ASCON
Submission to the CAESAR Competition

Christoph Dobraniuig, Maria Eichlseder,
Florian Mendel, Martin Schläffer
Our Team

- Christoph Dobraunig
- Maria Eichlseder
- Florian Mendel
- Martin Schläffer
ASCON
Main Design Goals

• Security
• Efficiency
• Simplicity
• Scalability

• Online
• Single pass
• Lightweight
• Side-Channel Robustness
ASCON
General Overview

• Nonce-based AE scheme
• Sponge inspired

<table>
<thead>
<tr>
<th></th>
<th>ASCON-128</th>
<th>ASCON-96</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>128 bits</td>
<td>96 bits</td>
</tr>
<tr>
<td>State size (b)</td>
<td>320 bits</td>
<td>320 bits</td>
</tr>
<tr>
<td>Capacity (c)</td>
<td>256 bits</td>
<td>192 bits</td>
</tr>
<tr>
<td>Rate (r)</td>
<td>64 bits</td>
<td>128 bits</td>
</tr>
</tbody>
</table>
The encryption process is split into four phases:

• Initialization

• Associated Data Processing

• Plaintext Processing

• Finalization
• **Initialization:** updates the 320-bit state with the key $K$ and nonce $N$
ASCON
Associated Data

- **Associated Data Processing**: updating the 320-bit state with associated data blocks $A_i$
• **Plaintext Processing**: inject plaintext blocks $P_i$ in the state and extract ciphertext blocks $C_i$. 

![Diagram of ASCON encryption process](image-url)
Finalization

- **Finalization**: inject the key $K$ and extracts a tag $T$ for authentication
ASCON
Permutation

• SP-Network:

  – S-Layer:

  – P-Layer:
ASCON
Permutation: S-Layer

• Algebraic Degree 2
  – Ease TI (3 shares)

• Branch Number 3
  – Good Diffusion

• Bit-sliced Impl.
ASCON
Permutation: P-Layer

• Branch Number 4

\[
\begin{align*}
\Sigma_0(x_0) &= x_0 \oplus (x_0 \gg 19) \oplus (x_0 \gg 28) \\
\Sigma_1(x_1) &= x_1 \oplus (x_1 \gg 61) \oplus (x_1 \gg 39) \\
\Sigma_2(x_2) &= x_2 \oplus (x_2 \gg 1) \oplus (x_2 \gg 6) \\
\Sigma_3(x_3) &= x_3 \oplus (x_3 \gg 10) \oplus (x_3 \gg 17) \\
\Sigma_4(x_4) &= x_4 \oplus (x_4 \gg 7) \oplus (x_4 \gg 41)
\end{align*}
\]
ASCON

Security Analysis

• Differential Cryptanalysis
  – 5 rounds: > 64 active Sboxes

• Impossible Differential
  – up to 5 rounds

• Linear Cryptanalysis
  – 5 rounds: > 64 active Sboxes
### Differential Cryptanalysis

<table>
<thead>
<tr>
<th>Rounds</th>
<th>Active Sboxes</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>$2^{-2}$</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>$2^{-8}$</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>$2^{-30}$</td>
</tr>
<tr>
<td>4</td>
<td>44</td>
<td>$2^{-88}$</td>
</tr>
<tr>
<td>5</td>
<td>74</td>
<td>$2^{-148}$</td>
</tr>
</tbody>
</table>
ASCON
Security Analysis

• Linear Cryptanalysis

<table>
<thead>
<tr>
<th>Rounds</th>
<th>Active Sboxes</th>
<th>Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>$2^{-2}$</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>$2^{-8}$</td>
</tr>
<tr>
<td>3</td>
<td>13</td>
<td>$2^{-26}$</td>
</tr>
<tr>
<td>4</td>
<td>43</td>
<td>$2^{-86}$</td>
</tr>
<tr>
<td>5</td>
<td>70</td>
<td>$2^{-140}$</td>
</tr>
</tbody>
</table>
ASCON
Implementation/Performance

• Software
  – Intel Core2 Duo
  – ARM Cortex-A8

• Hardware
  – High-speed
  – Low-area
ASCON
Software Implementation

• Intel Core2 Duo

<table>
<thead>
<tr>
<th></th>
<th>64</th>
<th>512</th>
<th>1024</th>
<th>4096</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ASCON-128</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(cycles/byte)</td>
<td>22.0</td>
<td>15.9</td>
<td>15.6</td>
<td><strong>15.2</strong></td>
</tr>
<tr>
<td><strong>ASCON-96</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(cycles/byte)</td>
<td>17.7</td>
<td>11.0</td>
<td>10.5</td>
<td><strong>10.3</strong></td>
</tr>
</tbody>
</table>
## ASCON

### Hardware Implementation

- **ASCON-128**

<table>
<thead>
<tr>
<th></th>
<th>Variant 1</th>
<th>Variant 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Area (kGE)</strong></td>
<td>8.9</td>
<td>4</td>
</tr>
<tr>
<td><strong>Throughput (MByte/s)</strong></td>
<td>400</td>
<td>1</td>
</tr>
</tbody>
</table>

H. Gross, E. Wenger

Threshold implementation coming soon!
ASCON
Choice of Parameters

• Now: \((c,r) = (256, 64)\)
  – Conservative choice

• Proposed: \((c,r) = (192, 128)\) [BDPV12]
  – Significant speedup (factor 2)
  – Limit on data complexity \(2^{64}\)

• Proposed: \((c,r) = (128, 192)\) [JLM14]
  – Significant speedup (factor 3)
  – More analysis needed
Ascon

Interesting Links

Ascon Resources

- Specification [v1.0]
- Submission document [v1.0]
- GitHub repositories with implementations [git collection]
  - C (reference / optimized) [git] [zip]
  - Python [git] [py]
  - Java [git] [zip]
  - Hardware [git] [zip]
Thank you!

http://ascon.iaik.tugraz.at