Threat Modelling Guided Trust-based Task Offloading for Resource-constrained Internet of Things

SenSys 2022, Boston, USA

8:40-8:51 8th November 2022

Matthew Bradbury, Arshad Jhumka, Tim Watson, Denys Flores, Jonathan Burton and Matthew Butler

https://petras-iot.org/

IoT Task Offloading

- IoT devices have limited resources
- Potentially want these devices to perform expensive tasks
- Tasks require too many resources
  - Big ML models (too much RAM)
  - Large datasets (too much Flash)
  - Computationally expensive (too much CPU)
- Instead – Offload tasks to the Edge

Zolertia RE Mote (CC2583)

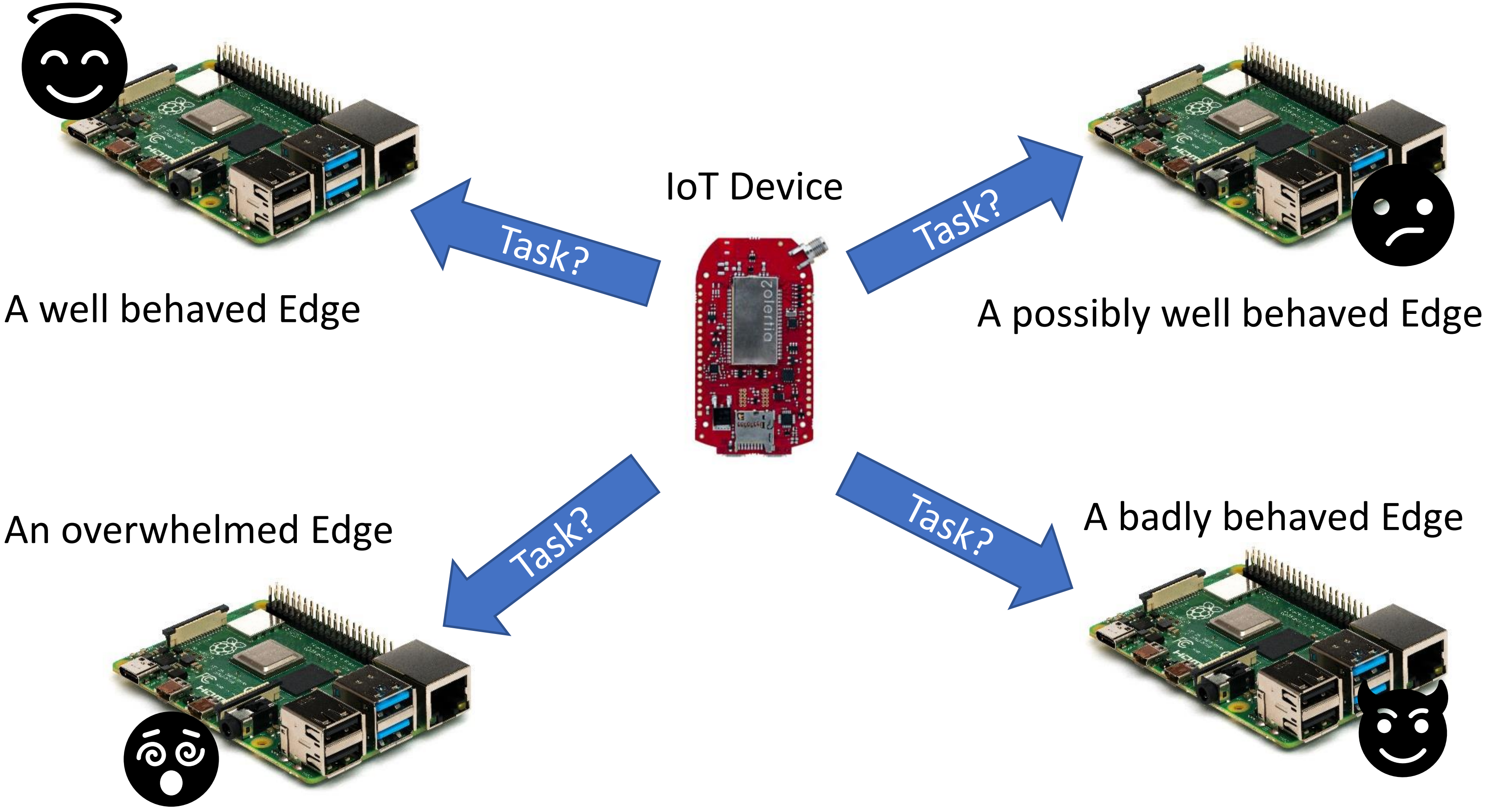
- CPU: 32 MHz
- RAM: 32 KiB
- Flash 512 KiB

