

TEEzz: Fuzzing Trusted Applications on COTS Android Devices

Marcel Busch, Aravind Machiry, Chad Spensky,
Giovanni Vigna, Christopher Kruegel, Mathias Payer

{marcel.busch, mathias.payer}@epfl.ch
amachiry@purdue.edu, chad@allthenticate.net
{chris, vigna}@ucsb.edu



HexHive

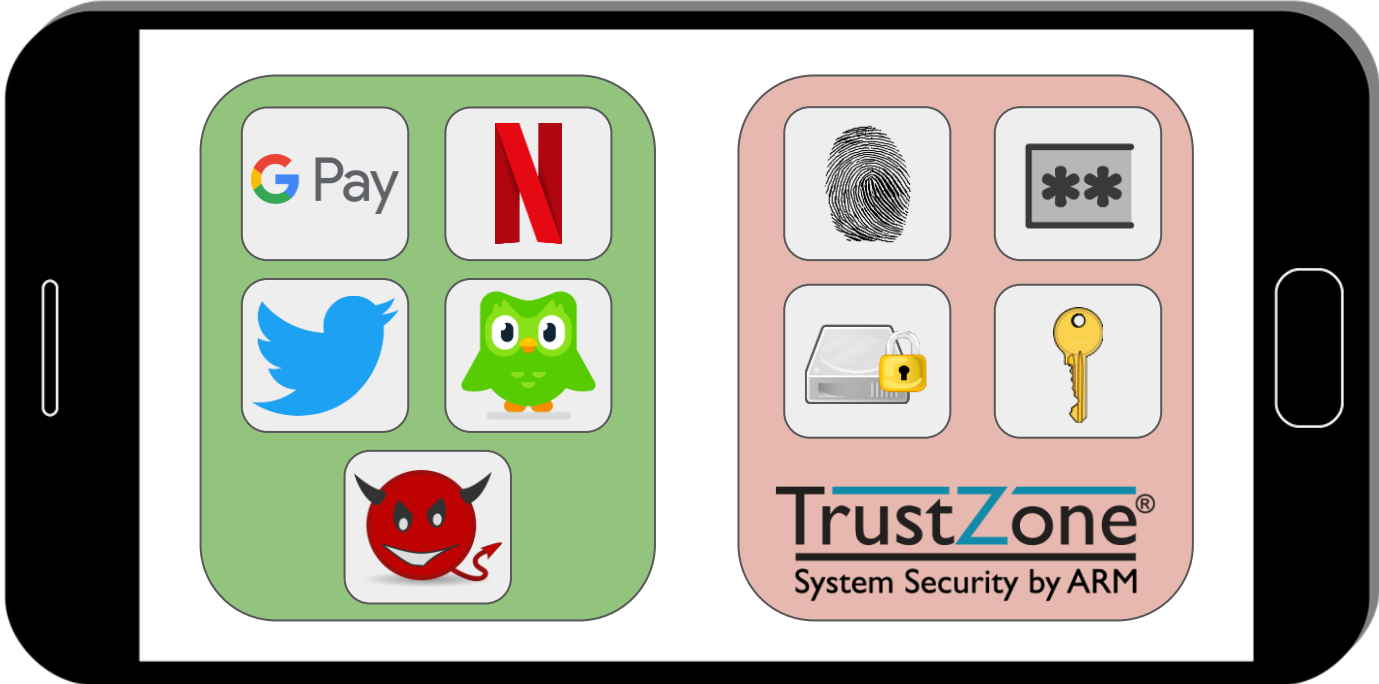
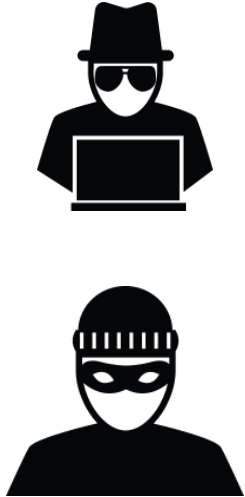
EPFL



