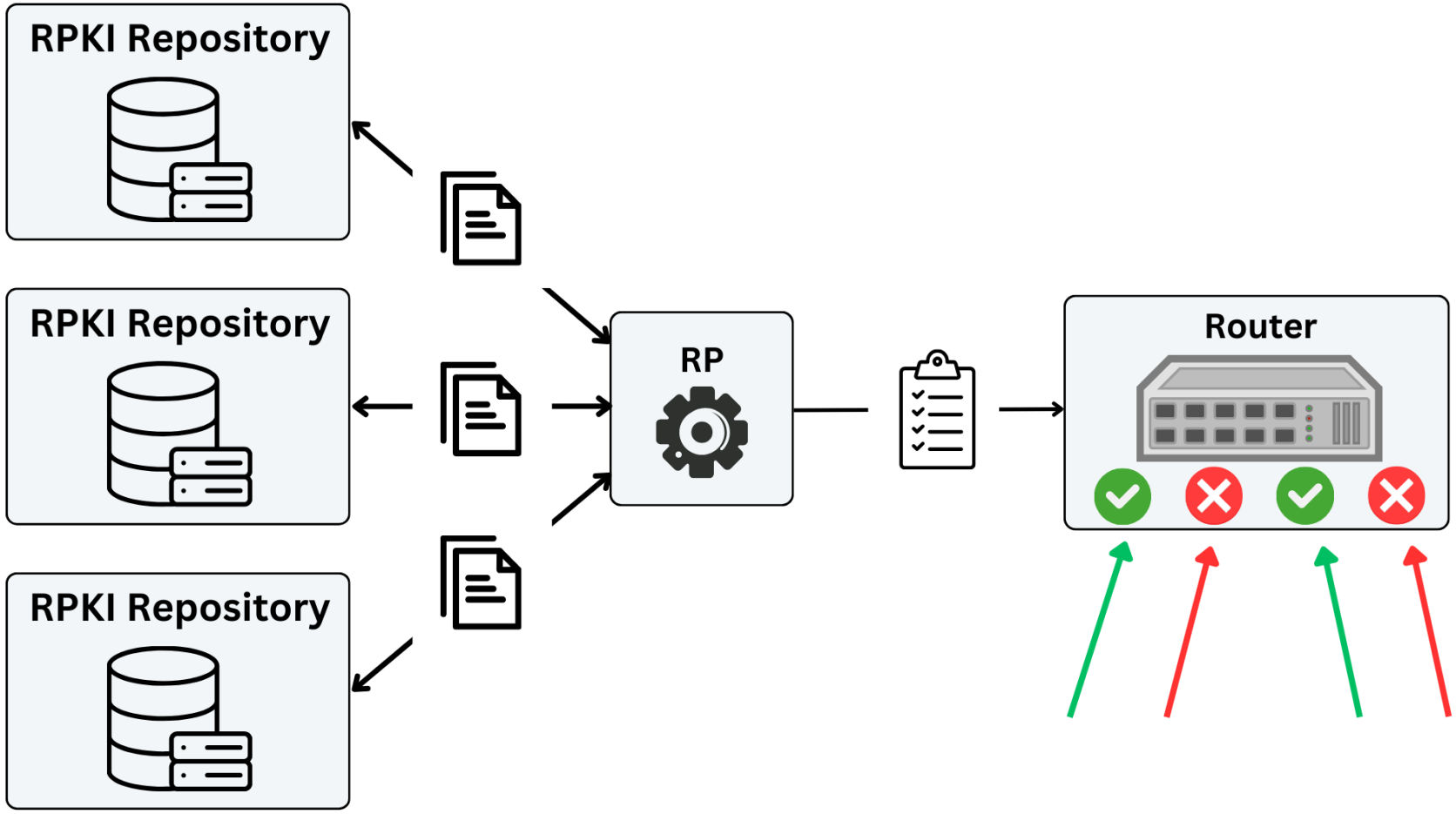
ATHENE
Nationales Forschungszentrum
für angewandte Cybersicherheit

