Tampere University

Point Intervention: Improving ACVP Test Vector Generation Through Human Assisted Fuzzing

2024 International Conference on Information and Communications Security (ICICS 2024)

Iaroslav Gridin and Antonis Michalas

Iaroslav Gridin,

Network and Information Security Group (NISEC),

Tampere University,

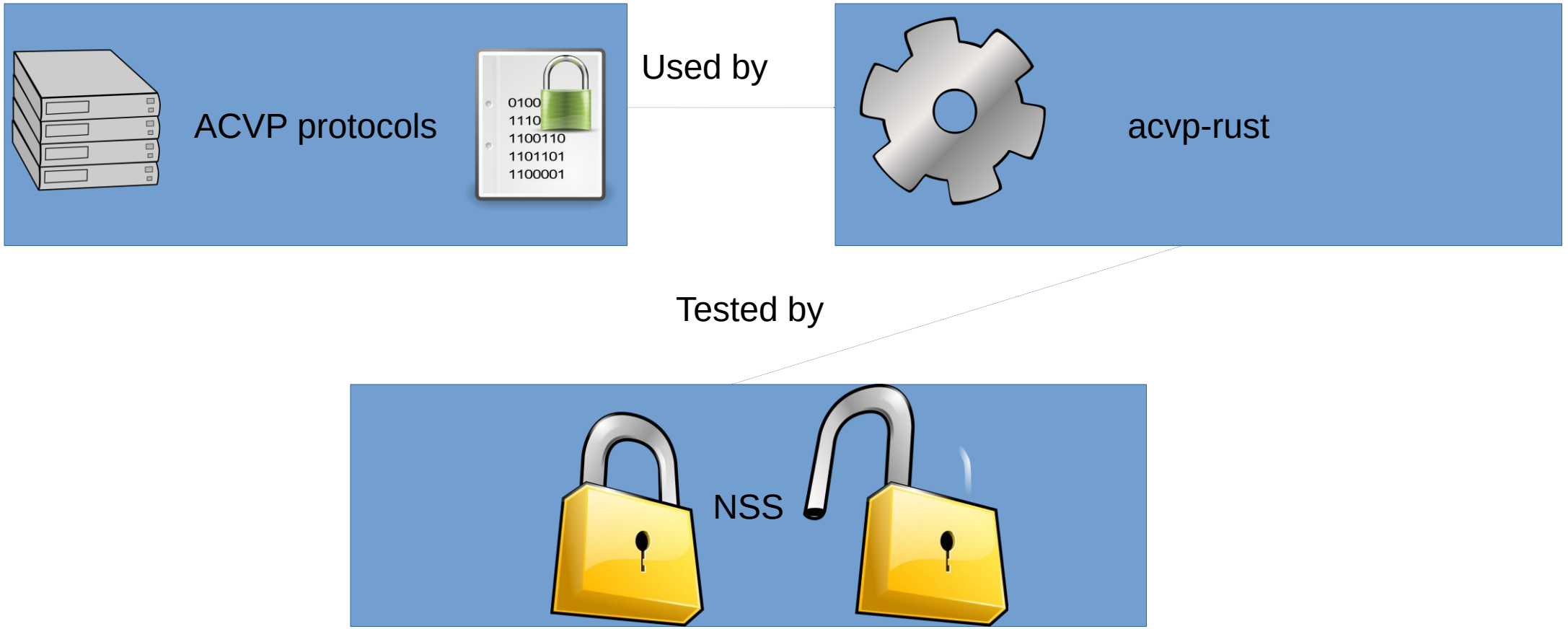
Tampere, Finland

https://research.tuni.fi/nisec/

