GPG Telegram Bot Command Reference Card

Key Management Commands

Command	Syntax	Description	Example
`/createkey`	`/createkey <name> <email>`	Creates a new GPG key pair that expires in 1 day. Returns the public key as a file.	`/createkey Alice alice@example.com`
`/importkey`	`/importkey` (with attached key file)	Imports someone else's public key. You must attach the key file to this message.	Send `/importkey` with a .asc file attached
`/listkeys`	`/listkeys`	Shows all public keys currently available to the bot, with their fingerprints and expiration dates.	`/listkeys`

Messaging Commands

Command	Syntax	Description	Example
`/encrypt`	`/encrypt <fingerprint> <message>`	Encrypts your message using the specified key fingerprint and sends it to the recipient.	`/encrypt A1B2C3D4E5F6 This is a secret message`
`/decrypt`	`/decrypt` (as reply) or `/decrypt <encrypted_text>`	Decrypts an encrypted message. Works best when used as a reply to an encrypted message.	Reply to an encrypted message with `/decrypt`

General Behavior

- **Regular Messages**: Any message you send that is not a command will be forwarded to the predefined recipient unencrypted.
- **Encrypted Messages**: Appear with 🔒 emoji prefix.
- **Decrypted Messages**: Appear with 🔓 emoji prefix.
- **Key Expiration**: All keys created with `/createkey` expire after 1 day.

Workflow Example

1. **Create key**: `/createkey YourName your@email.com`
2. **Share the key file** with the person who wants to send you encrypted messages
3. **Import their key**: Use `/importkey` with their key file attached
4. **Send encrypted message**: `/encrypt THEIR_FINGERPRINT Your secret message`
5. **Decrypt received message**: Reply to an encrypted message with `/decrypt`

Important Notes

- The bot must be running on your computer for these commands to work
- All encrypted messages are forwarded to the recipient defined in the script
- Keys are stored in `~/.gnupg` on your Linux system