On the Construction of Hardware-friendly $4 \times 4$ and $5 \times 5$ S-boxes

Stjepan Picek  Bohan Yang  Vladimir Rozic  Nele Mentens

KU Leuven ESAT/COSIC and iMinds, Kasteelpark Arenberg 10, B-3001
Leuven-Heverlee, Belgium

28th October 2016
Outline

1 Introduction
   - Introduction
   - Motivation
2 Background
   - Properties of Interest
   - Search space size
3 Methodology and Results
   - Random Search
   - Heuristics
4 Results
   - Power and Area Results
   - $4 \times 4$ Size
   - $5 \times 5$ Size
5 Discussion
Lightweight cryptography

- Besides the security, in lightweight cryptography we are also interested to make the ciphers as small and as efficient as possible.
- Therefore, some of goals are area-efficient ciphers, energy-efficient ciphers, and latency-efficient ciphers.
- Furthermore, large part of ciphers is realized as SPN structures.
- IN SPNs, the S-box part is often the most expensive one!
We concentrate on $4 \times 4$ and $5 \times 5$ sizes of S-boxes.

We explore how to obtain S-boxes that are hardware-friendly but have optimal cryptographic properties.

Search space is too large to conduct an exhaustive search when accounting the cost of the evaluation part.

Therefore, we explore how efficient is heuristics for such a task.
Cryptographic Properties

- Bijectivity.
- Nonlinearity.
- $\delta$-uniformity.
- Algebraic degree.
- Number of fixed points.
Static and Dynamic Power

- Dynamic power consumption originates from the switching activity of the circuit.
- Static power consumption is caused by subthreshold currents and gate leakage.
- Static power consumption is constant over time and does not depend on the clock frequency or the switching activity.
- In older technology nodes the dynamic power consumption was dominant in the total power consumption and the static power consumption was negligible.
- With smaller technology nodes, the relative contribution of the static leakage power consumption has increased.
4 × 4 Size

- Around $2^{44}$ bijective S-boxes.
- 16 optimal, not affine-equivalent classes.
- We are interested only in optimal S-boxes

$$S_a(x) = B(S_b(A(x) \oplus \vec{a})) \oplus \vec{b}, \quad (1)$$

- In general, it is possible to conduct exhaustive search (when considering only cryptographic properties).
- There are also other classifications.
5 × 5 Size

- Impossible to conduct exhaustive search.
- We know only classification of APN functions of dimension 5.
- Here, S-boxes with suboptimal cryptographic properties are also used.
Simulation Setup

- Clock frequency of 10 MHz and NANGATE 45 open cell library.
- Generate S-boxes in a lookup table style.
- Matlab script to generate the HDL description of the S-box.
- Synopsys Design Compiler to produce the gate-level netlist and the delay file.
- Test-bench that goes through all possible $n \times (n - 1)$ input transitions of the S-box.
- Modelsim SE PLUS 6.6d to simulate the wave file containing the switching activity of all nodes.
- Design Compiler is used to estimate the power consumption.
Power/area evaluation

Figure: Simulation setup for the generation/evaluation of S-boxes.
Random Search

- Create random S-boxes as permutations of values between 0 and $2^n - 1$ and check the results in terms of area and power.
- When evaluating only the optimal S-boxes, our results show that the power consumption is higher than 550 nW.
- In terms of area, the optimal S-boxes obtained through random search have an area larger than 20GE.
(a) Area results for randomly chosen $4 \times 4$ S-boxes.

(b) Area results for randomly chosen $4 \times 4$ S-boxes.

**Figure:** 1000 random and optimal $4 \times 4$ S-boxes
Genetic algorithms

- Metaheuristics intended to provide good results on a wide range of problems.
- Black-box technique.
- Natural representation for S-boxes is permutation-based.
- Natural representation for invertible matrices and constants is bitstring-based.
Pseudocode for population-based metaheuristics

1: Input: Parameters of the algorithm
2: Output: Optimal solution set
3: \( t \leftarrow 0 \)
4: \( P(0) \leftarrow \text{CreateInitialPopulation} \)
5: while TerminationCriterion do
6: \( t \leftarrow t + 1 \)
7: \( P'(t) \leftarrow \text{SelectMechanism}(P(t - 1)) \)
8: \( P(t) \leftarrow \text{VariationOperators}(P'(t)) \)
9: end while
10: Return OptimalSolutionSet(P)
Objective Functions