The CURE to Vulnerabilities in RPKI Validation

Donika Mirdita, Haya Schulmann, Niklas Vogel, Michael Waidner

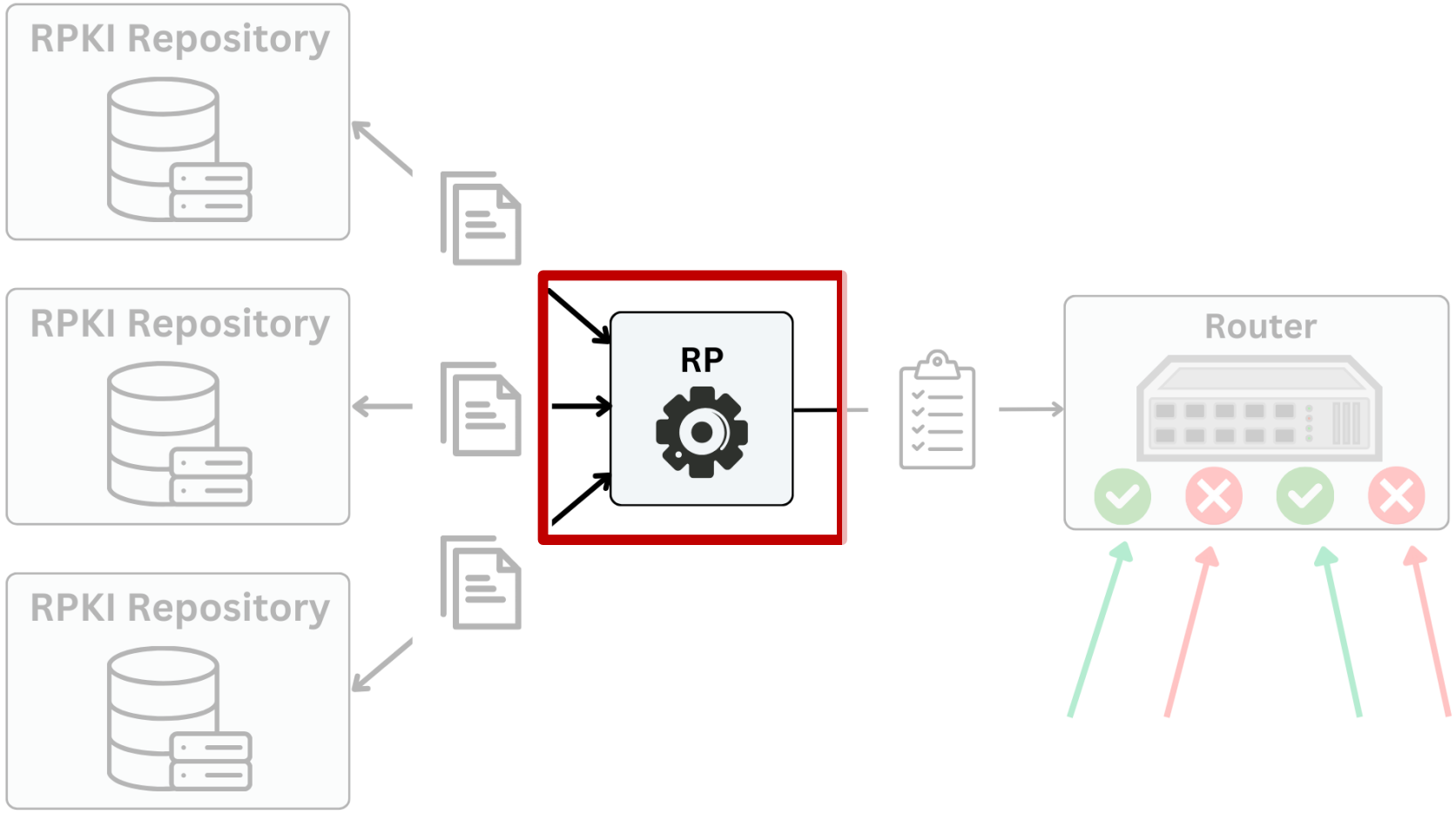
ATHENE Research Center for Applied Cybersecurity
Goethe University Frankfurt
Technical University Darmstadt
Fraunhofer SIT

