0xDACC

2024 Design Document

Secure MISC

Proposed Attest Changes

Store attestation PIN as a hash with enough rounds that it takes approximately 2 seconds.

- Limits brute force attempts
- Makes PIN unable to be extracted from flash

Store attestation data encrypted with symmetric key as 1 round less of attestation pin hash

- Also limits brute force and makes PIN unreadable from flash

Altogether these changes will help meet SR3 and SR4

Proposed Replace Changes

Store replacement token as a hash

- Makes token unable to be extracted from flash

Verify component authenticity

1. Store an asymmetric public key in flash
2. Generate a random number using onboard TRNG
3. Ask new component to sign random number
4. Verify using onboard public key

Altogether these changes will help meet SR1 and SR2

Proposed Boot Changes

Verify integrity of all 3 boards

- Store public keys A and D on AP
- Store public key B on Component1
- Store public key C on Component2

1. AP generates a random number and asks Component1 to sign
2. AP verifies signature
3. Component1 generates a random number and asks AP to sign
4. Component1 verifies signature
5. Component2 generates a random number and asks AP to sign
6. Component2 verifies signature
7. AP generates a random number and asks Component2 to sign
8. AP verifies signature

If any signatures are invalid, stop immediately and shut down.

Altogether these changes will help meet SR1 and SR2.

Proposed Secure TX & RX Changes

Public KEX

- Generate private key using RNG
- Create an encrypted channel even though unnecessary.
- Confidentiality will be provided to make RE'ing just a tiny bit harder
- Encrypt packets with negotiated key
- Negotiate HMAC key over new channel
- Append HMAC to all packets before encrypting

Altogether these changes will help meet SR5.

Other

Secure DAPLink firmware for RISC-V chip

- Only execute signed code
- Disable the DAPLINK flashing utility
- Disable code debugging

All of the above objectives are futile if the attacker can simply modify the flash or just set a breakpoint where the validation happens. By not allowing the chip to be debugged (easily) and only allowing signed code to be run, security becomes a lot more reasonable.