*send_2.py*', change the message string and group to their assigned number, and then run your program *after* the receiver and hacker have started theirs.
 - The **hacker**
 - Advance to '*hack_2.py*', change the group to their assigned number, and run the program *before* the sender has run theirs.
 - After your team runs the activity, the sender should change the *message* and *key* and share the key only with the receiver. Don't tell the hacker the new key; *keep it private*! Can the hacker read your message in cleartext as they did in the 'All Clear' activity? Can you look at the code and explain the result?

Code it

Sender role

```
EDITOR: SEND_2
PROGRAM LINE 0001
from microbit import *
from mb_radio import *
from utils_2 import *
radio.on()
disp_clr()
msg="Hedy Lamarr was a smart woman!"
key = input("key: ")
ch_list=make_ch_list(key)
# Change group to assigned number
gp = 1
```

Receiver role

```
EDITOR: RECV_2
PROGRAM LINE 0001
from microbit import *
from mb_radio import *
from utils_2 import *
radio.on()
disp_clr()
key = input("key: ")
channels=make_ch_list(key)
# Change group to assigned number
gp = 1
# the stop_chr marks end of message
```

Hacker role

```
EDITOR: HACK_2
PROGRAM LINE 0001
from microbit import *
from mb_radio import *
from utils_2 import *
radio.on()
disp_clr()
key = input("key: ")
channels=make_ch_list(key)
# Change group to assigned number
gp = 1
# the stop_chr marks end of message
```

Go further

- Rerun the activity in a different team role.
- Repeat the activity using different keys and messages.
- Try to discover how many channels are on the list for a given key.

Check your understanding

- For two radios to communicate, they must be on the same channel and group.
- A transmitting program can switch the channel after sending each character.
- A receiving program must know the channels the transmitter will use to send the message beforehand.
- Using a frequency channel hopping algorithm, like the one in this activity, can make a “man-in-the-middle” attack more difficult because the transmitting channel changes in a way the hacker may not know.

Help

- Check that everyone on the team is using their assigned group number.
- Ensure the receiver and hacker run their programs and wait before the sender transmits the message.
- Ensure the sender and receiver use the same key.
- Ensure the hacker does know the key.

Files

- Transfer the activity files below to your calculator using the TI Connect CE Software. The link to download is [here](#). The best practice is to load all files for this cybersecurity activity and then delete them before loading the next set of activity files. This helps keep your calculator organized.

Name	Description
pract_2.py	Practice making a list of channels base on a private key.
send_2.py	Sends text message to receiver using frequency hopping.
recv_2.py	Receives text message from sender using frequency hopping.
hack_2.py	Receives obfuscated text message from sender.
mk_key_2.py	A utility used to generate channel list used in the other four programs.