Light-Weight, Delay-Aware and Scalable Authentication for Smart-Grid System

Dr. Attila A. Yavuz, Oregon State University
Presented by Muslum Ozgur Ozmen
Research Need: Fast and Scalable Authentication

• **Critical vulnerabilities for smart-grids:**
  • False data injection attacks
  • Tampering commands
  • Cascade failures

• **Authentication of commands/measurements is vital!**
  • **Real-time:** 60-120 messages per second
  • **Scalable:** Broadcast authentication for large number of components
Research Gap: Lack of Real-time Signatures

- **Symmetric crypto methods:** Unscalable for large distributed systems, lack of non-repudiation and public verifiability.

- **Traditional PKC Signatures:** (e.g., RSA [2], ECDSA [3], and Schnorr [4])
  - High computational cost, they require modular exponentiation (ExpOp) at the signer side.

- **Pre-computation:** Token-ECDSA [5] and online/offline signatures [6,7] do not require ExpOp at the signer side.
  - Linear memory overhead, K items require storing O(K) keys at the signer.

- **One-time/multiple-time Signatures:** (e.g., HORS [8])
  - They are computationally very efficient.
  - Very large signature size (2.5/5 KB) and communication overhead
  - Very large one-time public key (5 KB) for each item to be signed
Our Contribution: A new Real-Time Signature

- Structure-Free Compact Real-Time Authentication (SCRA [1])

- **Generic Design:** Transform any aggregate signature into a fast signing signature.

- **Ultra-Low End-to-End Delay:** SCRA schemes offer the lowest end-to-end delay among their counterparts.
  - SCRA-C-RSA: It is 7 and 19 times faster than ECDSA (pre-computed) and RSA, respectively.

- **Compact Signatures:** The signature size is almost identical to base schemes with all these improved efficiencies.

- **Limitation:** A small constant-size table stored at the signer side (highly feasible even for some embedded devices).
Main Idea: Generic SCRA from Aggregate Signatures

- **Observation**: Signature aggregation is much faster than signature generation.

- Create offline signature components to be combined (aggregated) online!
  - d-bit hash output is split into b-bit L sub-field
  - $\text{Asig}$ is an aggregate digital signature scheme
  - $P$ is a random padding

\[
\begin{array}{cccc}
\text{Field 1 (b-bit)} & \text{Field 2 (b-bit)} & \cdots & \text{Field L (b-bit)} \\
\sigma_{1,0} = \text{Asig}_{sk}(1 || 0 || P) & \sigma_{2,0} = \text{Asig}_{sk}(2 || 0 || P) & \cdots & \sigma_{L,0} = \text{Asig}_{sk}(L || 0 || P) \\
\vdots & \vdots & \ddots & \vdots \\
\sigma_{1,2^{b-1}} = \text{Asig}_{sk}(1 || 2^{b-1} || P) & \sigma_{2,2^{b-1}} = \text{Asig}_{sk}(2 || 2^{b-1} || P) & \cdots & \sigma_{L,2^{b-1}} = \text{Asig}_{sk}(L || 2^{b-1} || P) \\
\end{array}
\]

- Pre-compute signature table $\Gamma$ (offline)

\[
(M || r), |r| = \kappa \text{-bit random number} \quad \xrightarrow{\text{Sign (online)}} \quad (M_1^*, ..., M_L^*) \leftarrow H(M || r) \quad \text{b-bit indexes}
\]

\[
\begin{array}{cccc}
\text{Field 1 (b-bit)} & \text{Field 2 (b-bit)} & \cdots & \text{Field L (b-bit)} \\
\sigma_1' & \sigma_2' & \cdots & \sigma_L' \\
\end{array}
\]

- Fetch corresponding signatures from table $\Gamma$ and aggregate them

\[
s \leftarrow \text{Asig.Agg}(\sigma_1', ..., \sigma_L') \quad \rightarrow \quad \sigma = (s, r)
\]

\[
(M_1^*, ..., M_L^*) \leftarrow H(M || r) \\
\{0,1\} \leftarrow \text{Asig.Vert}(1 || M_1^* || P, ..., L || M_L^* || P), s, PK)
\]
SCRA-C-RSA Instantiation

- C-RSA signature aggregation is just a modular multiplication and verification is very efficient ➔ Overall end-to-end delay is very low!

<table>
<thead>
<tr>
<th>Field 1 (8-bit)</th>
<th>Field 2 (8-bit)</th>
<th>............</th>
<th>Field 32 (8-bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,0</td>
<td>1,255</td>
<td></td>
<td>(1</td>
</tr>
<tr>
<td>H r n</td>
<td>H r n</td>
<td></td>
<td>d</td>
</tr>
<tr>
<td>σ = (s, r)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Pre-compute signature table Γ (offline)

Sign (online)

- Fetch corresponding signatures from table Γ and aggregate them

Verify (online)

- If $s^e = \prod_{j=1}^{32} H(j || M_j^* || P) \mod n$ return 1, else 0.
Performance Comparison (Commodity HW)

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Signing (ms)</th>
<th>Verification (ms)</th>
<th>End-to-End (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECDSA (pre-computed)</td>
<td>0.65</td>
<td>0.82</td>
<td>1.47</td>
</tr>
<tr>
<td>RSA</td>
<td>3.94</td>
<td>0.02</td>
<td>3.96</td>
</tr>
<tr>
<td>BGLS</td>
<td>0.46</td>
<td>34.00</td>
<td>34.46</td>
</tr>
<tr>
<td>NTRU</td>
<td>2.481</td>
<td>0.493</td>
<td>2.974</td>
</tr>
<tr>
<td>SCRA-C-RSA</td>
<td>0.1639</td>
<td>0.0513</td>
<td>0.2152</td>
</tr>
<tr>
<td>SCRA-BGLS</td>
<td>0.0251</td>
<td>34.21</td>
<td>34.2351</td>
</tr>
<tr>
<td>SCRA-NTRU</td>
<td>0.0048</td>
<td>0.507</td>
<td>0.5118</td>
</tr>
</tbody>
</table>

SCRA-C-RSA: Lowest end-to-end delay with mid-size table (2 MB)
SCRA-NTRU: Fastest signing with large-size table (12.33 MB)
SCRA-BGLS: The smallest table with larger delay (160 KB)

• We extended SCRA implementations to GPU setting with our collaborators!
Future Research Directions

• Post-Quantum (PQ) Public Key Infrastructure (PKI) for Smart-Grid System

• There are recently proposed efficient PQ key exchange schemes (e.g., New Hope [11]).

• There is a significant research gap in PQ authentication, especially for resource-limited devices.
  • We will develop new digital signature schemes, and create a practical PQ PKI to protect smart grids.
  • Such a PKI will have broader impact: e-commerce, Bitcoin infrastructure and IoT systems.
References


Funded by the U.S. Department of Energy and the U.S. Department of Homeland Security