- Maximize the value of:

\[ \text{fitness} = N_F + (2^m - \delta) + (2^m - nr\_fixed\_points). \]  \hspace{1cm} (2)

- Two-stage evaluation.
Objective Functions

- Maximize the value of:

\[
fitness = N_F + (2^m - \delta) + (2^m - nr\_fixed\_points).
\]  

- Two-stage evaluation.
- Only optimal S-boxes are evaluated for power/area.
Objective Functions

- Maximize the value of:

$$fitness = N_F + (2^m - \delta) + (2^m - nr\_fixed\_points).$$  \hspace{1cm} (2)

- Two-stage evaluation.

- Only optimal S-boxes are evaluated for power/area.

- The whole population is sent at the same time for evaluation.
### 4 × 4 Size

**Table: Best evolved 4 × 4 S-boxes**

<table>
<thead>
<tr>
<th>Area results</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>With fixed points</td>
<td>13, 1, 3, 11, 12, 2, 7, 10, 0, 5, 8, 9, 4, 6, 14, 15</td>
</tr>
<tr>
<td>Area:</td>
<td>14.33GE</td>
</tr>
<tr>
<td>Without fixed points</td>
<td>7, 10, 11, 8, 5, 3, 1, 9, 6, 2, 15, 0, 4, 14, 12, 13</td>
</tr>
<tr>
<td>Area:</td>
<td>13.33GE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Power results</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>With fixed points</td>
<td>3, 1, 2, 10, 14, 5, 7, 15, 4, 6, 0, 11, 13, 12, 8, 9</td>
</tr>
<tr>
<td>Dynamic Power:</td>
<td>237.16nW</td>
</tr>
<tr>
<td>Leakage Power:</td>
<td>297.52nW</td>
</tr>
<tr>
<td>Area:</td>
<td>14.67GE</td>
</tr>
<tr>
<td>Without fixed points</td>
<td>13, 5, 10, 4, 7, 1, 2, 0, 14, 6, 8, 12, 15, 3, 9, 11</td>
</tr>
<tr>
<td>Dynamic Power:</td>
<td>206.1nW</td>
</tr>
<tr>
<td>Leakage Power:</td>
<td>240.73nW</td>
</tr>
<tr>
<td>Area:</td>
<td>12.67GE</td>
</tr>
</tbody>
</table>
Involutional S-boxes

- 2,027,025 involutive S-boxes with 0 fixed points, 16,216,200 with 2 fixed points, 18,918,900 with 4 fixed points, 7,567,570 with 6 fixed points.
- Out of that, there are ≈ 3,000,000 optimal S-boxes with various number of fixed points.
Involutve S-boxes

- 2,027,025 involutive S-boxes with 0 fixed points, 16,216,200 with 2 fixed points, 18,918,900 with 4 fixed points, 7,567,570 with 6 fixed points.
- Out of that, there are \( \approx 3,000,000 \) optimal S-boxes with various number of fixed points.
- For S-boxes with 4 fixed points the best result for area is 13GE while the best result for power is an S-box with a dynamic power of 201.8418\(nW\) and a leakage power of 255.1868\(nW\).
Involutive S-boxes

- 2027025 involutive S-boxes with 0 fixed points, 16216200 with 2 fixed points, 18918900 with 4 fixed points, 7567570 with 6 fixed points.
- Out of that, there are ≈ 3,000,000 optimal S-boxes with various number of fixed points.
- For S-boxes with 4 fixed points the best result for area is $13GE$ while the best result for power is an S-box with a dynamic power of $201.8418\, nW$ and a leakage power of $255.1868\, nW$.
- For optimal involutive S-boxes with 6 fixed points the best result for area is $15GE$ and the best result for power is an S-box with a dynamic power of $223.3748\, nW$ and a leakage power of $293.5608\, nW$. 