PURDUE
UNIVERSITY®

UC SANTA BARBARA

Modern TZ-based TEEs on Android Mobile Devices

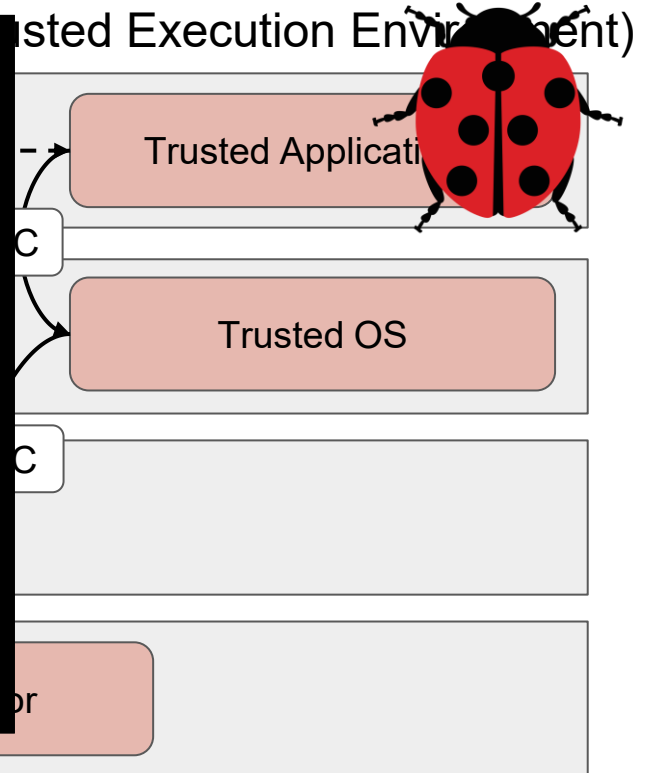


ARM TrustZone Privilege Levels

Normal World

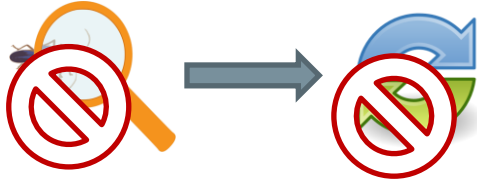


Secure World

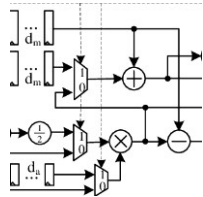


Challenges of Fuzzing Trusted Applications

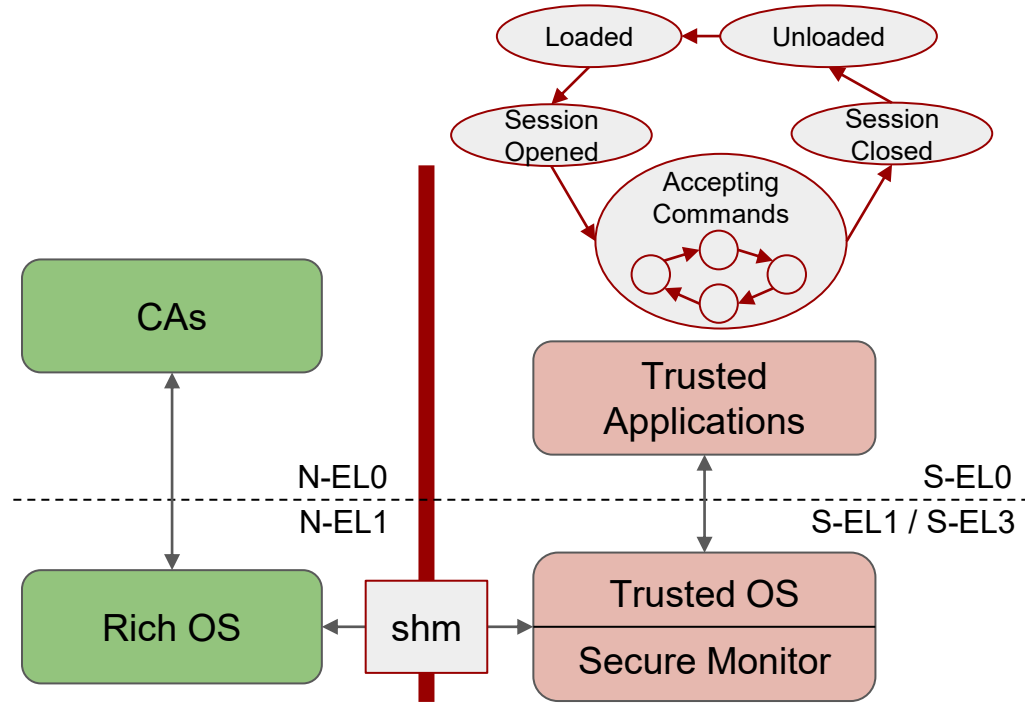
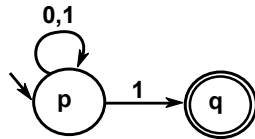
1. Limited introspection