nRF 52840

- CPU: 64 MHz
- RAM: 256 KiB
- Flash: 1 MiB

- Other hardware platforms have similar specifications

Task Offloading



- Resource-constrained IoT device offloads expensive tasks to resource-rich Edge
- **How to decide who to offload to**?
- Measure trustiness of accepting task and executing it correctly and timely

Assessing if an Edge can be trusted is hard

- Typical: Store large amounts of data on actions and feed into a trust model
- IoT devices do not have the memory/flash capacity for this
- Reality: Need to use lightweight trust models
- Beta Reputation System

$$E[X] = \frac{\alpha}{\alpha + \beta} \text{ where } X \sim \mathrm{Beta}(\alpha, \beta)$$

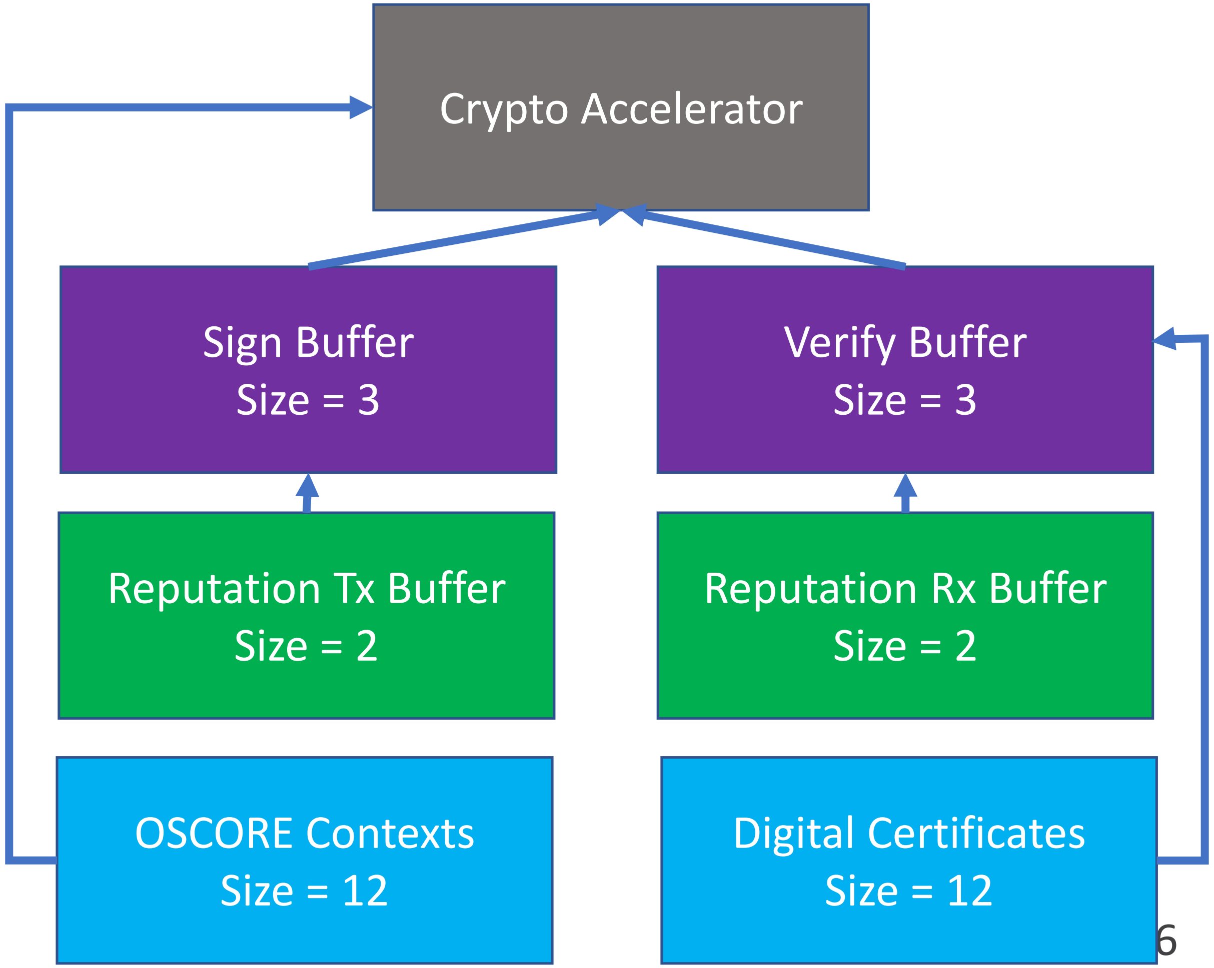
Category	Flash (B)	Flash (%)	RAM (B)	RAM (%)
applications/monitoring	1388	1.2	384	1.3
applications/routing	3968	3.3	505	1.7
contiki-ng	7232	6.0	826	2.8
contiki-ng/cc2538	14 572	12.1	2356	7.9
contiki-ng/coap	8774	7.3	2388	8.0
contiki-ng/net	27 080	22.5	8236	27.8
contiki-ng/oscore	5652	4.7	1010	3.4
newlib	26 415	22.0	2534	8.5
system/common	3420	2.8	37	0.1
system/crypto	7022	5.8	5173	17.4
system/mqtt-over-coap	1494	1.2	503	1.7
system/trust	13 106	10.9	5724	19.3
Total Used	120 123	100	29 676	100
Total Available	524 288		32 768	

