# TZ - Password Manager

THARUN KUMAR JAYAPRAKASH(TJ2557) HUGO MATEOUSK(HM2953)

Estimated Total Hours: 12



## Plans (Past Week)

- Setup OP-TEE Environment on Raspberry Pi
- Architecture detailed slides, threat model
- Secure API definition
- Basic "frontend" (unsecured application)
- Basic "backend" (secured application)





### What Actually Happened

#### Work completed:

- Encryption/decryption and key management architecture
- Modularized host application, advanced TEE internal API functions used for TA

#### **Major challenges+roadblocks:**

- Faced Issues running OP-TEE on Raspberry Pi 4b
- Header file management and file structure when compiling as a part of the TEE image (and QEMU)

#### **Attribution:**

 Official OP-TEE support resources exist only for Raspberry Pi 3





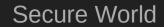
### Plans (Next Week)

- Continuing with attempts to make it work on RPi 4
- Key derivation and secure storage tests for TA
- Host application basic "UI" and application logic



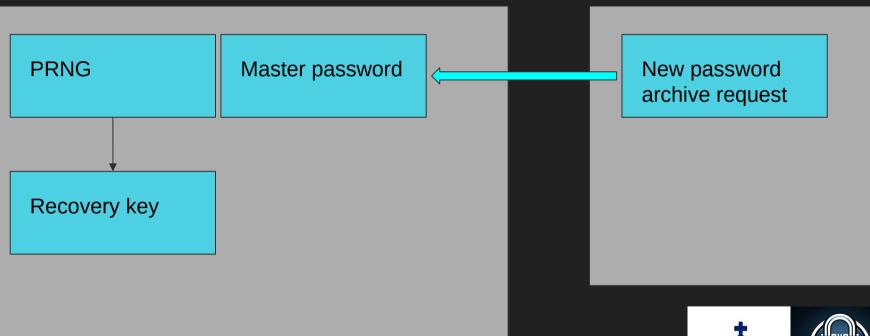


# Architecture Normal World New password archive request





Normal World

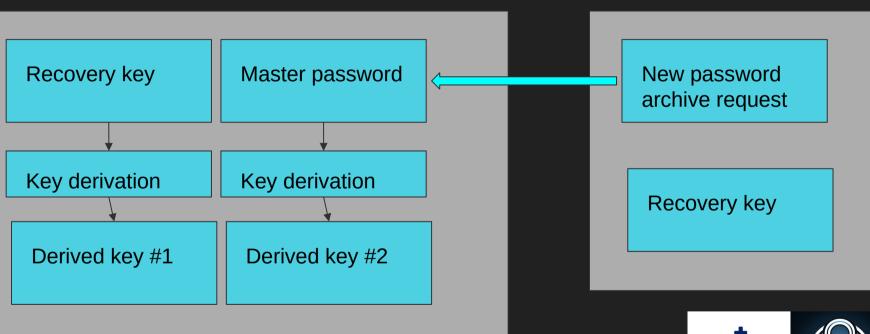






# Architecture Normal World **PRNG** Master password New password archive request Recovery key

# Architecture Normal World Master password New password archive request Recovery key Recovery key

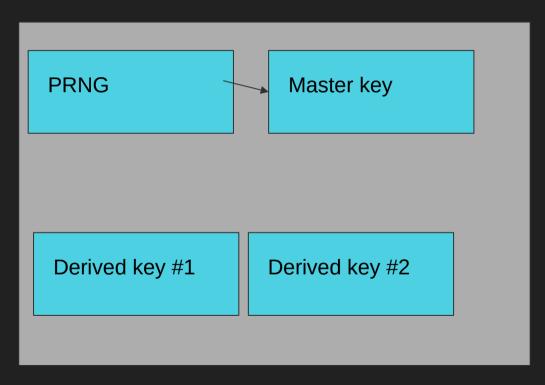




#### Normal World







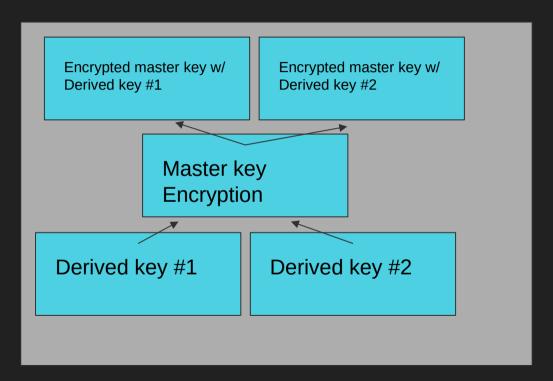
#### Normal World

New password archive request

Recovery key







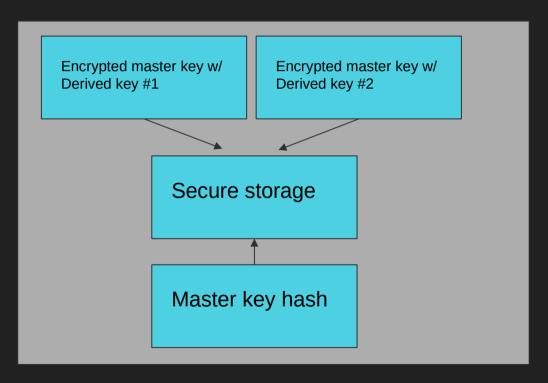
#### Normal World

New password archive request

Recovery key







#### Normal World

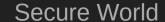
New password archive request

Recovery key





# Architecture Normal World Master password Access to + hash of the request archive request Master key hash Derived key #2





# Architecture Normal World Master password Access to + hash of the request archive request

Encrypted master key w/

Derived key #2





Master key hash

Normal World

Hash of the request

Master password to derive key

Master key hash

Encrypted master key w/ Derived key #2 Access to archive request





Normal World

Hash of the request

Derived key #2

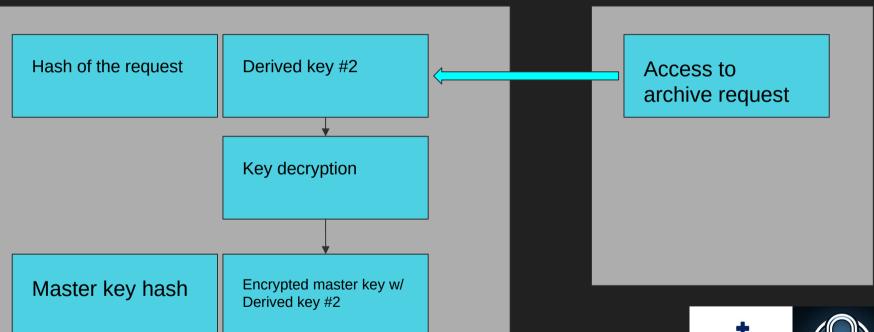
Master key hash

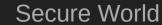
Encrypted master key w/ Derived key #2 Access to archive request





# Normal World

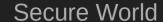








### Normal World Hash of the request Access to archive request Comparison of Decrypted "master hashes key" Master key hash Encrypted master key w/ Derived key #2





Hash of the request Access to archive request Derived key for the -**Decrypted master** request key Encrypted master key w/ Master key hash Derived key #2



Normal World



Normal World

Hash of the request

Derived key for the request

Master key hash

Encrypted master key w/ Derived key #2

Decrypt/encrypt entry

Access to archive request





#### Normal World

Access to archive request

Decrypted/encrypted Response

Master key hash

Encrypted master key w/ Derived key #2

entry





### Summary/Overall Progress

- Encryption/decryption and key management architecture prepared
- Modularized host application
- Advanced internal TEE API functions tested

