Files Provided

- `challenge_noisy.wav` → contains XOR-encrypted flag hidden as Base64 text.
- `mystery_audio.wav` → contains the XOR **key**, embedded as reversed or distorted audio.

Step 1: Extract Hidden Base64 from `challenge_noisy.wav`

Method A (Easiest - With Notepad):

1. Right-click `challenge_noisy.wav` → **Open with** → **Notepad**.
2. Scroll or **Ctrl+F** and search for any random-looking string that:

- Has characters like A-Z, a-z, 0–9, `+`, `/`, `=`
- Is roughly **~40+ characters** in one line 3.

Example:
`d8akvCwOP0jdsgJbMPBhkRiyBiBwd1cBO565Bdw6qiOK`

4. Copy that string to a new file called `encoded.txt`.

Method B (Using Command Prompt):

1. Open **Command Prompt** in the folder where your `.wav` is.
2. Run:

```
cmd findstr /R "[A-Za-z0-9+/=]\{30,\}" challenge_noisy.wav
```

This will show Base64-looking lines.

Step 2: Decode Base64 to Encrypted Bytes

Option 1: Use `certutil` (Built-in on Windows)

1. Create a text file called `encoded.txt` with the extracted Base64 string.
2. Run in CMD:
   `cmd certutil -decode encoded.txt output.bin`
3. This creates `output.bin`, which contains XOR-encrypted flag bytes.

Step 3: Extract the XOR Key from `mystery_audio.wav`

Tools: **Audacity** (or **Sonic Visualiser**) – Free but optional

Method A: Reverse Audio to Hear Key

1. Open `mystery_audio.wav` in **Audacity**.
2. Select the whole track (Ctrl+A).
3. Go to **Effect** → **Reverse**.
4. Press **Play**. You'll hear a human voice speak the XOR key (e.g., `key12345`).
5. Write down exactly what is spoken — case sensitive!

> Why reversed? Because the key was **embedded backwards** to prevent easy detection.

Step 4: XOR Decrypt the Output Using the Extracted Key

You now have:

- Encrypted file: `output.bin`
- Key: e.g., `key12345`

Method A: Use CyberChef (Browser-based)

1. Go to CyberChef.
2. Load `output.bin` by dragging it in.
3. On the left pane:

- Add **"XOR"** operation. * Set the key as what you heard from audio (e.g., `key12345`).

4. Click "Bake" — the output will show the decrypted flag.