2. Complex input

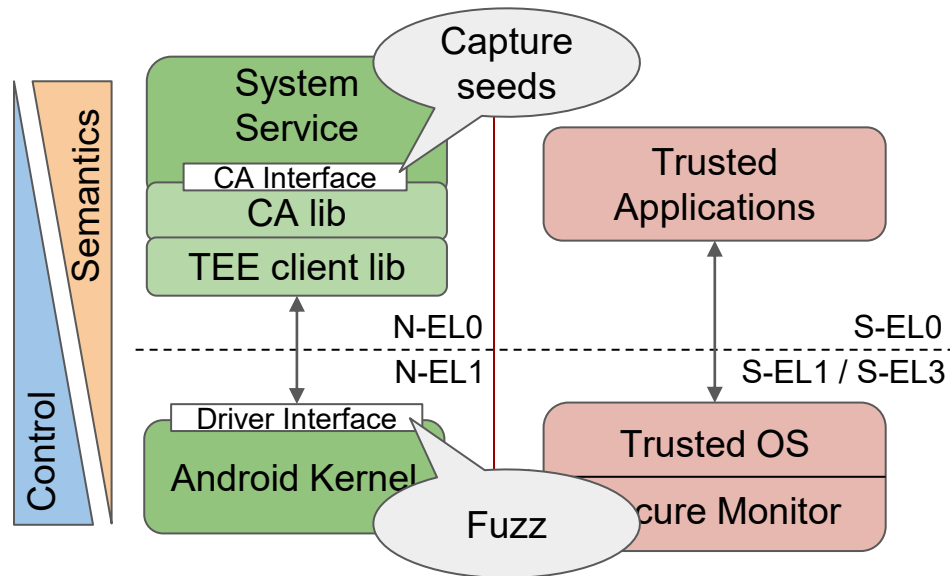


3. Statefulness

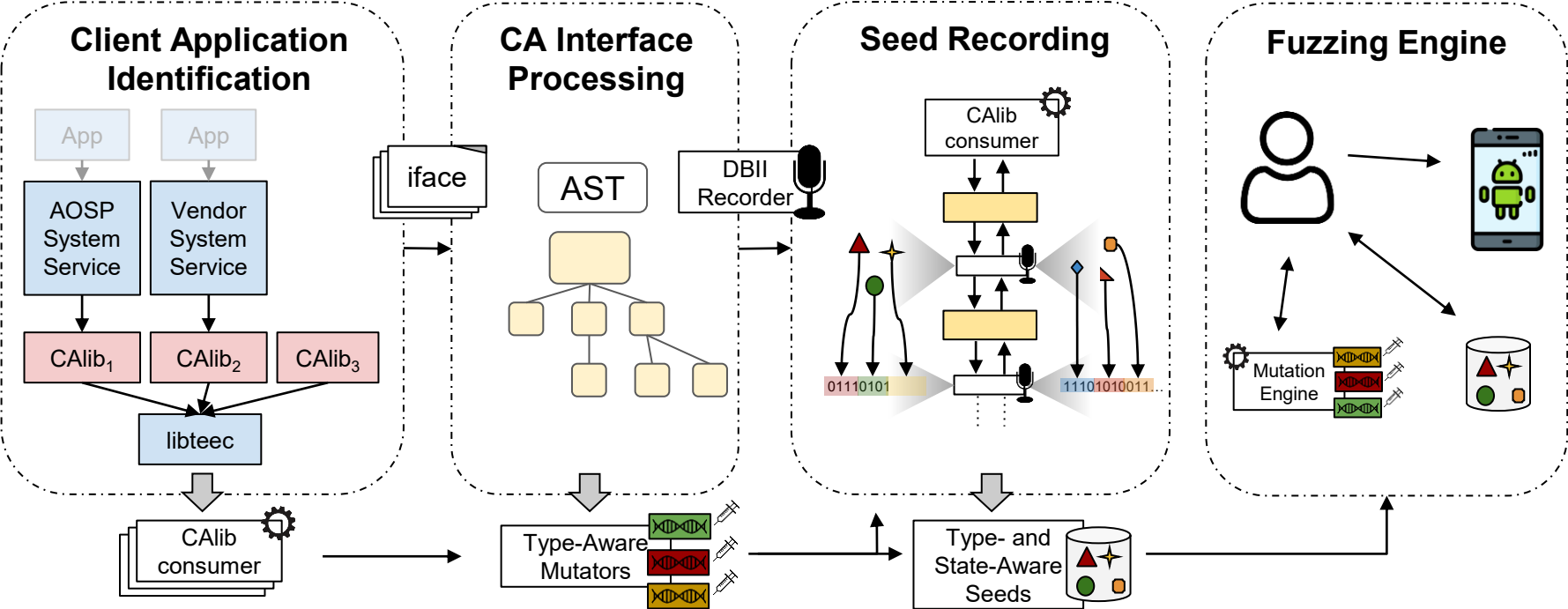


Observations and Intuitions

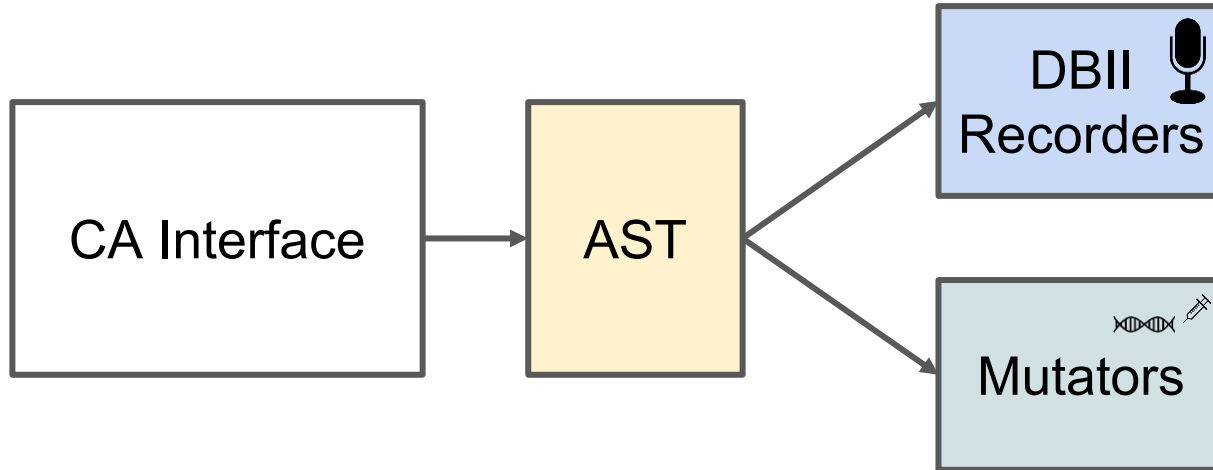
- Clients located in “normal world”
- Semantics decrease towards lower levels of abstraction
- Control over input increases towards lower levels of abstraction