A Short Introduction to RPKI



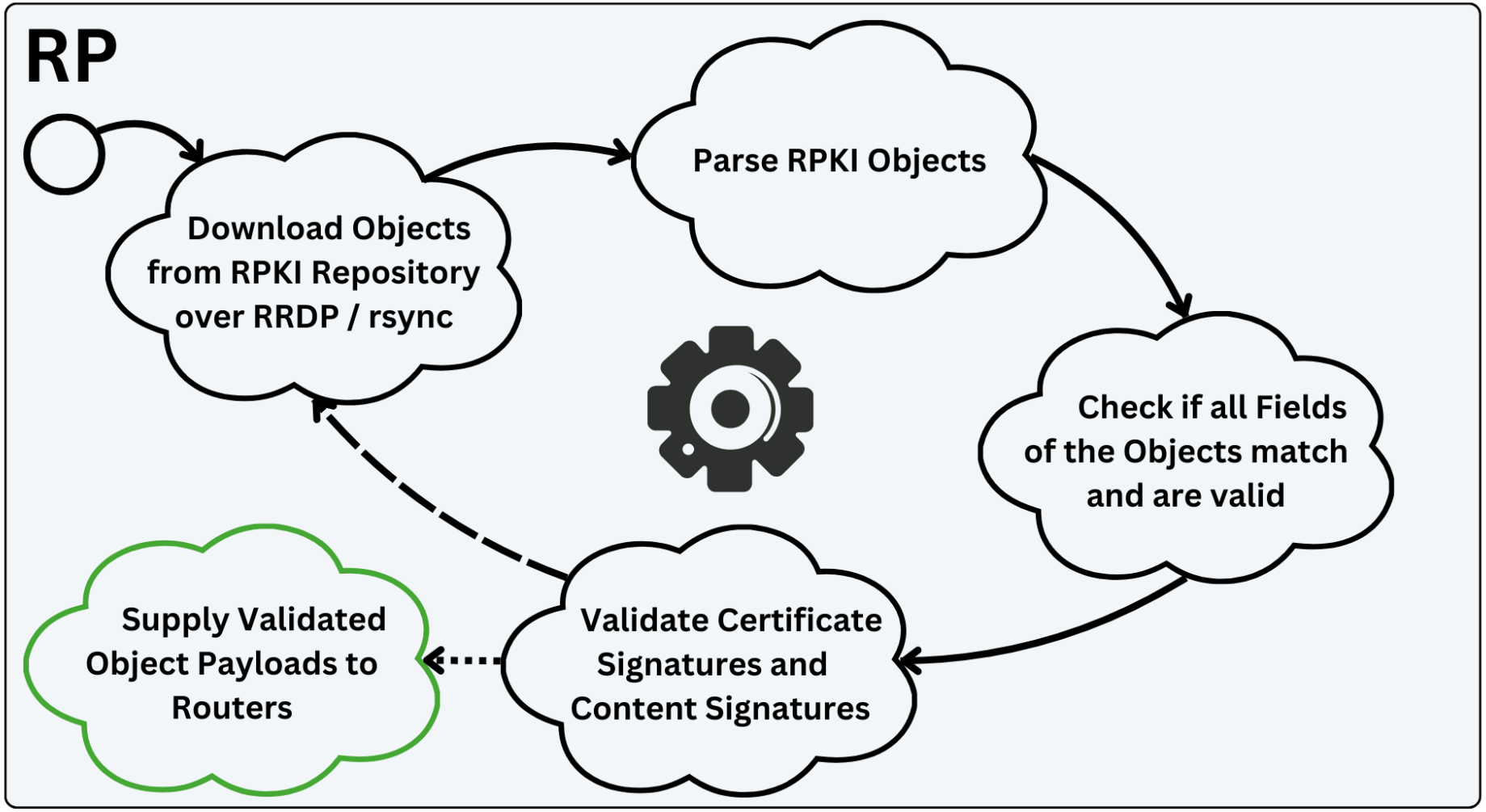
RPKI stores Routing Information and makes it available to Routers