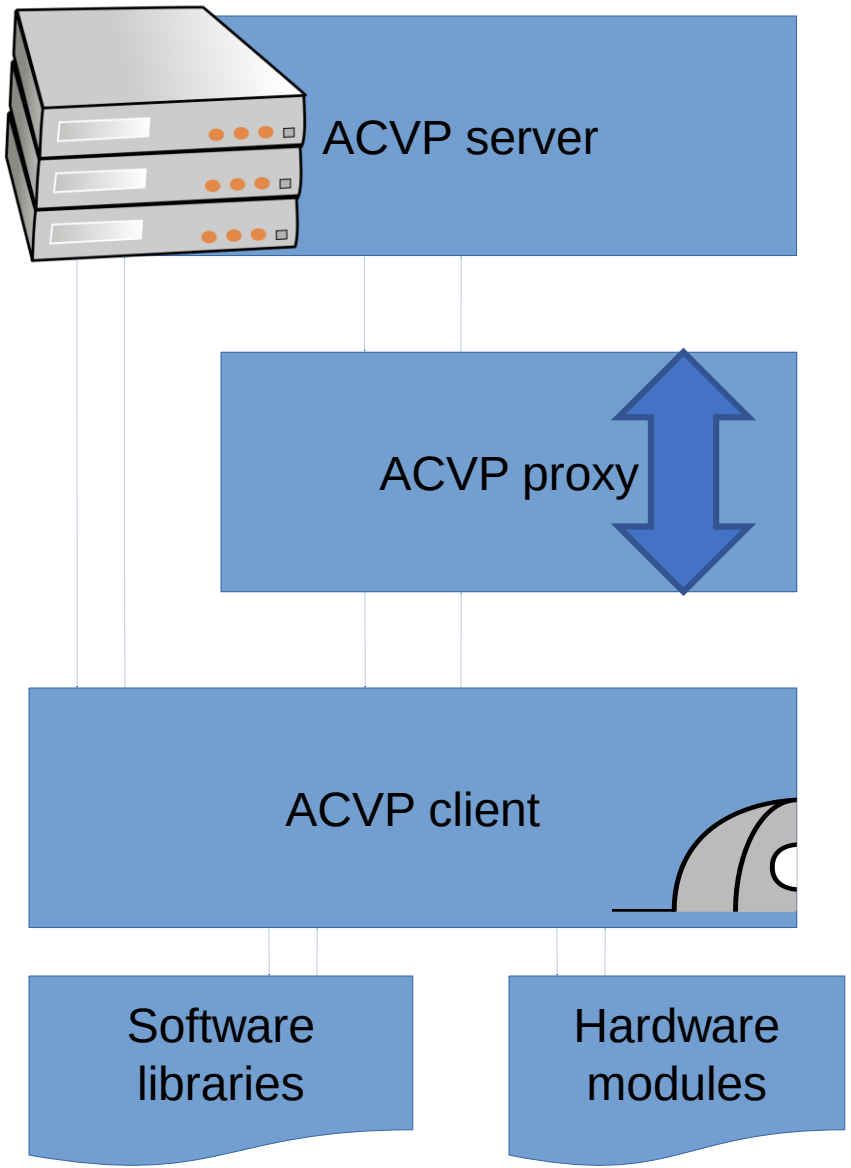
iaroslav.gridin@tuni.fi

Tampere University

Outline

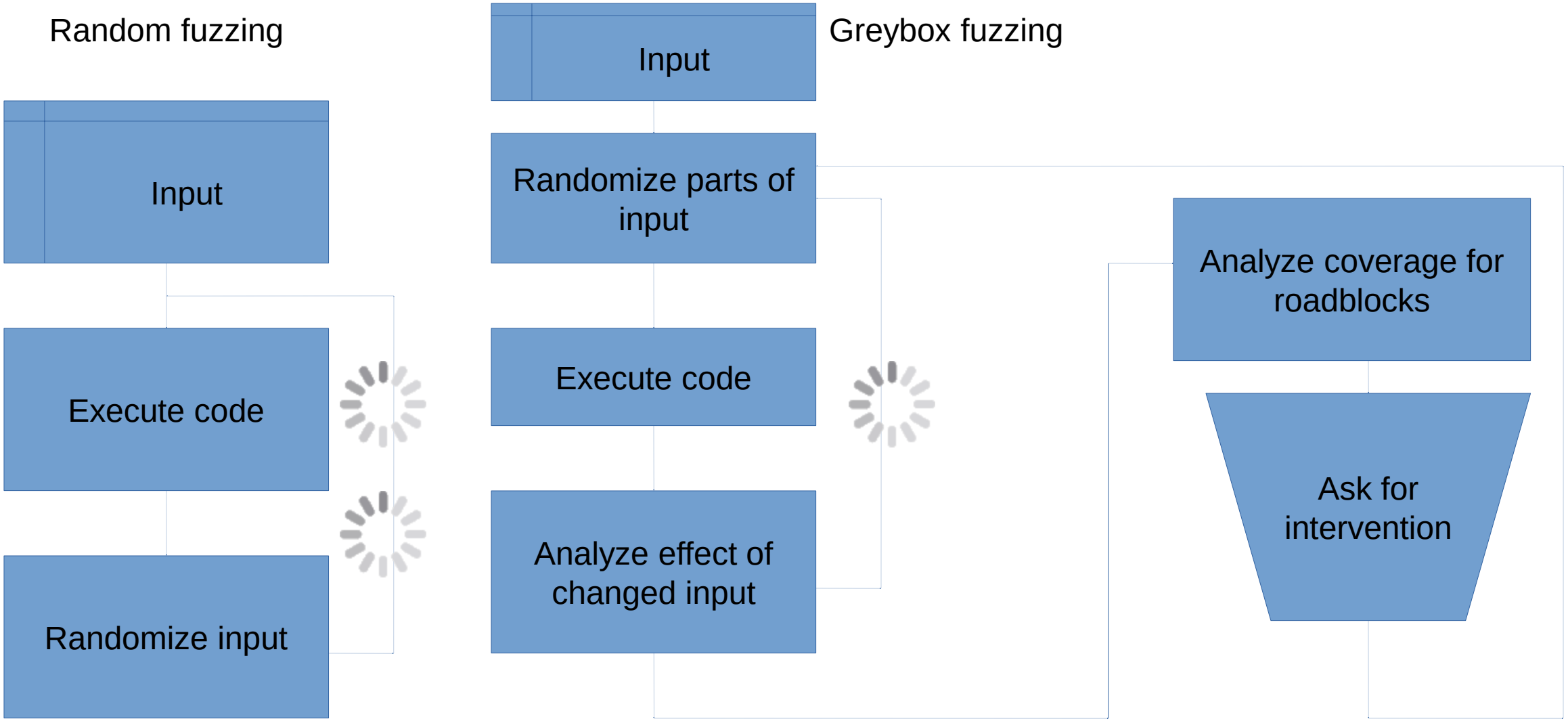


ACVP



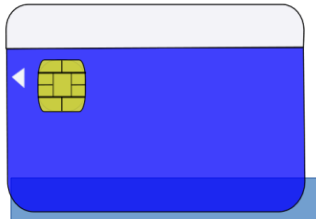

```
{
  "vsId": 805548,
  "algorithm": "ACVP-AES-GCM",
  "revision": "1.0",
  "isSample": true,
  "testGroups": [{
    "tgId": 1,
    "testType": "AFT",
    "direction": "encrypt",
    "keyLen": 128,
    "ivLen": 96,
    "ivGen": "external",
    "ivGenMode": "8.2.1",
    "payloadLen": 256,
    "aadLen": 120,
    "tagLen": 104,
    "tests": [{
      "tcId": 1,
      "pt": "28E3FB...9809",
      "key": "C19A...AD2",
      "aad": "E9FB...1B",
      "iv": "C4...DEFB"
    }]
  }]
}
```

Fuzzing schemes



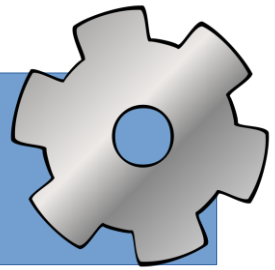
Tampere University

the framework



Cryptographic module(s)