TEEzz – End-to-End



Automatically Generating Type-aware Seeds

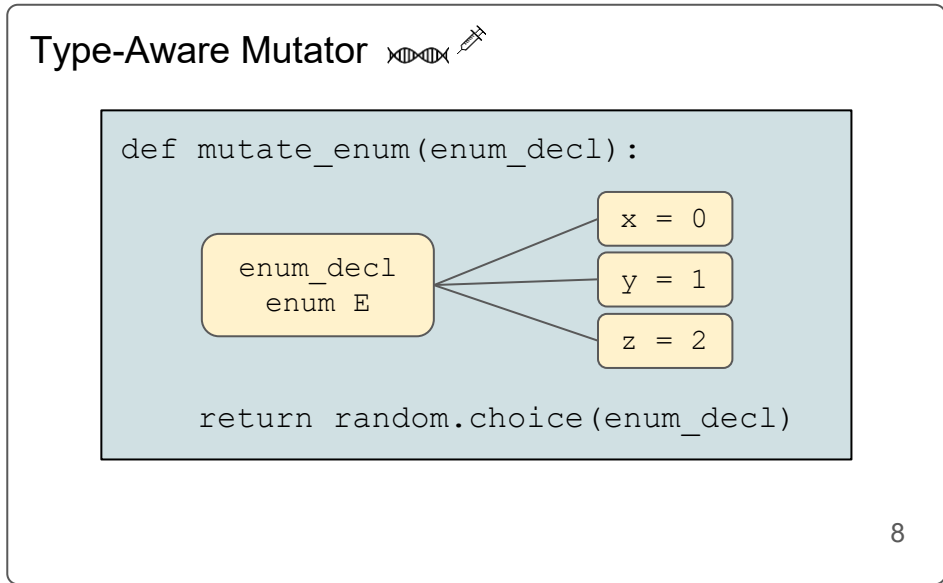
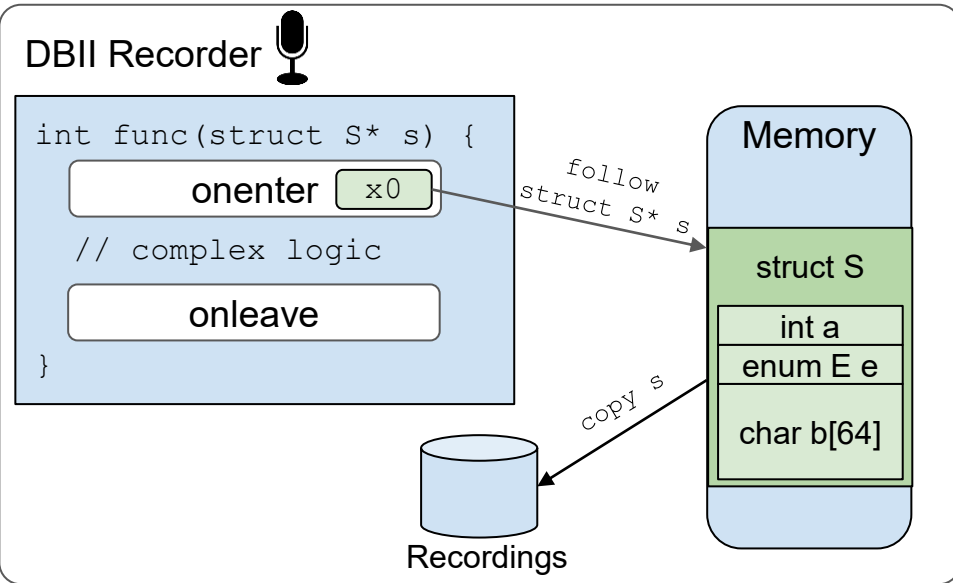
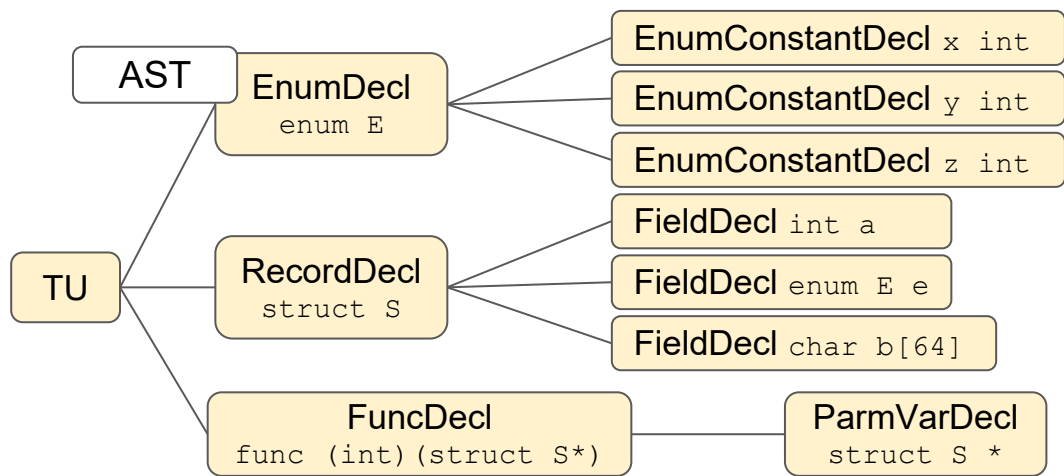
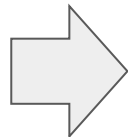


