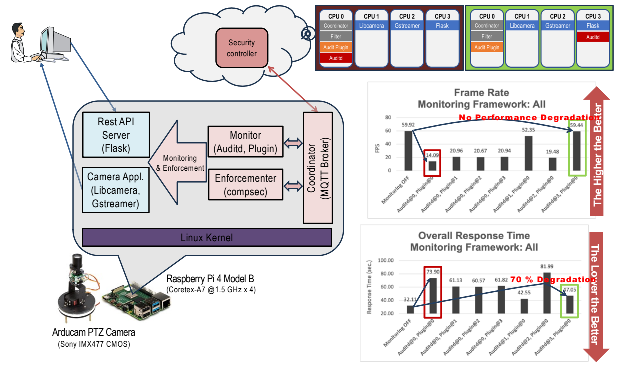
ZT-IoT Research Topics

https://zt-iot.nii.ac.jp/en/

Monitoring & Policy Enforcement

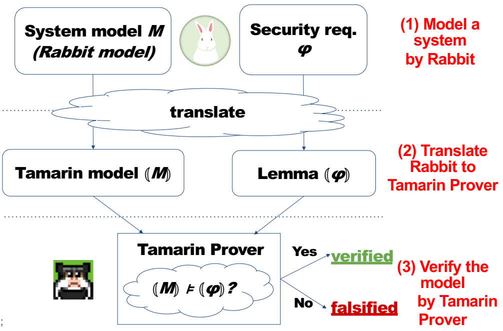


- A monitoring and enforcement framework* based on Linux Audit and MQTT for IoT devices is being designed and implemented.
- As shown in the above figure, a prototype system carries out a real-time execution environment for application and system security monitoring using optimum process-to-core mapping. Many existing works using Linux Audit pointed out the undesirable runtime overhead to the system.

* A Linux Audit and MQTT-based Security Monitoring Framework," IEEE COMPSAC 2023, June 2023.

Formal Verification

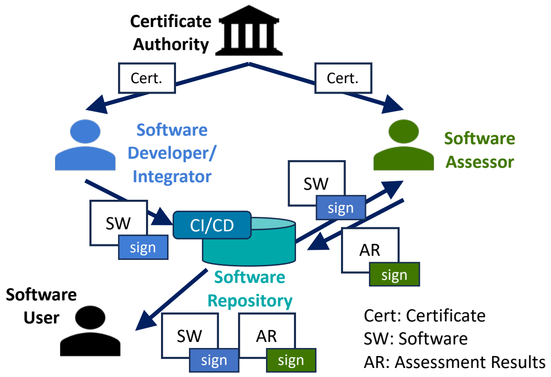
```
procedure CameraServer {
  ServerREE :
    for i in range(3) {
      let p = recv(ch);
      let res = verify(p.snd, p.fst, pkey);
      if (res) {
        skip @ !ImgRecvValid(p.fst);
      } else {
        skip @ !ImgRecvInvalid(p.fst);
      }
    }
}
procedure CameraClient {
  ClientREE :
    let image_fd = open(campath);
    copy skeypath to ClientTEE as t_skeypath;
  ClientTEE :
    let skey0 = read_from_path(t_skeypath);
    let skey = dec(skey0, fek);
  ClientREE :
    for i in range(3) {
      let image = read(image_fd);
      copy image to ClientTEE as t_image;
      ClientTEE :
        let t_sig = sign(t_image, skey);
        copy t_sig to ClientREE as sig;
      ClientREE :
        send(ch, (image, sig)) @ !ImgSend(image);
    }
}
```



- ***Rabbit*** is a system modeling language for formally verifying cybersecurity, its main target being non-experts in formal verification.
- The (current) implementation of Rabbit works as a translator to the input lang. of the Tamarin Prover**, which is a model checker for security protocols.
- Unlike Tamarin, which is based on the theory of multiset rewriting, Rabbit provides more familiar syntax to describe IoT systems.

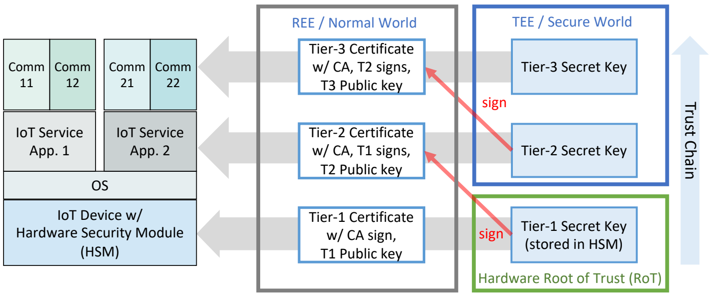
** Tamarin Prover: https://tamarin-prover.github.io/

Software Certification Service



- Security in research software, or software reliability, is now a crucial issue in research communities to certify that research software is properly and securely executed.
- We propose a software certification framework to assure software reliability, where the user can confirm:
  - The integrity of the software by the digital signature of the software developer/integrator
  - The vulnerability assessment results from the digital signature of the authorized third-party organization.

Tiered Key Management using TEE


REE / Normal World
TEE / Secure World
Comm 11
Comm 12
Comm 21
Comm 22
IoT Service App. 1
IoT Service App. 2
OS
IoT Device w/ Hardware Security Module (HSM)
Tier-3 Certificate w/ CA, T2 signs, T3 Public key
Tier-2 Certificate w/ CA, T1 signs, T2 Public key
Tier-1 Certificate w/ CA sign, T1 Public key
sign
sign
Tier-3 Secret Key
Tier-2 Secret Key
Tier-1 Secret Key (stored in HSM)
Hardware Root of Trust (RoT)
Trust Chain

- We propose a tiered PKI key management scheme using TEE to address integrity and authenticity:
  - A multi-authenticator model is applied to realize the trust chain from RoT.
  - Hardware RoT is used as the Tier-1 secret key.
  - Tier-2/Tier-3 keys are generated and stored in TEE for each application/communication.
- We have developed the prototype system using Arm TrustZone, OP-TEE, and PKCS #11 and confirmed that the secret key stored in TEE can be used in client auth. of the MQTT communication over TLS.

This work was supported by JST, CREST Grant Number **JPMJCR21M3.**

NII

National Institute of Informatics
https://ccrd.nii.ac.jp/sc23/

SC23 DENVER NOV 12–17