A Short Introduction to RPKI



RPKI stores Routing Information and makes it available to Routers

Relying Parties – A trusted component



RPs are trusted by routers to do all checks and validations

Why fuzzing RPs is hard

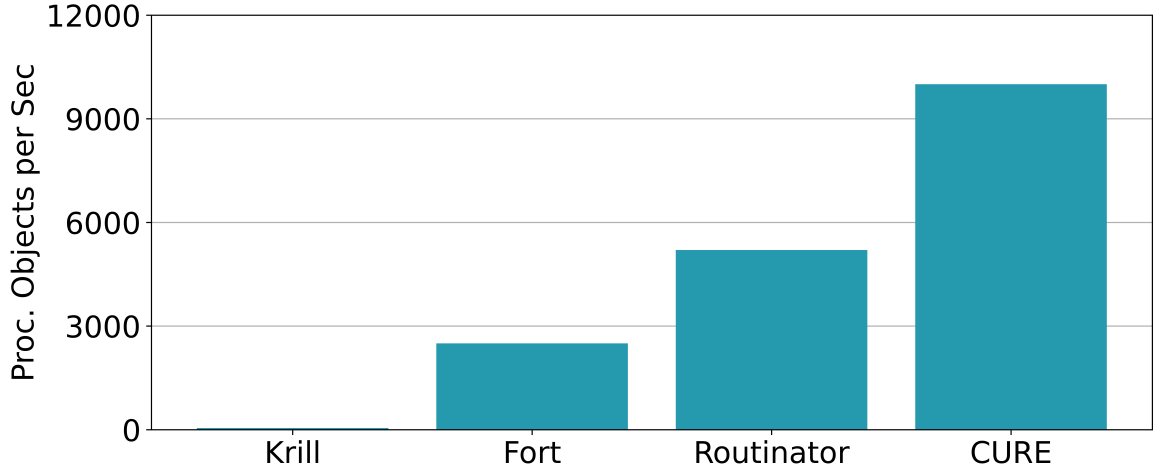
- Fuzzers mutate objects
  - Mutation breaks signatures
- Fuzzers tests one input at a time
  - RPKI Validation involves multiple inputs
- Fuzzers usually work on raw data
  - RPKI Objects are complex and interdependent

=> Fuzzing most RPKI functionality is not possible with traditional fuzzers like AFL++ or LibFuzz

