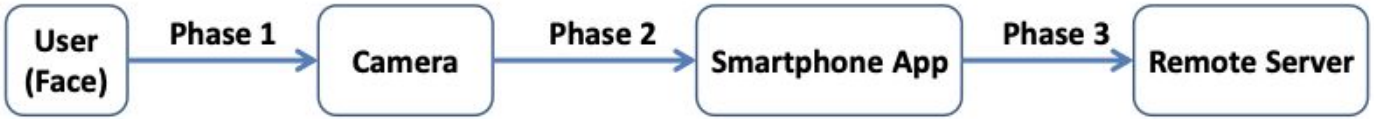
TrustZone Face Recognition

Shivam Shekhar (ss6960)
Ritwik Goel (rg3546)

Phases of Data Flow



Phase 1

- Phase 1 is to capture the image from the front camera.
- It is vulnerable to 2D attack.
- Countermeasure - We capture the photo and collect the accelerometer data in TrustZone secure world.

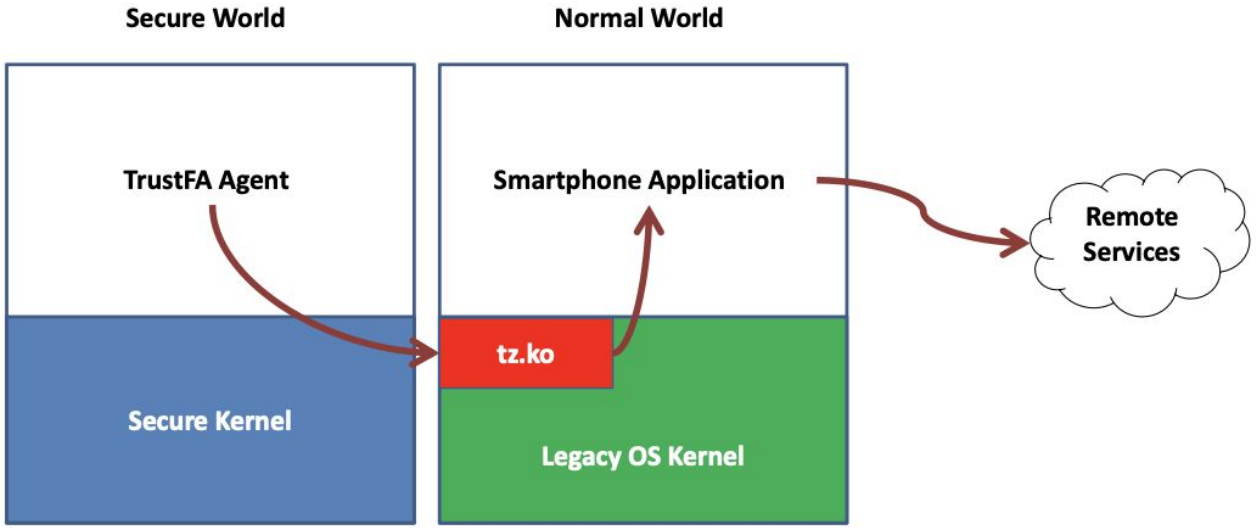
Phase 2

- In Phase 2, we retrieve the the image by smartphone application via the legacy OS.
- The untrusted legacy OS would tamper the photo captured by the camera, or replace the captured photo/video with pre-captured ones.
- Countermeasure: We leverage the ARM TrustZone technology to ensure the trust of data from camera/accelerometer. Both photo and accelerations are collected in TrustZone secure world

