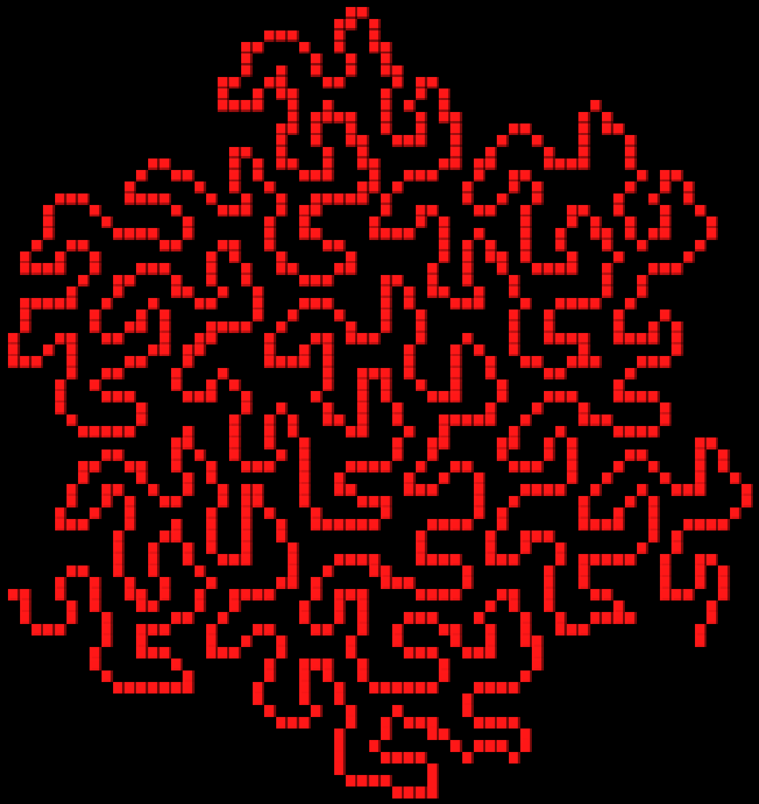

Pavel Polach
HW, Firmware: i_a@rmxwallet.org

Robin Nemeth
PC-side: pocin@rmxwallet.org

- “Hello” slide
- Why we are doing this

- 5 Project challenges

- Project setup
- Project goals

- Actual state
- Questions

Why we are doing this?

- Deepfake & social media => collapse of trust → signing, encrypting
- Monero support (10 Tx/s versus XX Tx/s?)

“Challenges”

(1) Keep the project going

(2) HW messenger

(3) AES file encryption

(4) Monero implementation

(5) HW Security

(1) Keep it going

- Open hardware is challenging itself
- Thinking about next step(s)
- Avoiding burn-outs
- Contributors, motivations
- Funding

(2) HW messenger

- RMX to RMX encrypted
- XMPP as a transport protocol
- Any XMPP server..?
- 256B long messages, (user experience close to SMS)
- Messages are encrypted, then sent as a plaintext
- Each message is symmetrically encrypted with a one-time key

(2) HW messenger

- Encryption variables:

  1. Get one time random r [32B]
  2. Get recipient’s pubkey P (query XMPP server)

- Creating encryption key

  1. Creating one time encryption key $K = rP$ [32B]
  2. $X = rG$ (ed25519) – will be added to encrypted payload

- Encryption

  symmetric AES encryption, CBC MODE,

  Randomization vector SHA3(K)

- Payload

  [encrypted string 256B || X]

- Receiving

  $K' = privkeyX$

rmxwallet.org

(2) HW messenger

- Sending over XMPP

  Passing login and password to PC-side

  PC-side opens a session with XMPP server

  Takes care about logging in, keys, sending, receiving

- Improvements?

  Logging in by signing a challenge

  Running own XMPP server (+, - …)

  ..and user experience of course:)

(3) AES file encryption

- AES encryption implemented in CEC1702, supported modes:

```
#define AES_MODE_ECB        (0ul)
#define AES_MODE_CBC        (1ul)
#define AES_MODE_CTR        (2ul)
#define AES_MODE_CFB        (3ul)
#define AES_MODE_OFB        (4ul)
#define AES_MODE_CCM        (5ul)
#define AES_MODE_GCM        (6ul)
#define AES_MODE_XTS        (7ul)
#define AES_MODE_CMAC       (8ul)
```

- Key length 128b, 192b, 256b
- Message length 2048B, one operation around 400us
- Possibility to encrypt/decrypt around 0.4MB/s

(3) AES file encryption

Not implemented so far:)

(1) Brainstorming session - sketch the bigger picture and functionalities, define protobuf message

(2) Program it

(3) ???

(4) profit

(4) Monero implementation

- Monero has encrypted blockchain
- viewkey, spendkey — functions segregation

SCANNING:

- Tx: Output's public key P, Tx's public key R
- P - H(aR)G
- 2x Ed25519 Elliptic curve multiplication, one SHA-3 hash, one point substraction for each Tx's output
- Check if the result == with public spend key

(4) Monero implementation

- Public scanning – offloading viewkey to PC
- If match, unmasking

- Private scanning
- Too heavy for arm cortex m3/4
- Need for faster ed25519 multiplications

(4) Monero implementation

- Microchip CEC1702
- Cortex M4F + hardware accelerator
- Scanning one output in 4ms (100 Tx/s*)
- 8 Txs in a block every 2 minutes
- One day scanned in 46s, one year in 5h

*one Tx contains two outputs

rmxwallet.org

(5) HW security

- Secure boot feature
- Only signed images
- Cortex m4 -Fault injections? Glitches?
- Passphrase?
- Source of random
- Secure element versus encrypted secret

Our setup

- Free time project
- Lean approach
- “No lab”
- “As open as we can”
- No stressing out:)

Actual state

- Proven PCB ready to “mass” production
- GUI, accelerated crypto functions, “ready as a platform”
- PC-side in progress, communicating stuff
- Secure messenger ready to debug
- AES file encryption tested
- Monero soon(TM)

Future?

- Key manager

Project goals

- Keep it going
- Contribute to ecosystem
- Monero HW wallet with private scanning
- AES file encryption token
- Secure HW messenger (multisig)
- 2FA, Key manager
- Source of entropy
- Vision: “ARDUINO” like device

rmxwallet.org

rmxwallet.org

THANK YOU

Pavel Polach
HW, Firmware: i_a@rmxwallet.org

Robin Nemeth
PC-side: pocin@rmxwallet.org