acvp-rust



Fuzzed test corpus and annotated output

Guided human intervention

Improved corpus in ACVP format

Tampere University

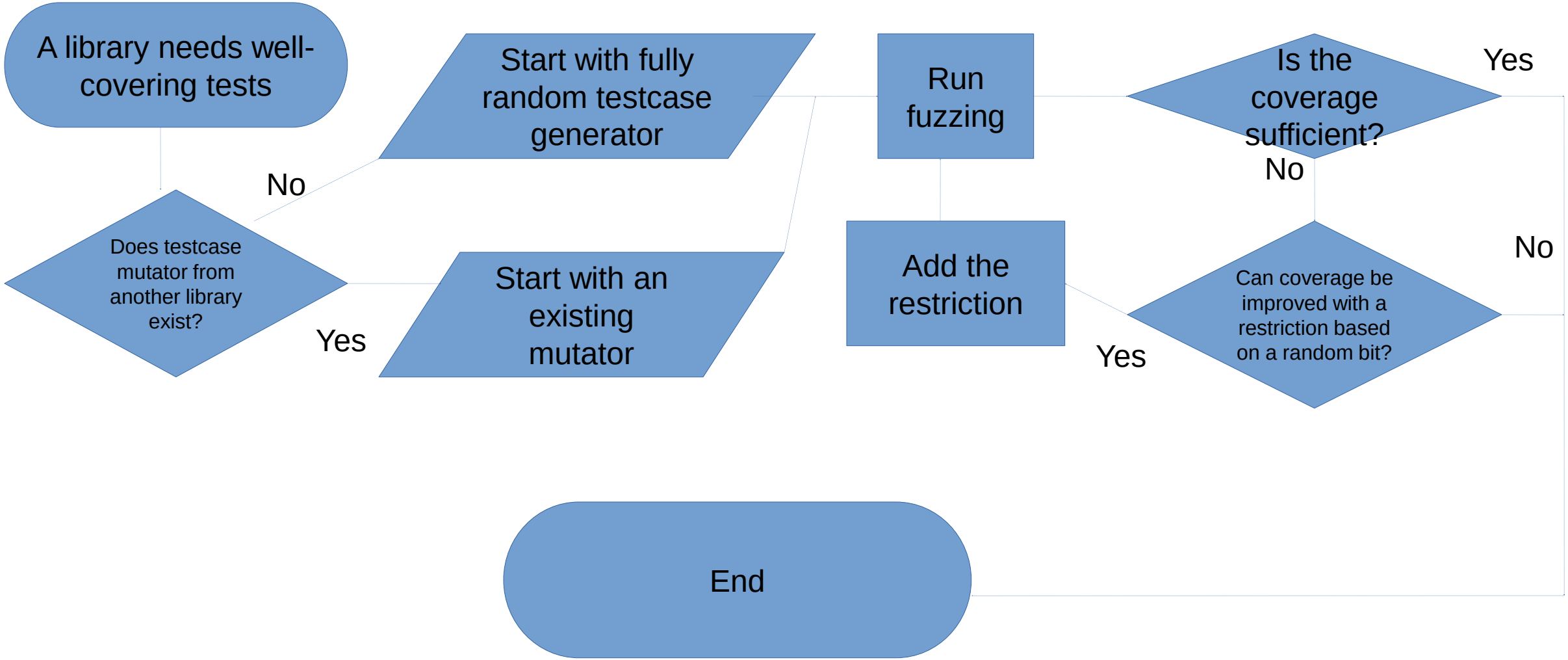
Human intervention scheme

```
1336
1337  166   hash = HASH_GetRawHashObject(hashAlg);
1338  166   dbMaskLen = emLen - hash->length - 1;
1339
1340        /* Step 3 + 4 */
1341  166   if ((emLen < (hash->length + salt
1342  166       (em[emLen - 1] != 0xbc)) {
1343  166       PORT_SetError(SEC_ERROR_BAD_SIGNATURE);
1344  166       return SECFailure;
1345  166   }
1346
1347        /* Step 6 */
1348    0   zeroBits = 8 * emLen - emBits;
1349    0   if (em[0] >> (8 - zeroBits)) {
1350    0       PORT_SetError(SEC_ERROR_BAD_SIGNATURE);
1351    0       return SECFailure;
1352    0   }
1353
```

```
1  let sane_modulus = bool::arbitrary(u)?;
2  let modulus = if sane_modulus {
3          let min_size = if let Some(min_size) =
4                  salt_len.checked_add(hash_alg.digest_len() + 2 /* +1 for possible leading zero */)
5                  min_size
6          } else {
7                  return Err(arbitrary::Error::IncorrectFormat);
8          };
9          let mut modulus =
10             HexString::clamp_length_arbitrary(Some(min_size as usize), None, u)?;
11         if modulus.data[0] == 0 {
12                 modulus.data[0] = 1;
13         };
14         if modulus.data.len() > 0 {
15                 let len = modulus.data.len();
16                 modulus.data[len - 1] = 0xbc;
17         };
18         modulus
19 } else {
20         HexString::arbitrary(u)?
21 };
```

Tampere University

Hybrid fuzzing flowchart



Results with NSS

- We used acvp-rust framework and methodology to search for vulnerability in NSS, a popular cryptographic library used in Firefox, Thunderbird and many other applications

- Using the framework allowed to produce an improved test set, compared to already present fuzzed corpus, improving coverage in critical areas.

- We discovered a bug in RSA code causing access beyond the end of an array under specific conditions. The bug and fix for it were accepted by the NSS maintainers.

- We made many other improvements to NSS preventing possible future issues.

Future Research

- Further automatization
- More subspecifications and improvements to ACVP ones
- Integrating side-channel vulnerability detection tools

Open Science & Reproducible Research

- ACVP-RUST and Results
- Check here: https://gitlab.com/nisec/acvp-rust

Tampere University

Thanks!

https://research.tuni.fi/nisec/

- Iaroslav Gridin
- Network and Information Security Group (NISEC),
- Tampere University,
- Tampere, Finland

https://research.tuni.fi/nisec/
iaroslav.gridin@tuni.fi