Step-by-Step Guide to Using the GPG Telegram Bot

Prerequisites

- The bot is running on your computer (`python3 sendMessage.py`)
- You have a Telegram account and the app installed
- You have set the correct BOT_TOKEN in the script

Setting Up Your Bot

1. Find Your Bot on Telegram

- Open Telegram
- Search for your bot using the bot username you created with BotFather
- Start a chat with your bot

2. Create a GPG Key

- Send the command: `/createkey YourName your@email.com`
- The bot will generate a new GPG key pair
- The bot will send you the public key as a file
- Save this file if you want to share it with others

Sending Encrypted Messages

3. Encrypt and Send a Message

- To encrypt a message, send: `/encrypt [fingerprint] Your secret message`
- The `[fingerprint]` is the GPG key fingerprint shown when you created the key
- Example: `/encrypt A1B2C3D4E5F6G7H8I9J0 This is a secret message`
- The bot will encrypt the message and forward it to the recipient

4. Import Someone Else's Key

- If someone sends you their public key, you can import it
- Attach the key file to a message
- Send the command: `/importkey`
- The bot will import the key and confirm success

Receiving Encrypted Messages

5. Receive and Decrypt Messages

- When you receive an encrypted message, it will start with "🔒 Encrypted message:"
- To decrypt it, reply to that message with the command: `/decrypt`
- The bot will decrypt the message using your private key
- The decrypted message will be shown prefixed with "🔓 Decrypted message:"

Managing Your Keys

6. List Available Keys

- To see all available public keys, send: `/listkeys`
- The bot will show all key fingerprints and associated names/emails

Important Notes

1. The bot forwards all your regular (non-command) messages to the recipient specified in the code.
2. Keys created with `/createkey` expire after 1 day.
3. The bot must be running on your computer for these commands to work.
4. The recipient must have access to the private key to decrypt messages.
5. To stop the bot, press Ctrl+C in the terminal where it's running.

Example Workflow

1. You create a key: `/createkey Alice alice@example.com`
2. The bot sends you your public key file
3. You share this file with Bob
4. Bob imports your key using `/importkey`
5. Bob encrypts a message to you: `/encrypt [your_fingerprint] Hello Alice, this is secret`
6. The bot forwards the encrypted message to you
7. You decrypt the message by replying with `/decrypt`
8. You see the original message from Bob