croc Simple file transfer

START HERE

`croc send <file>`
Send, get a code phrase

`croc --yes <code>`
Accept without prompting

`croc <code>`
Receive with the code

`croc --out <dir> <code>`
Receive into a folder

`croc send <dir>`
Send a whole folder

`croc send --text "<msg>"`
Send text or a URL

`croc send --code <phrase>`
Pick your own code

`CROC_SECRET=<code> croc`
Receive, code not in ps

Sender croc send → code phrase → **Relay** matches both sides → encrypted → **Receiver** croc <code>

CONCEPTS

The code phrase is the shared secret: it keys the end-to-end encryption, so share it over a different channel than the transfer.

Both ends must run croc. Direct local transfer is tried first; the relay only carries ciphertext when no direct path exists.

Typing the code on the command line shows it in the process list; use CROC_SECRET on shared machines.

SEND

`croc send <file>`
Send a file

`croc send <file> <dir>`
Send several files and folders

`croc send --text "<msg>"`
Send text or a URL

`croc send --code <phrase>`
Use a custom code (6+ chars)

`cat <file> | croc send`
Send from stdin

`croc send --zip <dir>`
Zip a folder before sending

`croc send --qr <file>`
Show the code as a QR code

SEND (cont.)

`croc send --exclude "<a>,<b>"`
Skip paths containing these strings

`croc send --exclude-file "<p1>,<p2>"`
Skip exact relative paths

`croc send --git <dir>`
Respect .gitignore

`croc send --no-local <file>`
Skip the local relay, use remote

`croc send --hash imohash <dir>`
Faster hash for huge files

`croc send --transfers <n>`
Number of transfer ports

`croc send --port <n>`
Base relay port (default 9009)

RECEIVE

`croc <code>`
Receive with a code phrase

`CROC_SECRET=<code> croc`
Receive without the code in ps

`croc --out <dir> <code>`
Save into a directory

`croc --yes <code>`
Auto-accept the transfer

`croc --overwrite --yes <code>`
Overwrite existing files silently

RECEIVE (cont.)

`croc --rename <code>`
Keep both: save as name (1)

`croc --yes --stdout <code> > <file>`
Write received data to stdout

`croc --ignore-stdin`
Do not read piped stdin

`croc --classic`
Classic mode: plain prompt for the code

OPTIONS

`croc --ask send <file>`
Confirm on both sides

`croc --no-compress send <file>`
Skip compression (media, archives)

`croc --throttleUpload 500k send <file>`
Cap upload speed

`croc --local send <file>`
Local network only, no relay

`croc --curve p521 send <file>`
Pick the PAKE curve

`croc --quiet send <file>`
No output

`croc --disable-clipboard send <file>`
Do not copy the code

OPTIONS (cont.)

`croc --extended-clipboard send <file>`
Copy full receive command

`croc --remember send <file>`
Save these options for next time

`croc --debug <code>`
Verbose debug output

RELAY AND NETWORK

`croc relay`
Run your own relay (ports 9009-9013)

`croc relay --ports 1111,1112`
Choose relay ports (2 or more)

`croc --relay <host:9009> send <file>`
Use a specific relay

`croc --relay <host:9009> --pass <pw> <code>`
Receive via private relay

`CROC_RELAY=<host:9009> CROC_PASS=<pw> croc`
Relay settings from env

`croc --ip <host:9009> <code>`
Connect to the sender directly

RELAY AND NETWORK (cont.)

`croc --socks5 127.0.0.1:9050 send <file>`
Go through a SOCKS5 proxy

`croc --connect <host:port> send <file>`
Go through an HTTP proxy

`croc send --transport relay <file>`
Force relay transport

STORE AND EXTRAS

`croc store <file>`
Upload encrypted for later pickup

`croc store --downloads 3 --expiration 3d <file>`
Limit downloads and lifetime

`croc ssh`
Host a shared terminal session

`CROC_SECRET=<invite> croc ssh`
Join a shared terminal

`croc update --check`
Look for a newer release

`croc update`
Install the latest release

`croc --revoke <id>`
Delete a stored transfer

`croc --version`
Show installed version

STORE AND EXTRAS (cont.)

`croc --help`
List commands and flags

`croc send --help`
Flags for send

croc update only works for the script install (curl https://getcroc.com | bash); with a package manager, update through it.

RECIPES

Send a project without junk

```
croc send --git --zip <project-dir>
# receiver:
croc <code>
```

Pipe a stream over croc

```
tar czf - <dir> | croc send
# receiver:
croc --yes --stdout <code> | tar xzf -
```

Private relay between your machines

```
# on the server
CROC_PASS=<pw> croc relay
# on each client
export CROC_RELAY=<server>:9009
CROC_PASS=<pw>
croc send <file>
```

Unattended receive

```
CROC_SECRET=<code> croc --yes --overwrite --out <dir>
```