SCRAPS: Scalable Collective Remote Attestation for Pub-Sub IoT Networks with Untrusted Proxy Verifier

Lukas Petzi¹, Ala Eddine Ben Yahya¹, Alexandra Dmitrienko¹, Gene Tsudik², Thomas Prantl¹, and Samuel Kounov¹

¹University of Würzburg, Germany
²UC Irvine
Motivation

- E-Health
- Smart Factory
- Smart Home
- Smart City
- Environmental Monitoring

User A
User B
User C

IoT

Broker
Challenges of Attestation in Pub/Sub IoT Networks

- Integrity/Confidentiality/Freshness
- Asynchrony
- Low Memory Capacity
- Computationally Limited
- Heterogeneity
Challenges of Remote Attestation in IoT

- Poor Scalability
- Complex Key Management
- Synchronous Communication
- Uninterrupted availability
- Device Heterogeneity
Collaborative Remote Attestation Schemes

One Verifier – Many Provers

<table>
<thead>
<tr>
<th>Properties</th>
<th>One Verifier – Many Provers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scalability</td>
<td>✓</td>
</tr>
<tr>
<td>On-Demand Attestation</td>
<td>(✓)</td>
</tr>
<tr>
<td>Heterogeneity</td>
<td>(✓)</td>
</tr>
<tr>
<td>Suitable for asynchronous communication</td>
<td>X</td>
</tr>
<tr>
<td>Support for Sleeping Devices</td>
<td>X</td>
</tr>
</tbody>
</table>
Collaborative Remote Attestation Schemes

Many Verifiers – Many Provers

<table>
<thead>
<tr>
<th>Properties</th>
<th>Many Verifiers – Many Provers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scalability</td>
<td>(✔)</td>
</tr>
<tr>
<td>On-Demand Attestation</td>
<td>✔</td>
</tr>
<tr>
<td>Heterogeneity</td>
<td>✔</td>
</tr>
<tr>
<td>Suitable for asynchronous communication</td>
<td>✘</td>
</tr>
<tr>
<td>Support for Sleeping Devices</td>
<td>✘</td>
</tr>
</tbody>
</table>
Hybrid Approach: SCRAPS General Idea

- **Approach:**
  - Combine both approaches

- **ProxyVerifier**
  - Always online – never sleeps
  - Trustless
ProxyVerifier Instantiation: Technical Challenges

1. Smart contracts are passive entities → ProxyVerifier cannot initiate attestation
   Change to self-attestation triggered by IoT platforms

2. Smart contracts are public → Confidentiality of symmetric keys cannot be protected
   Change attestation evidence to use public key cryptography

3. No source of randomness → Random nonce cannot be generated
   Use blockchain height to guarantee freshness
SCRAPS Design

1. Upload Publisher's Configuration
2. Registration
3a. Attestation
3b. Fetch Prover's Configuration
4. Query State

Manufacturer

Manufacturer's Smart Contract

Blockchain/Ledger

ProxyVerifier Smart Contract

Broker

Publisher/Prover

(1b) Deploy attestation scheme

Scraps Design
Evaluation: SCRAPS vs. LegIoT[1]

[1] Neureither et al., LegIoT: Ledgered trust management platform for IoT. In European Symposium on Research in Computer Security (ESORICS), 2020
Evaluation: SCRAPS vs. LegIoT[1]

HitPercentage = \( \frac{\text{QueryHits}}{\text{QueryHits} + \text{QueryMisses}} \times 100 \)

[1] Neureither et al., LegIoT: Ledgered trust management platform for IoT. In European Symposium on Research in Computer Security (ESORICS), 2020
## Conclusion

<table>
<thead>
<tr>
<th>Schemes</th>
<th>One Verifier – Many Provers</th>
<th>Many Verifiers – Many Provers</th>
<th>Hybrid Approach (SCRAPS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scalability</td>
<td>✓</td>
<td>(✓)</td>
<td>✓</td>
</tr>
<tr>
<td>On-Demand Attestation</td>
<td>(✓)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Heterogeneity</td>
<td>(✓)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Suitable for asynchronous communication</td>
<td>X</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Support for Sleeping Devices</td>
<td>X</td>
<td>X</td>
<td>✓</td>
</tr>
</tbody>
</table>

First suitable solution for Pub/Sub Environments