Stjepan Picek, Bohan Yang, Vladimir Rozic, Nele Mentens
Table: Best evolved $5 \times 5$ S-boxes

<table>
<thead>
<tr>
<th></th>
<th>Area results</th>
<th>Power results</th>
</tr>
</thead>
<tbody>
<tr>
<td>With fixed points</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area:</td>
<td>39.33GE</td>
<td>Dynamic Power: 801.8934nW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Leakage Power: 777.7131nW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Area: 39.67GE</td>
</tr>
<tr>
<td>Without fixed p.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Area:</td>
<td>38GE</td>
<td>Dynamic Power: 734.7164nW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Leakage Power: 754.2006nW</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Area: 39.33GE</td>
</tr>
</tbody>
</table>
Affine Transformation Results

Table: Best evolved 5 × 5 S-boxes, affine transformations

<table>
<thead>
<tr>
<th></th>
<th>Dynamic Power:</th>
<th>Leakage Power:</th>
<th>Area:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keccak</td>
<td>488.6914nW</td>
<td>496.4189nW</td>
<td>26GE</td>
</tr>
<tr>
<td>PRIMATEs</td>
<td>751.4109nW</td>
<td>723.7496nW</td>
<td>37GE</td>
</tr>
<tr>
<td>APN S-box</td>
<td>913.5057nW</td>
<td>942.5685nW</td>
<td>48GE</td>
</tr>
</tbody>
</table>
Discussion

- For $4 \times 4$ size, the results are very good.
- For $5 \times 5$ size, the results are somewhat worse when looking for S-boxes, but again good when investigating affine transformations.
Discussion

- For $4 \times 4$ size, the results are very good.
- For $5 \times 5$ size, the results are somewhat worse when looking for S-boxes, but again good when investigating affine transformations.
- Difficult to compare with other approaches.
- LUT based S-box are maybe not the best way how to represent/implement, but as far as we are aware, they are the most fair approach.
- Furthermore, synthesis tools do translate LUTs into combinatorial circuits, only not optimal ones.
Discussion

- Alternatively, we could do a two-stage search where we look for good S-boxes but then also for good representations of those S-boxes.
- The role of fixed points is not completely clear at this point.
- Our approach is scalable over technologies and properties of interest.
Discussion

- Alternatively, we could do a two-stage search where we look for good S-boxes but then also for good representations of those S-boxes.
- The role of fixed points is not completely clear at this point.
- Our approach is scalable over technologies and properties of interest.
- Interesting approach would be to evolve only involutive S-boxes.
Discussion

- Alternatively, we could do a two-stage search where we look for good S-boxes but then also for good representations of those S-boxes.
- The role of fixed points is not completely clear at this point.
- Our approach is scalable over technologies and properties of interest.
- Interesting approach would be to evolve only involutive S-boxes.
- Currently, we are running an exhaustive search on optimal involutive S-boxes, but for now we obtained no better results than with GA (we found S-box with the same area, but worse power consumption).
Conclusions

- Heuristics is able to find S-boxes with good cryptographic properties that are also small/power efficient.
Conclusions

- Heuristics is able to find S-boxes with good cryptographic properties that are also small/power efficient.
- We give two techniques; either evolving S-boxes directly or evolving affine transformations.
Heuristics is able to find S-boxes with good cryptographic properties that are also small/power efficient.

We give two techniques; either evolving S-boxes directly or evolving affine transformations.

Results obtained seem to be able to compete with any other technique, but to obtain them we need only a fraction of time.
Conclusions

- Heuristics is able to find S-boxes with good cryptographic properties that are also small/power efficient.
- We give two techniques; either evolving S-boxes directly or evolving affine transformations.
- Results obtained seem to be able to compete with any other technique, but to obtain them we need only a fraction of time.
- Naturally, our results are as good as the synthesis tools allow.
Conclusions

- Heuristics is able to find S-boxes with good cryptographic properties that are also small/power efficient.
- We give two techniques; either evolving S-boxes directly or evolving affine transformations.
- Results obtained seem to be able to compete with any other technique, but to obtain them we need only a fraction of time.
- Naturally, our results are as good as the synthesis tools allow.
- Good results in naive combinatorial circuit implementation can only improve with smarter implementations, while vice versa does not necessarily hold.
Thank You for Your attention.
Questions?