A Detailed Report on the Overhead of Hardware APIs for Lightweight Cryptography

Patrick Karl and Michael Tempelmeier

Technical University of Munich
Department of Electrical and Computer Engineering
Chair of Security in Information Technology

October 16, 2020
Exemplary API compliant implementations

<table>
<thead>
<tr>
<th>API</th>
<th>Design</th>
<th>LUT</th>
<th>FF</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAESAR</td>
<td>Ascon128 [1]</td>
<td>1595</td>
<td>818</td>
</tr>
<tr>
<td></td>
<td>SpoC-64 [2]</td>
<td>2136</td>
<td>876</td>
</tr>
<tr>
<td>LWC</td>
<td>Ascon128 [2]</td>
<td>1802</td>
<td>539</td>
</tr>
<tr>
<td></td>
<td>SpoC-64 [2]</td>
<td>1565</td>
<td>728</td>
</tr>
<tr>
<td></td>
<td>Gimli [3]</td>
<td>946</td>
<td>235</td>
</tr>
</tbody>
</table>
Exemplary API compliant implementations

- What do absolute numbers tell us?

<table>
<thead>
<tr>
<th>API</th>
<th>Design</th>
<th>LUT</th>
<th>FF</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAESAR</td>
<td>Ascon128 [1]</td>
<td>1595</td>
<td>818</td>
</tr>
<tr>
<td></td>
<td>SpoC-64 [2]</td>
<td>2136</td>
<td>876</td>
</tr>
<tr>
<td>LWC</td>
<td>Ascon128 [2]</td>
<td>1802</td>
<td>539</td>
</tr>
<tr>
<td></td>
<td>SpoC-64 [2]</td>
<td>1565</td>
<td>728</td>
</tr>
<tr>
<td></td>
<td>Gimli [3]</td>
<td>946</td>
<td>235</td>
</tr>
</tbody>
</table>
Exemplary API compliant implementations

<table>
<thead>
<tr>
<th>API</th>
<th>Design</th>
<th>LUT</th>
<th>FF</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAESAR</td>
<td>Ascon128 [1]</td>
<td>1595</td>
<td>818</td>
</tr>
<tr>
<td></td>
<td>SpoC-64 [2]</td>
<td>2136</td>
<td>876</td>
</tr>
<tr>
<td>LWC</td>
<td>Ascon128 [2]</td>
<td>1802</td>
<td>539</td>
</tr>
<tr>
<td></td>
<td>SpoC-64 [2]</td>
<td>1565</td>
<td>728</td>
</tr>
<tr>
<td></td>
<td>Gimli [3]</td>
<td>946</td>
<td>235</td>
</tr>
</tbody>
</table>

- What do absolute numbers tell us?
- Common API → fair comparison?
Exemplary API compliant implementations

<table>
<thead>
<tr>
<th>API</th>
<th>Design</th>
<th>LUT</th>
<th>FF</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAESAR</td>
<td>Ascon128 [1]</td>
<td>1595</td>
<td>818</td>
</tr>
<tr>
<td></td>
<td>SpoC-64 [2]</td>
<td>2136</td>
<td>876</td>
</tr>
<tr>
<td>LWC</td>
<td>Ascon128 [2]</td>
<td>1802</td>
<td>539</td>
</tr>
<tr>
<td></td>
<td>SpoC-64 [2]</td>
<td>1565</td>
<td>728</td>
</tr>
<tr>
<td></td>
<td>Gimli [3]</td>
<td>946</td>
<td>235</td>
</tr>
</tbody>
</table>

- What do absolute numbers tell us?
- Common API → fair comparison?
- What about different API implementations?
Why?

- "FPGA Benchmarking of Round 2 Candidates..." by GMU [4]

- 24 submissions:
  - 13 using unmodified dev. Package
  - 8 using modified dev. Package
  - 3 not using dev. Package

→ What does that mean for comparison?
API compliant Development Packages

Patrick Karl (TUM) | A Detailed Report on the Overhead of Hardware APIs for Lightweight Cryptography
API compliant Development Packages

- LWC exclusive features:
  1. Hash support
  2. Extended width conversion
  3. Multi-Segment messages
API compliant Development Packages

- LWC exclusive features:
  1. Hash support
  2. Extended width conversion
  3. Multi-Segment messages

→ Benchmark: common interface

→ Pre-/PostProcessor, FIFO included!
Resource Comparison of CAESAR and LWC

- LWC outperforms CAESAR

E.g. 32-bit designs with default configurations
Resource Comparison of CAESAR and LWC

- LWC outperforms CAESAR
- Exception: 8-bit design with minimized FIFO
- Feature cost constant (e.g. hash, multi-segment)
CAESAR FIFO configuration

- Optional tag buffering

- E.g. 32-bit implementation, 128-bit tag $\rightarrow$ 8 entries sufficient
CAESAR FIFO configuration

- Optional tag buffering

- E.g. 32-bit implementation, 128-bit tag $\rightarrow$ 8 entries sufficient

- Default: 1024 entries
Exemplary API compliant implementations

<table>
<thead>
<tr>
<th>API</th>
<th>Design</th>
<th>LUT</th>
<th>FF</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAESAR</td>
<td>Ascon128 [1] ¹</td>
<td>1595</td>
<td>818</td>
</tr>
<tr>
<td></td>
<td>SpoC-64 [2] ²</td>
<td>2136</td>
<td>876</td>
</tr>
<tr>
<td>LWC</td>
<td>Ascon128 [2]</td>
<td>1802</td>
<td>539</td>
</tr>
<tr>
<td></td>
<td>SpoC-64 [2]</td>
<td>1565</td>
<td>728</td>
</tr>
<tr>
<td></td>
<td>Gimli [3]</td>
<td>946</td>
<td>235</td>
</tr>
</tbody>
</table>

¹ HeaderFifo: 4 x 24-bit
² HeaderFifo: 512 x 32-bit

- CryptoCore by different designers

→ Ideally: multiple designs per cipher

- FIFO: 512 vs. 4 entries

→ FIFO dominates

- FIFO: 512 vs. 4 entries
  - FIFO dominates
  
  → Removing FIFO? Critical Path!

- CipherCore difference not that huge

→ FIFO difference (API package)

- Equal assumptions for API implementation

- Equal assumptions for API implementation

→ Improved API implementation + config

→ Fair comparison possible
Conclusion

- Absolute numbers can lead to false impressions
  - Improvement of LWC over CAESAR package
Conclusion

- Absolute numbers can lead to false impressions
  - Improvement of LWC over CAESAR package

- Compare ciphers only
  - Benchmark int. interface, i.e. CryptoCore?
Conclusion

• Absolute numbers can lead to false impressions
  ➢ Improvement of LWC over CAESAR package

• Compare ciphers only
  ➢ Benchmark int. interface, i.e. CryptoCore?

• Ciphers require API
  ➢ Include API implementation?
Thank you for your attention!
References