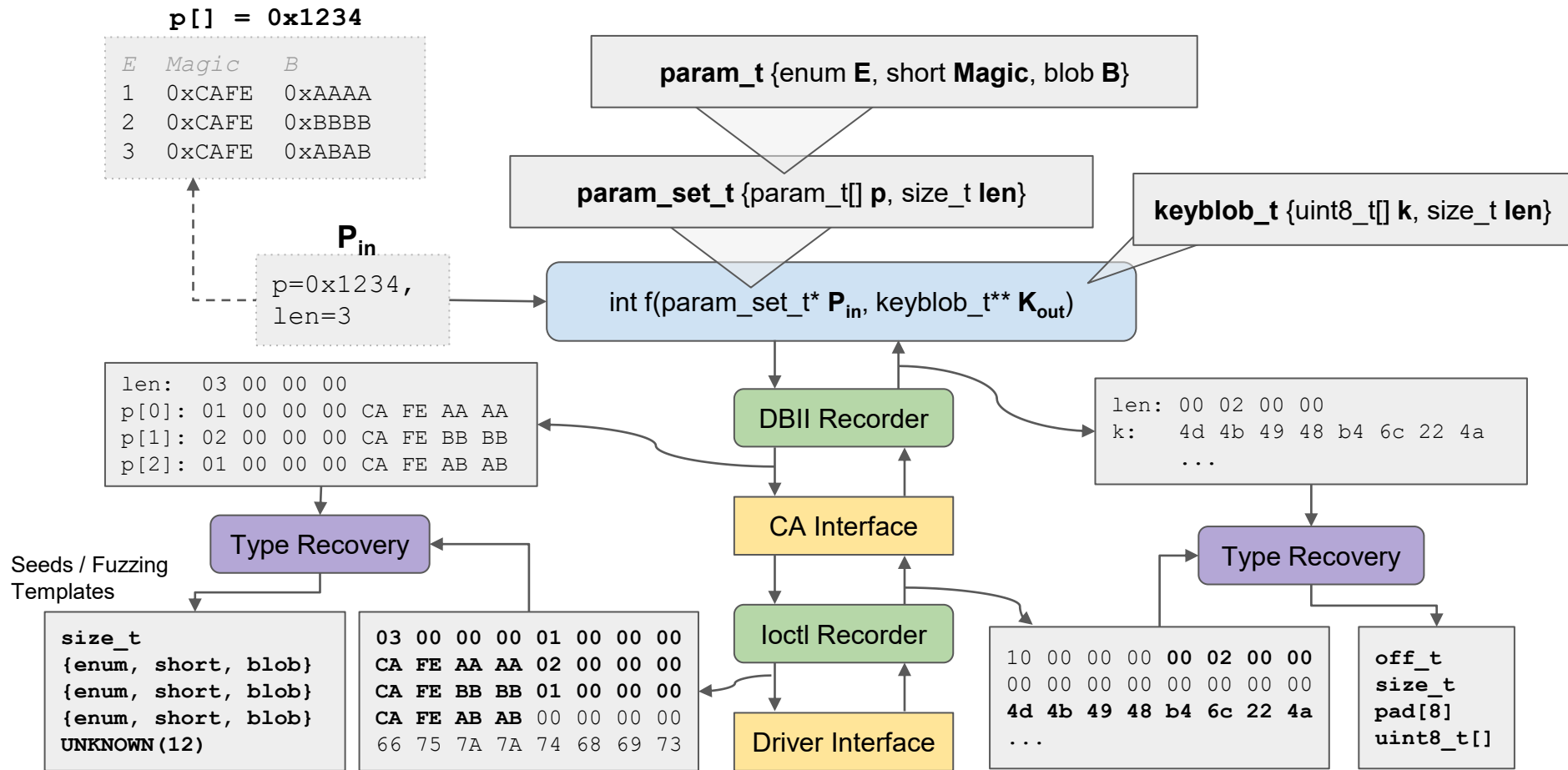
Interface Definition

```
enum E { x, y, z };