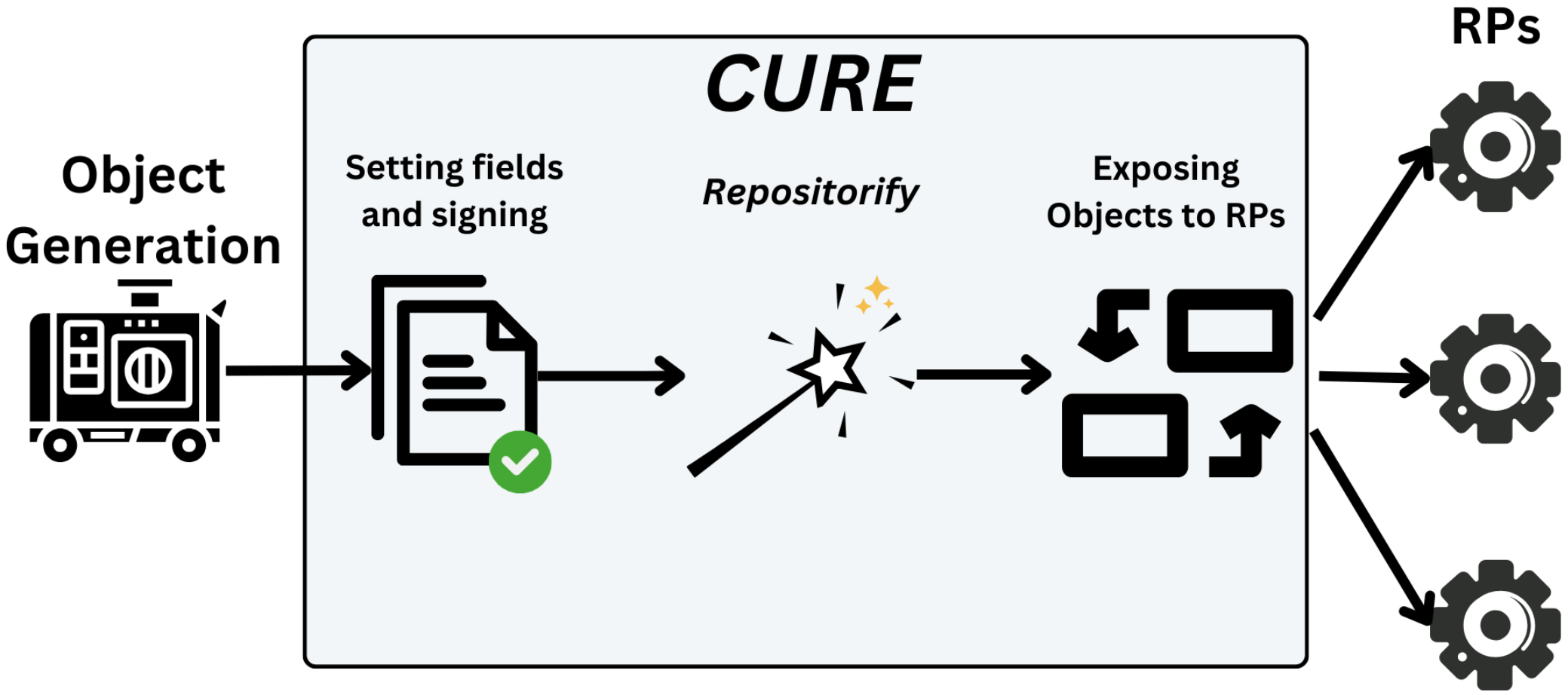
Introducing CURE for RP fuzzing

- Combining fuzzing features with RPKI functionality
- Generate mutated objects, feed them to RPs, look for crashes and inconsistencies (like a fuzzer)
- Sign objects, construct valid RPKI repository around an object (like an RPKI software)

- CURE can create valid RPKI repositories faster than RPs can process them!