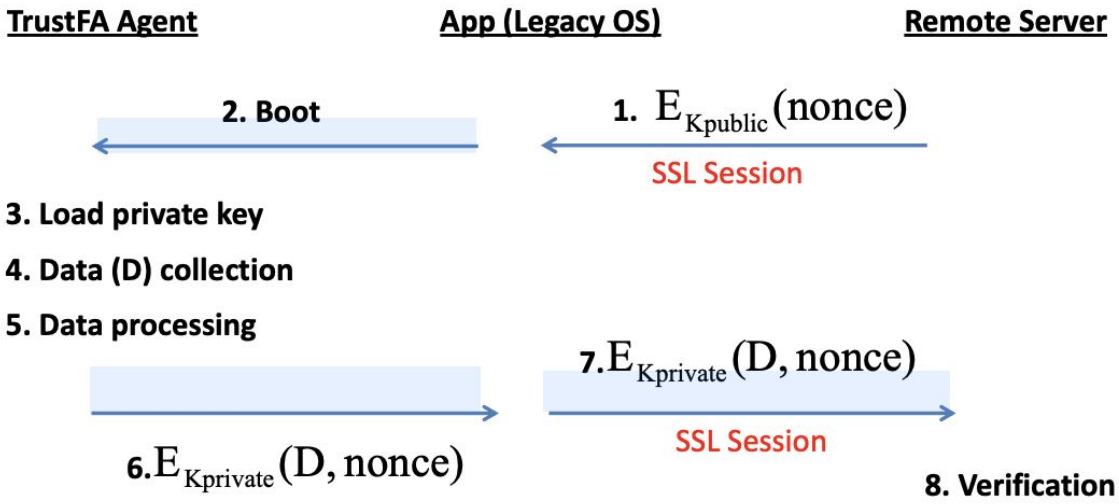
Phase 3

- In Phase 3, the photo (or features extracted from photos) is sent to the remote service to authenticate the user.
- This phase is secured using SSL.

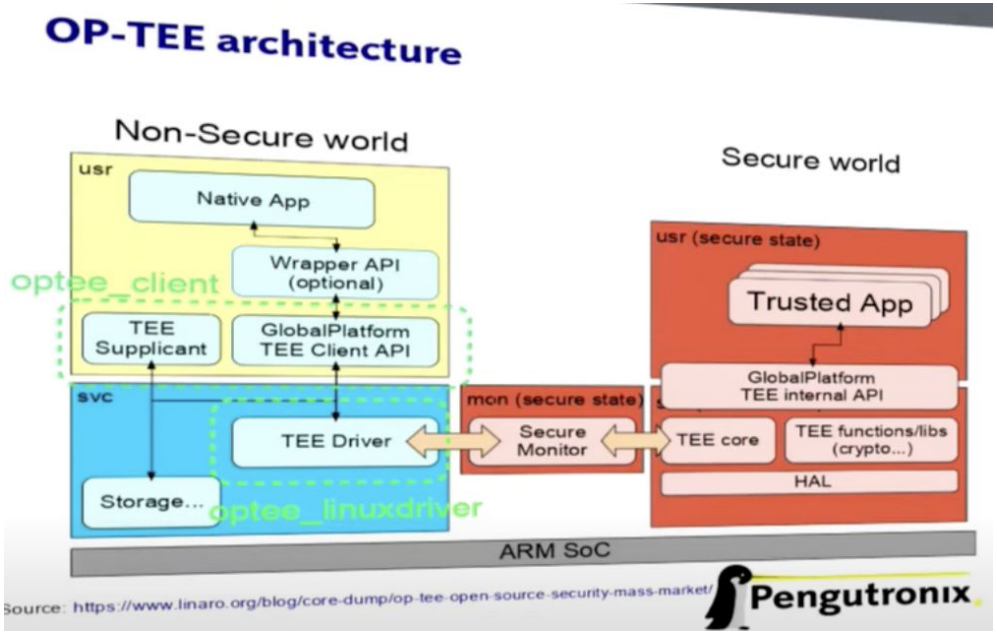
Design


Secure World
Normal World
TrustFA Agent
Smartphone Application
Remote Services
tz.ko
Secure Kernel
Legacy OS Kernel

Workflow


TrustFA Agent
App (Legacy OS)
Remote Server
2. Boot
1. $E_{Kpublic}(nonce)$
SSL Session
3. Load private key
4. Data (D) collection
5. Data processing
7. $E_{Kprivate}(D, nonce)$
SSL Session
6. $E_{Kprivate}(D, nonce)$
8. Verification

OP-TEE Architecture



Some Questions

Trusted apps should be kept as small as possible to lower the probability of introduction of security flaws and also to execute and finish fast.-> so how faceID

How would we use the Camera module directly without involving the Non-Secure World? Execution in the normal world jumps to the secure world by explicitly issuing the Secure Monitor Call (SMC) instruction. (Answer Maybe)

References

1. https://www.donglizhang.org/trustfa.pdf
2. https://source.android.com/docs/security/features/trusty
3. https://optee.readthedocs.io/en/latest/general/about.html
4. https://optee.readthedocs.io/en/latest/architecture/globalplatform_api.html