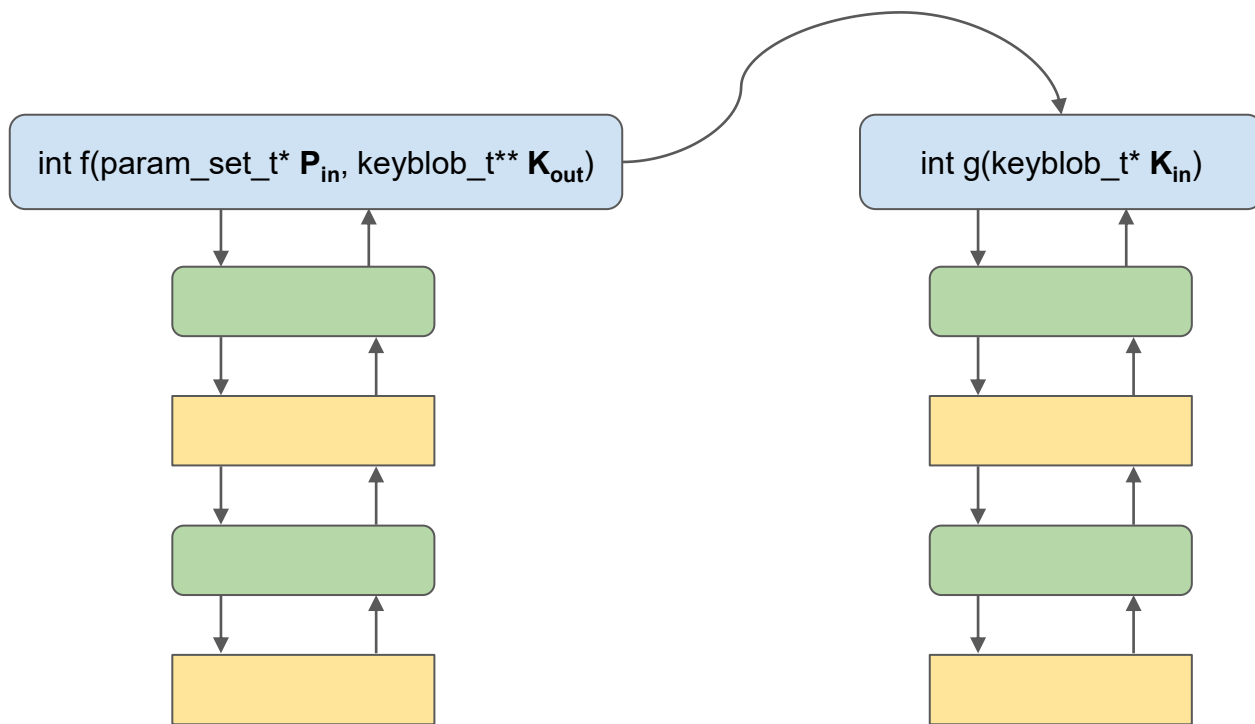
struct S {
    int a;
    enum E e;
    char b[64];
}

int func(struct S* s);
```

