Practical runtime verification of quantum-future communication software

Peter Sýs

Mathematical institute
Slovak Academy of Sciences

Secure Communication in the Quantum Era

- supported by **NATO Science for Peace and Security Programme**
- project create quantum future safe cryptographics protocol and its implementation for key agreement

Group Key Establishment in a Quantum-Future Scenario

- key establishment quantum-future safe protocol
- contains authentication based on pre-shared secret
- from cryptographics primitives, require post-quantum safe KEM and MAC

* María Isabel González Vasco, Ángel L. Pérez del Pozo, Rainer Steinwandt, Group Key Establishment in a Quantum-Future Scenario, Informatica 31(2020), no. 4, 751-768, DOI 10.15388/20-INFOR427

Secure Implementation of a Quantum-Future GAKE Protocol

- protocol implementation
- minor performance related modification of protocol
- simple communication application based on cryptographics protocol
- runtime verification-trusted execution environment(RV-TEE)

*Abela, R. *et al.* (2021). Secure Implementation of a Quantum-Future GAKE Protocol. In: Roman, R., Zhou, J. (eds) Security and Trust Management. STM 2021. Lecture Notes in Computer Science(), vol 13075. Springer, Cham. https://doi.org/10.1007/978-3-030-91859-0_6

Attacks to system

Is bugless application really secure?

- regardless of security of specific application, usually there is still possible attack it via system
- security should consider also hardware, firmware, operation system, …
- attack to any level of system can attack everything on top of attack target

Isolated execution

- Isolating critical part of application improve security
- you want create 2 independent environments: secure and unsecure
- communication between secure and unsecure environments should be minimalistic and strictly defines
- secure part should be minimal(reduce probability of bug) and with higher coding&QA standards

Hardware security module(HSM)

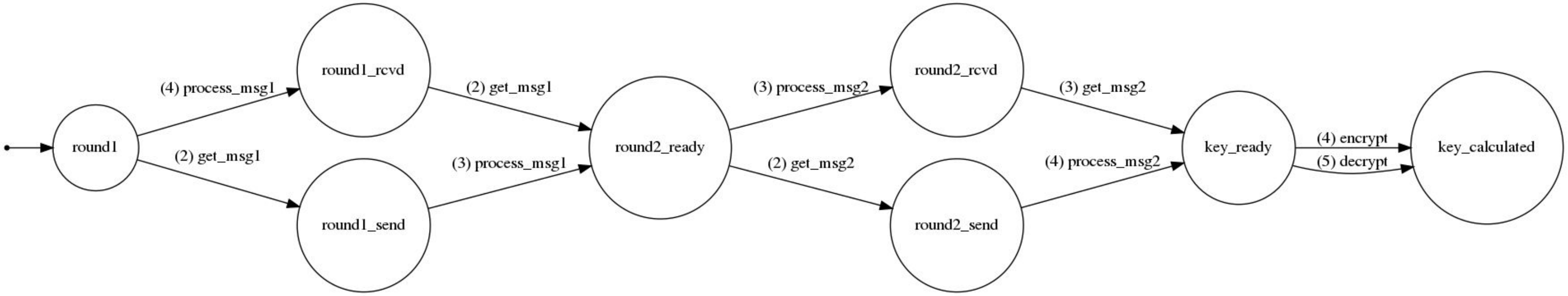
- physical device used to move execution to dedicated hardware
- USB stick, PCI card, …
- very often for PKI

Trusted execution environment(TEE)

- define some isolated service inside system(usually called “secure enclave”)
- usually based on hardware functionality
- can be implemented on software or firmware level

Runtime verification(RV)

- analyzation of application run and verifying it correctness
- require knowledge of application properties which is verified
- property can be application independent(ex. verifying memory access) or application specific
- require event collector(usually use techniques similar to debuger)
- require monitor which process captured events and verify properties



RV-TEE for GAKE protocol

Use combination of RV and HSM to provide additional security of implementation

- TEE part
  - HSM
  - USB device
  - based pn SEcube™
  - run computation part of protocol
- RV part
  - use dynamic instrumentation based on Frida toolkit
  - finite automata based properties checker based on LARVA runtime verification platform

RV checked properties

Crypto related properties

- basic protocol flow(ex. round 1 run before round 2)
- randomness quality
- correct usage of crypto primitives

Low level properties

- pointer arguments are always valid memory reference
- sensitive informations are always scrubbed from memory

High level properties

- chat room properties
- transferred data is valid text

???