Threats via system implementation

- Limited resources mean denial of service attacks are very easy to perform
  - On memory buffers
  - On computational resources (e.g., cryptographic accelerators)
- Also need to consider the capability to impact trust assessment
  - Can an adversary eliminate history of their bad behaviour?
- System design is important to ensure that an attack on one sub-system does not have a significant impact on another

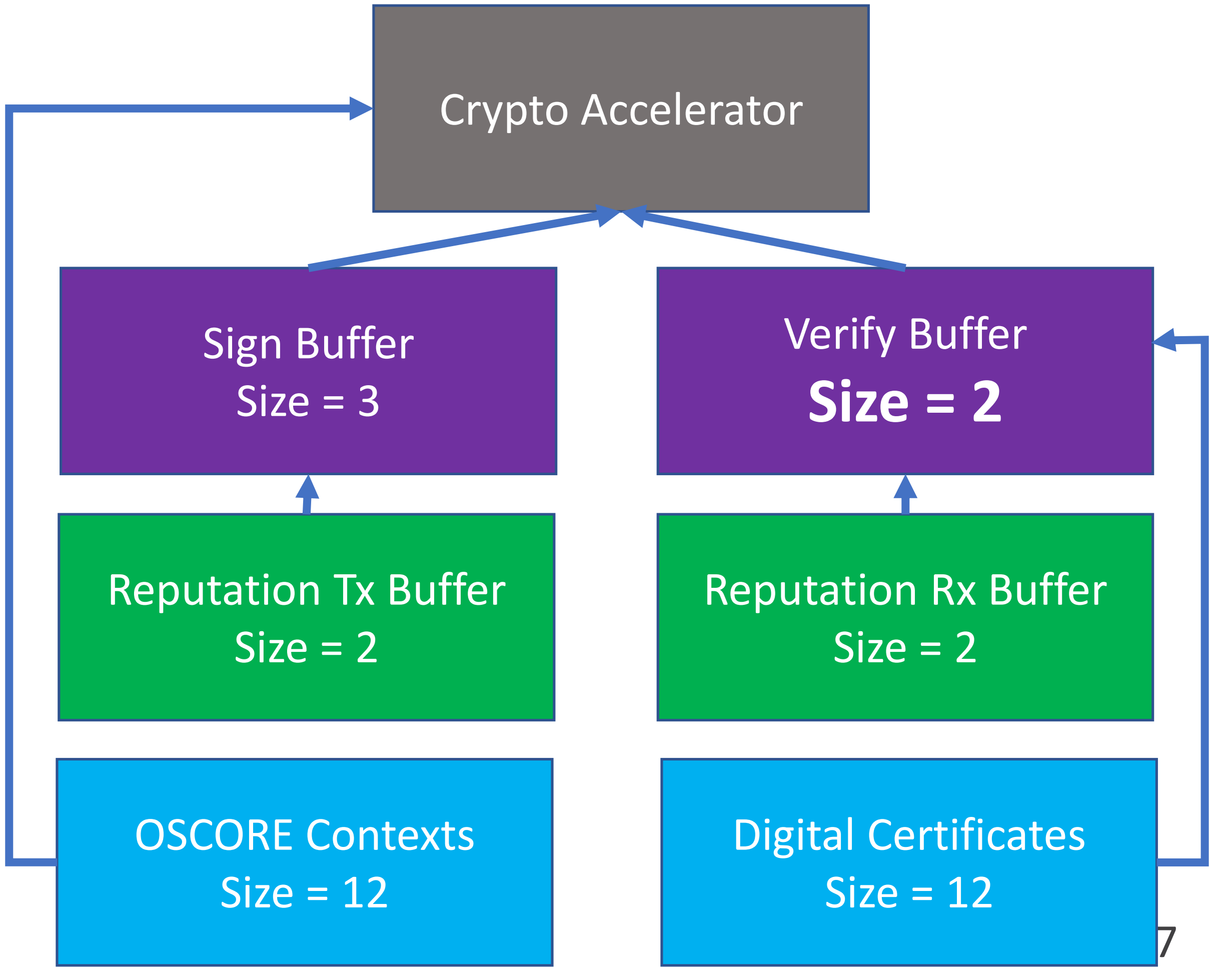
Attack: Signature Verification DoS

- Shared cryptographic accelerator
  - Sign: 360ms
  - Verify: 711ms
  - ECDH: 344 ms
- Cannot sign/verify/ECDH at the same time
- Pressure on signature verification
  - To check received reputation
  - To verify digital certificates