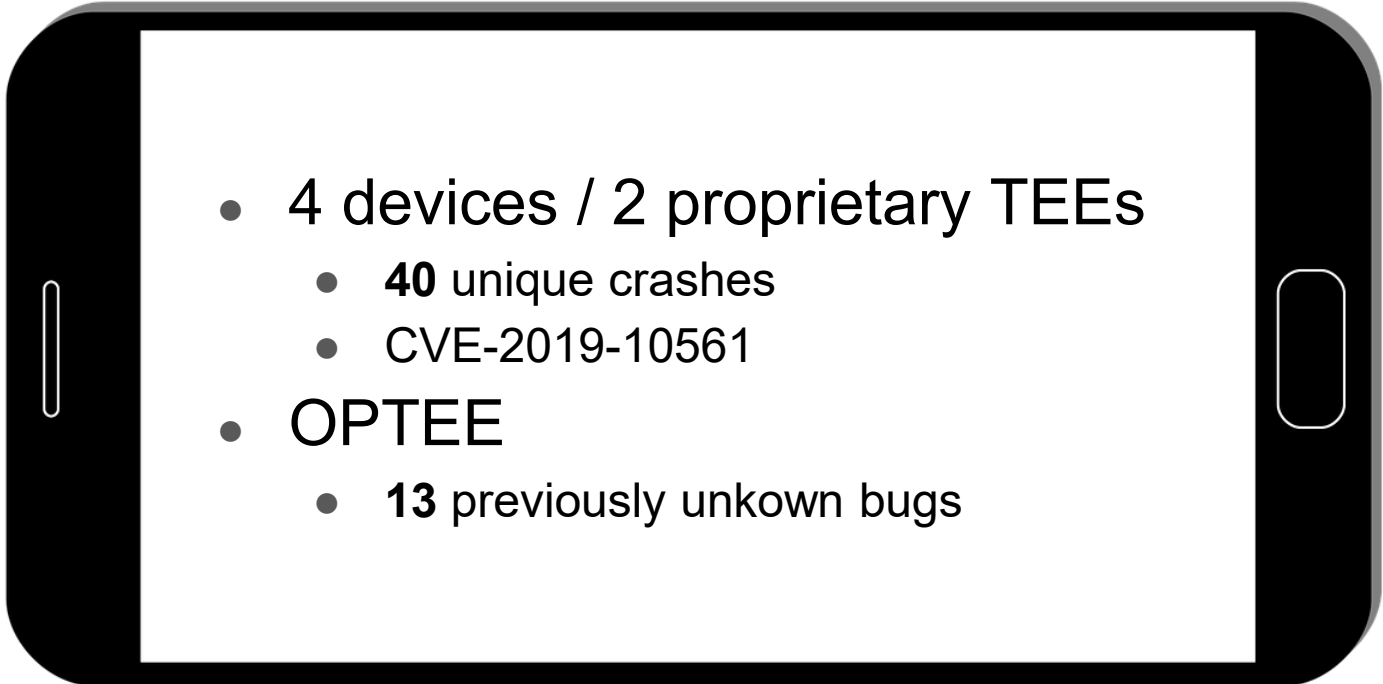




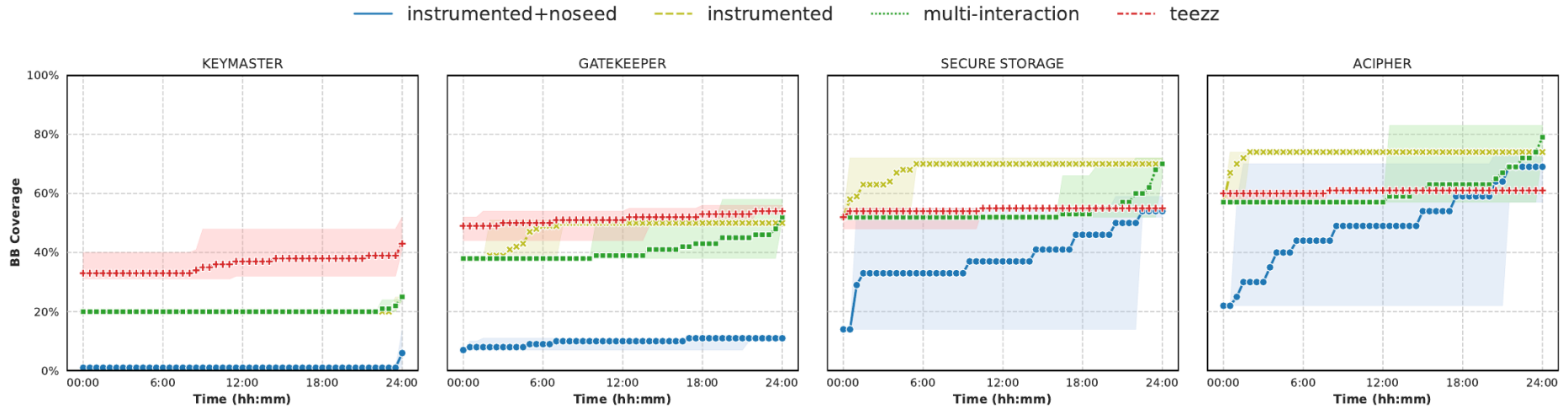
State-Awareness



Evaluation – Finding Bugs

- 
- 4 devices / 2 proprietary TEEs
 - **40** unique crashes
 - CVE-2019-10561
 - OPTEE
 - **13** previously unknown bugs

Evaluation – Ground-truth Coverage Experiments



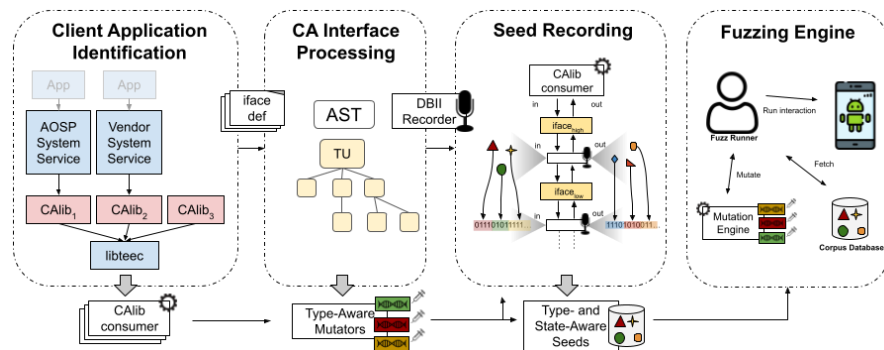
TEEzz: State- and Type-aware Black-box Fuzzing



Paper



HexHive



- 4 devices / 2 proprietary TEEs
40 unique crashes
CVE-2019-10561
- OPTEE
13 previously unknown bugs