



Cryptopuck

Encrypt your flash drives on the fly

About me

A photograph of a man with short brown hair, wearing a dark jacket over a collared shirt, smiling. He is standing in front of a large, ornate building with a prominent clock tower and a green dome. The scene is outdoors with trees and a clear blue sky.

Dimitrios Platis

- Grew up in Rodos, Greece
- Software Engineer @ Cellink, Gothenburg
- Course responsible @ DIT112, GU
- Interests:
 - Embedded systems
 - Software Architecture
 - API Design
 - Open source software & hardware
 - Robots, Portable gadgets, IoT
 - 3D printing
- Blog: <https://platis.solutions/blog>

What is it?

-
- Portable device to encrypt your removable media
 - Easy to use
 - Plug & play
 - Secure*

* To the extent a non-audited project using hobby-grade components can be

How to use

1. Turn on
2. Wait until LED is on
3. Plug in flash-drive
4. Wait until LED stops blinking
5. ??????
6. Profit (your drive is encrypted)



Why did I make it?

- "Dumbed down" encryption for everyone
- Should be cheap
- OS-agnostic
- Should be able to carry in pocket
- Inspired by encryptable SD card from Zifra
 - Discovered them during [Foss-North 2017](#)
- Wanted to learn more about practical cryptography

What is it made of?

- Raspberry Pi Zero
- 3D printed case
- DC step-up module (3.3V to 5V)
- Micro-USB OTG cable
- 1400mAh Li-Po battery
- On/Off switch
- 5mm LED
- 220 Ω resistor



How does it work?

Crypto 101: Symmetric encryption

-
- Overview
 - Same secret (e.g. mnemonic password) used to both encrypt and decrypt information
 - Must be securely stored
 - Advantages
 - Simple, effective & fast
 - Disadvantages
 - Has to be shared securely for a different party to decrypt
 - One cannot (plausibly) deny the ability to decrypt data, e.g.: if someone puts a gun on your head and you know the secret, you may be coerced to give it up)

How does it work?

Crypto 101: Asymmetric encryption

-
- Overview
 - Two keys, one for decryption ("secret key") and one for encryption ("public key")
 - Only the "secret key" must be securely stored
 - Advantages
 - No need to share secrets in order to decrypt information
 - Can plausibly deny the ability to decrypt information since the secret key does not have to be in your possession (e.g. can be controlled by your employer)
 - Disadvantages
 - Slow & inefficient especially for (large) files

Best of all worst of none

- Master Ken

-
1. Encrypt fast without having to share secrets
 2. Produce a random secret
 3. Symmetrically encrypt the data
 4. Asymmetrically encrypt the secret
 5. Share the asymmetrically encrypted key and symmetrically encrypted data (possibly over unsecured channels)
 6. Decrypt the secret
 7. Use decrypted secret to decrypt data

How does it work?

- Cryptopuck encryption

1. Automount flash drive using [udiskie](#)
2. Get a callback when new drive is mounted via [inotify](#)
3. Generate random **32-bit** secret using `/dev/hwrng` which will be used to **AES-256** encrypt the drive
4. Use public key on flash drive to RSA encrypt (according to **PKCS#1 OAEP**) the secret and save it on the drive
5. Recursively encrypt (**AES CBC mode**) all files individually
 - a. Avoid zipping everything and then encrypting due to performance limitations
6. Hash (**sha512**) the filenames and encrypt the file that contains the map between the salted hashes and the filenames

How does it work?

- Cryptopuck decryption

1. Use safely stored secret key to RSA decrypt the secret found on the flash drive
2. Use the decrypted secret to AES-256 decrypt the map containing the file structure
3. Decrypt the rest of the files while restoring the file structure

Use cases

- Reporter or spy in a war zone who just collected sensitive data
 - Private key held by the employer or agency, therefore unable to decrypt data
- Developer that needs to send a transfer or ship a physical disk with proprietary information to a remote site or a customer
 - Private key held by the receiver. If someone needs to transfer data to Bob, then just connect your disk to Bob's Cryptopuck.
- Going through TSA airport control without wanting to disclose personal information
- Any other situation where you need **discretely** to encrypt **removable** media **on the fly**

Disclaimer

This is a hobby project!

Do **not** use it if your
<life/freedom/industrial secrets>
depends on it

Ideas for improvement

- Use a stronger computer or FPGA instead of RPi Zero
- Design better case
- Design PCB to mount all the components
- Create a Yocto image instead of manually configuring Raspbian

FAQ

- Why not encrypt the entire volume?
 - Raspberry Pi Zero is too slow for that
 - Encrypted volume (e.g. LUKS) would not be read out of the box across operating systems
- Why don't you ZIP the files before encrypting them so not to expose metadata?
 - Raspberry Pi Zero is too slow for that
- Traces may remain despite removing the clear-text files
 - Accept it as a trade-off, since overwriting unused space with random data would slow things down a lot and would not necessarily guarantee complete removal of traces due to wear leveling of flash devices
- What happens if they confiscate my Cryptopuck?
 - Nothing much. They will still not be able to decrypt your files. However, you probably should not use it again as they might have tampered with it.

Questions?

GitHub repo:

<https://github.com/platisd/cryptopuck>

Demo video