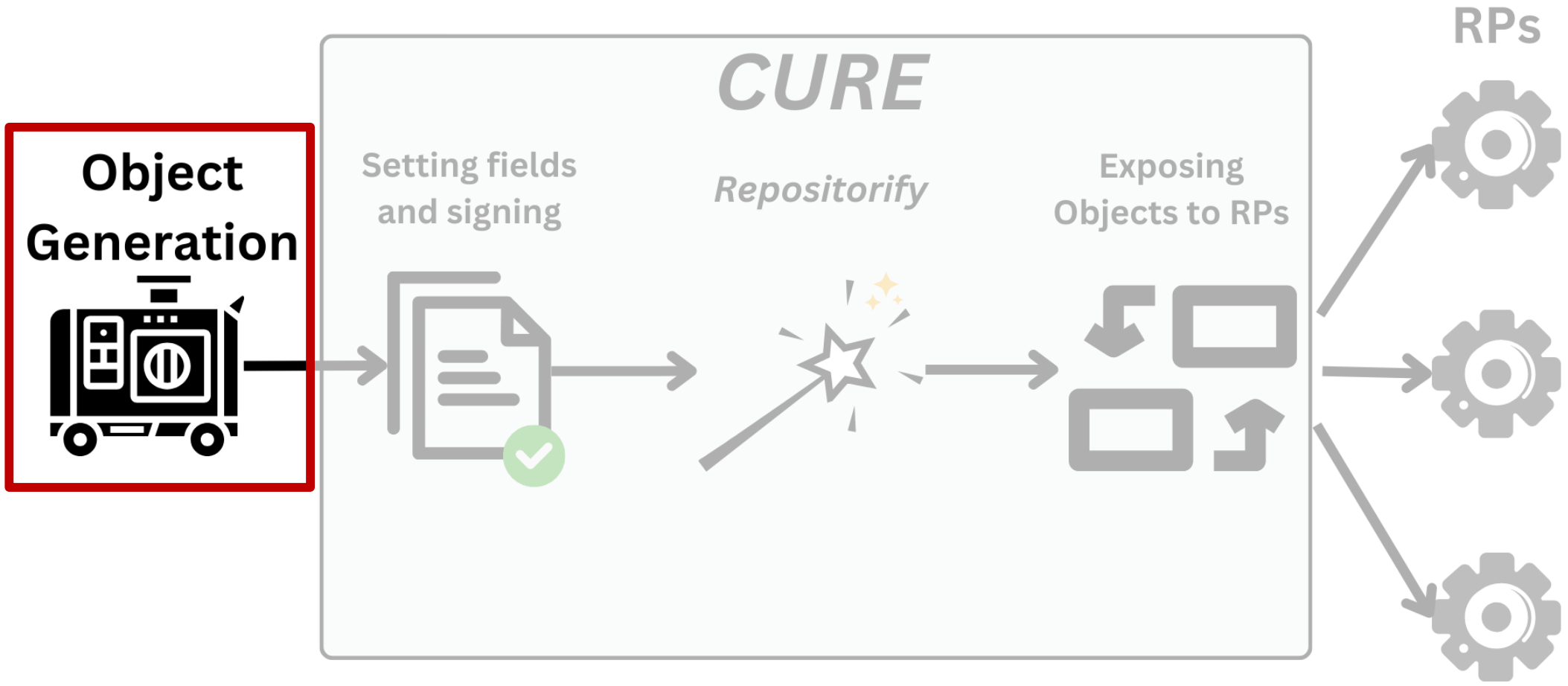


Inner Workings of CURE



CURE can feed arbitrary objects efficiently to the RPs

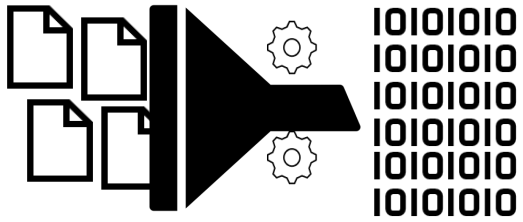
Inner Workings of CURE



CURE can feed arbitrary objects efficiently to the RPs

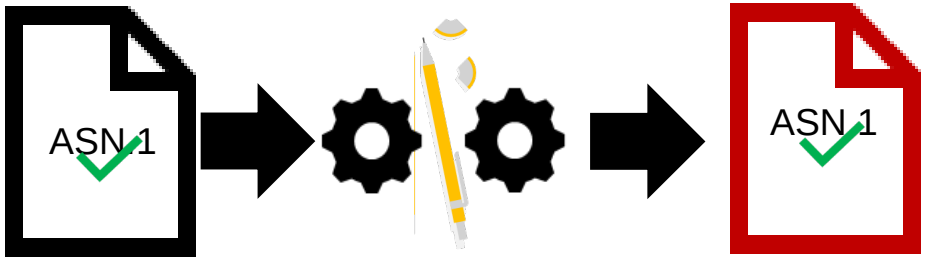
Object Generation in CURE

1. Random Byte Mutation



i. feed the randomizer a set of valid objects
ii. splice file and generate random mutations
iii. targets programming, parsing & schematic errors

2. Structure Aware Mutation



i. schema-abiding and correctly encoded objects
ii. manipulate content of fields to non-conforming types
iii. targets processing and validation logic

CURE supports multiple Object Generation schemes

Results

Vulnerability Overview

- **18 severe vulnerabilities, 5 CVEs, 7 RFC Inconsistencies**

Path Traversal/ Cache Poisoning	DoS from Object Parsing	DoS from Processing	DoS from RTR packet	VRP Inconsistencies
Routinator	Routinator OctoRPKI	Routinator OctoRPKI	Fort	Routinator OctoRPKI Fort RPKI-Client

Vulnerability: Path Traversal/Cache Poisoning

- RPs use object names as storage locations
- Path traversal allows an attacker to place arbitrary files anywhere on the disc of Routinator instances
- Can be exploited e.g. to add malicious trust anchor
  - fully circumvent RPKI validation
  - poison the router VRP cache
- 57.9% affected by Path Traversal
- 32.7% affected by Cache Poisoning

(status: December 2023)

Notification.xml

```
<notification [Header]>
 <snapshot
   uri="https://server.com/data/../../../fake.TAL"
   hash="33f969c5b6fd9ab501f9def2d47f7576ba80
         0a91d09d34a080ed2cf90a86d1ec"
/>
</notification>
```

Vulnerability: DoS

- Crashing the RP eventually leads to routers downgrading RPKI protection
- We found crashes in multiple modules:
  - Parsing of ASN.1 Data
  - Processing of Object Fields
  - Processing of RTR Requests
- Could be exploited by any RPKI repo against ALL active RP instances
- 56% of instances affected by DoS

(status: December 2023)

Routinator.log

```
thread '<unnamed>' panicked at 'index out of bounds:
the len is 2 but the index is 2',
bcder/src/tag.rs:line:column
note: run with `RUST_BACKTRACE=1` environment
variable to display a backtrace
Aborted
```