Attack: Signature Verification DoS

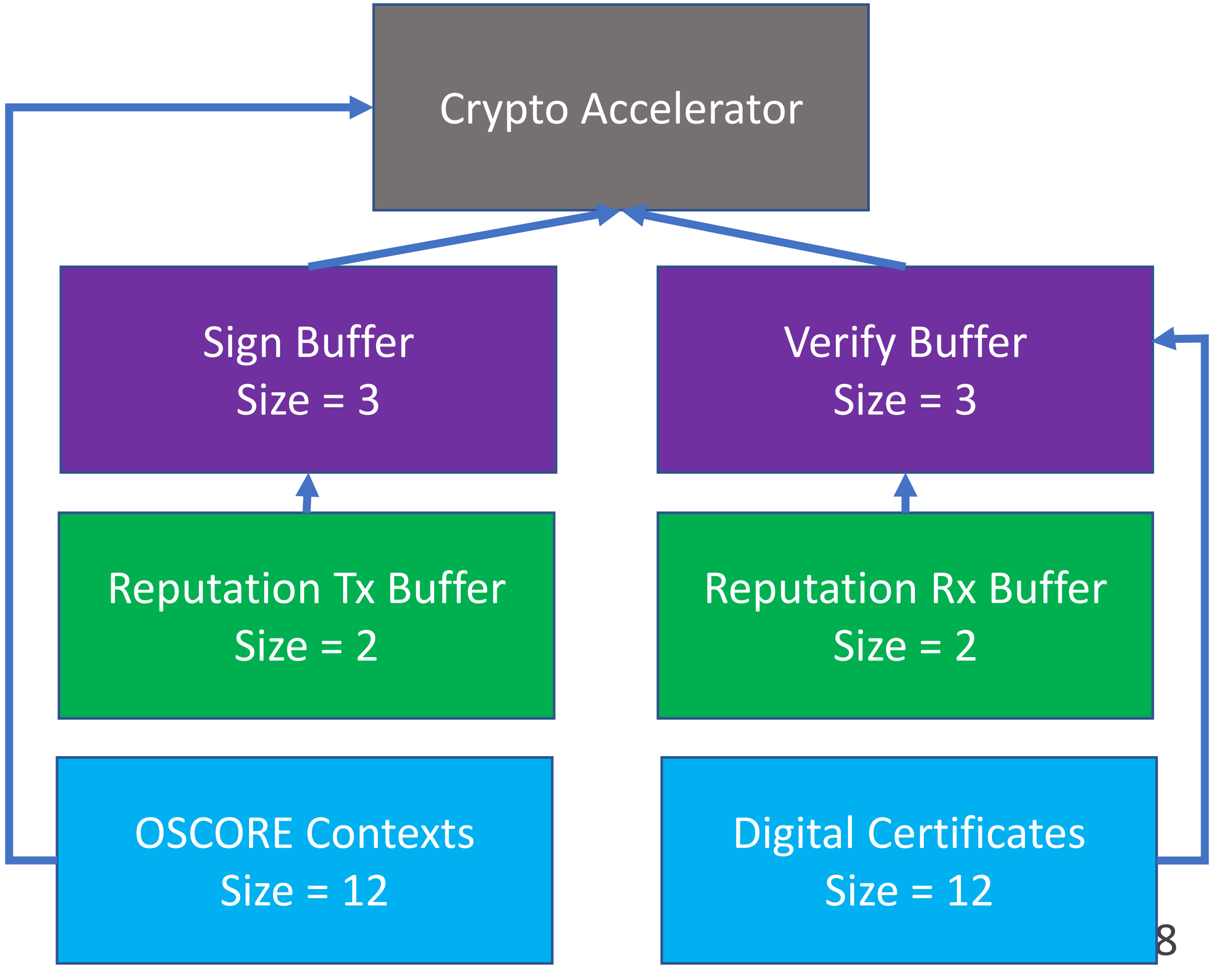
- Pressure on verify buffer from two sources
- Adversary repeatedly broadcasting signed reputation messages
- Verify buffer too small can prevent digital signature verification
- Which prevents establishing security contexts with new Edges
- Also prevents verifying genuine reputation messages



Subtle bug discovered during testing

- Also need to consider fairness of access to crypto accelerator
- Contiki-NG uses cooperative instead of pre-emptive scheduling
- Implementation did not yield after sign/verify/ECDH
- So possible to keep verifying and never sign/ECDH

https://github.com/MBradbury/iot-trust-task-alloc/commit/c6c1b1cd36101a7155b908325fb48fc136b61995



Attack: Remove Bad Interactions

- Limited memory in IoT devices
- More Edges than space in memory -> need to think about who to keep
- Complex due to how an Edge can add/remove capabilities and their availability

- Announce – Edge says they are available
- Capability Add – Edge says they have the capability to execute a type of task
- Capability Remove – Edge no longer can execute a certain type of task
- Unannounce – Edge and its capabilities no longer available

M. Bradbury, A. Jhumka, and T. Watson. Information Management for Trust Computation on Resource-constrained IoT Devices. Future Generation Computer Systems, 135:348–363, 2022. doi:10.1016/j.future.2022.05.004.

Attack: Remove Bad Interactions

- Eager Removal
  - Simple to implement and low overhead
  - Adversary able to use to make IoT devices forget bad behaviour
- Lazy Removal
  - Complex to implement and higher memory/computational costs
  - Limits adversaries capability to force IoT devices to forget bad behaviour
  - As long as there are fewer bad adversaries than space in memory

Conclusions

- Resource-constraints make some attacks highly feasible
- Some capability to mitigate
- Careful design, implementation and testing/verification needed

Thank you for attending, any questions?