3-Share Threshold Implementation of AES S-box without Fresh Randomness

Takeshi Sugawara
The University of Electro-Communications, Japan
University of Michigan, US

This work is funded by JSPS KAKENHI Grant Number 17H06681 and JP18H05289
Overview

Methodology

Threshold implementation (Nicova et al., ICICS2006)

Changing of the guards (Daemen, CHES2017)

Generalized Changing of the guards (This work)

Implementation

Difficulty in realizing 3-share + Uniform TI for AES and Keccak for 10+ years

3-Share + Uniform Keccak S-box (Daemen, CHES2017)

4-Share + Uniform AES S-box (Wegener & Moradi, COSADE2018)

3-Share + Uniform AES S-box (This work)
TI: Threshold Implementation

- Implement crypto while keeping shared representation of intermediate variables

**Input share** \((x_a, x_b, x_c)\):
\[x_a \oplus x_b \oplus x_c = x\]

Sharing \(\{\psi_a, \psi_b, \psi_c\}\) maps a share to another share

**Correctness:** 
\(\{\psi_a, \psi_b, \psi_c\}\) gives the correct result

**Non-completeness:**
Each map uses only a proper subset

**Output share** \((X_a, X_b, X_c)\):
\[X_a \oplus X_b \oplus X_c = X\]
Uniformity

• **Uniformity about shares**
  - For each (raw) value, all the possible shares should appear equally
  - Necessary for security against statistical attack

• **Uniformity about sharing**
  - The sharing preserves the uniformity about shares:
    
    Input share is uniform  \(\Rightarrow\) output share is uniform

---

Example:
3-share of 1-bit variable

<table>
<thead>
<tr>
<th>Raw value</th>
<th>Share</th>
<th>Prob.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(0,0,0)</td>
<td>1/16</td>
</tr>
<tr>
<td>0</td>
<td>(0,1,1)</td>
<td>1/16</td>
</tr>
<tr>
<td>0</td>
<td>(1,0,1)</td>
<td>1/16</td>
</tr>
<tr>
<td>0</td>
<td>(1,1,0)</td>
<td>1/16</td>
</tr>
<tr>
<td>1</td>
<td>(0,0,1)</td>
<td>3/16</td>
</tr>
<tr>
<td>1</td>
<td>(0,1,0)</td>
<td>3/16</td>
</tr>
<tr>
<td>1</td>
<td>(1,0,0)</td>
<td>3/16</td>
</tr>
<tr>
<td>1</td>
<td>(1,1,1)</td>
<td>3/16</td>
</tr>
</tbody>
</table>
Uniformity is difficult to satisfy

- There had been no 3-share + uniform sharing for Keccak and AES S-boxes until 2017

- If no uniformity, we should add fresh randomness to make the output share uniform again
  - 1—10 Kbits/AES
  - 10—50 bits/cycle
CotG: Changing of the Guards (Daemen, CHES2017)

• Using a neighboring input share for (pseudo) remasking

• Applicable to bijective mapping
  • Succeeded in making 3-share + uniform Keccak S-box

---

### Diagram

```
x^1_a  x^1_b  x^1_c  
S_a   S_b   S_c

x^2_a  x^2_b  x^2_c  
S_a   S_b   S_c

x^3_a  x^3_b  x^3_c  
S_a   S_b   S_c
```

```
x^0_c
x^0_b  
```

```
x^1_a  x^1_b  x^1_c  

x^2_a  x^2_b  x^2_c  

x^3_a  x^3_b  x^3_c  
```

```
x^0_c
```

```
x^0_b
```
Why we can’t use CotG for 3-share AES S-box

• We need to decompose S-box to reduce the number of shares, and we get **multiplications that are not bijective**

Canright’s S-box implementation
Basic idea toward generalization

• Transform the target mapping $\psi$ into an equivalent mapping $\psi^R$ that has a uniform sharing

\[
\begin{align*}
\psi & \quad \longrightarrow \quad \times \quad \longrightarrow \quad \{\psi_a, \psi_b, \psi_c\} \\
\psi^R & \quad \longrightarrow \quad \{\psi^R_a, \psi^R_b, \psi^R_c\}
\end{align*}
\]
Expansion

- Transforming the target $\psi$ into a bijective mapping $\psi^E$ using the (unbalanced) Feistel network.
Expansion cont.

- $\psi^E$ always has a uniform sharing $\{\psi_a^E, \psi_b^E, \psi_c^E\}$
  - $\therefore$ The sharing is bijective because the Feistel structure is preserved
  - $\therefore$ A sharing is bijective $\Rightarrow$ the sharing is uniform

\[
\begin{align*}
\psi^E & \quad \{\psi(x) \oplus y \quad x\} \\
\psi & \quad \text{is a non-uniform sharing of } \psi
\end{align*}
\]
Expansion is not enough

• Feeding $\psi^E(x)$ to CotG does not make a lot of sense since it outputs $\psi(x) \oplus y$ instead of $\psi(x)$

• $y$ should be 0 and we need to get it from somewhere
Restriction

• Converting the unnecessary output to zero
• Feeding it to a neighboring mapping as a zero input
Restriction cont.

• The null mapping $\bot$ has a uniform sharing
  
• $\{x_a, x_b, x_c\} \mapsto \{x_b \oplus x_c, x_b, x_c, \}$

Converting unnecessary share to another one representing 0
Chaining

• For a target map having the same input and output sizes \((m = n)\), we can easily chain zero outputs and inputs.

• The right figure shows 3-parallel mapping given by

\[
(0, x^1, x^2, x^3) 
\mapsto (\psi(x^1), \psi(x^2), \psi(x^3), 0)
\]
Chaining cont.

• By substituting each $\psi^R$ with its sharing, we get a uniform sharing of a layer of parallel $\psi^R$s.
Why it is a generalization of CotG

• This sharing is the same as Daemen’s CotG
• Now we can also support non-bijection mapping
A map with different input/output sizes

- Input is larger: we get additional zero outputs that we can use later
- Output is larger: we need additional zero inputs

Additional inputs for the Changing of the Guards

Additional outputs
Application to AES S-box

- 4-stage Canright’s S-box is expanded to make all the stages uniform
  - + **6-bit** additional input
  - + **6-bit** additional output

- Register overhead
  \[ \approx \text{Initial randomness:} \]
  - **6 bits** * 3 shares * 16 S-boxes
    \[ = 288 \text{ bits} + \text{some more} \]
Conclusion

• A generalization of the Changing of the Guards that supports non-bijective targets

• The first 3-share and uniform threshold implementation of the AES S-box