RFC Inconsistencies

- RP implementations exhibit differences in object processing:
  - RFC non-conforming validation and parsing
  - Undefined non-essential corner cases with critical outcomes

- Related standards: RFC6482, RFC6487, RFC8182, RFC8897, RFC9286

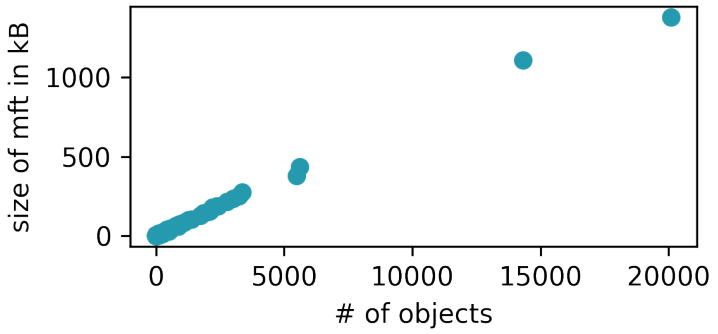
- Example 1: acceptance of non-conforming CRLs with missing fields
  - (risk: certificate integrity)
- Example 2: no concurrency checks for session_id during RRDP
  - (risk: replay attack)

Cache Disparity

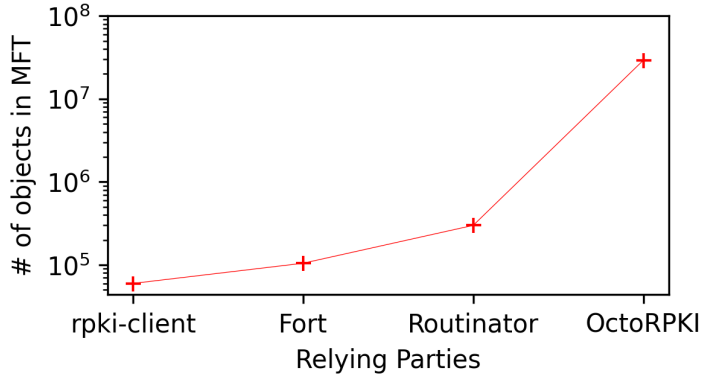
- **Snapshot parsing failure due to object sizes**

RP	ROA / MFT	CRL	CERT	ASPA	GBR
Routinator	20MB	100MB	5MB	20MB	48MB
Octorpki	1.9GB	700MB	5MB	1.9GB	1.9GB
Fort	7MB	10MB	5MB	10MB	10MB
rpki-client	4MB	4MB	5MB	5MB	5MB

TABLE IV: Single file size to crash snapshot.xml parsing.



- **Publication Point DoS**
- **Silent downgrade of VRP coverage**
- **MFT object size threshold**



Inconsistent Validation on the Internet

- Processing inconsistencies are observable in real-world RPKI objects
  - We analyze the RPKI objects with CURE
  - Disclaimer: CURE limitations allow the detection of only a subset of inconsistencies
- Example 1: 6405 Amazon prefixes not processed by Fort due to the presence of OrganisationName instead of SubjectName in certificates
- Example 2: OctoRPKI discards 1744 prefixes for having max length

  > /24 for v4 and > /48 for v6

Fort.log
ERR [Validation]: rsync://my.server.com/data/example1.roa: The 'subject' name has an unknown attribute. (NID: 17)

Conclusion

Conclusions and Observations

- ✓ **RP inconsistencies lead to silent downgrade of RPKI protection**
- ✓ **Availability of fuzzing frameworks is essential**
  - **we offer the Comprehensively Usable RP Evaluator (CURE)**
- ✓ **CURE detected 18 severe vulnerabilities and 7 RFC Inconsistencies**
- ✓ **RPKI deployment is increasing fast, software maturity must outpace it**
- ✓ **Resilience and standardization should be emphasized in RPKI software**

Thank you for your attention!

For any questions, you can contact us at
donika.mirdita@sit.fraunhofer.de
n.vogel@em.uni-frankfurt.de

תודה רבה!

çok teşekkürler

谢谢

Merci beaucoup!

Thank you very much!

Dank je wel!

Vielen Dank!

Muchas gracias

ありがとうございました

Dziękuję!

zor spas

Grazie mille!

اشكرك