ShiftRows Alternatives for AES-like Ciphers and Optimal Cell Permutations for Midori and Skinny

Gianira N. Alfarano\textsuperscript{1}, Christof Beierle\textsuperscript{2}, Takanori Isobe\textsuperscript{3}, Stefan Köbl\textsuperscript{4}, Gregor Leander\textsuperscript{2}

March 25th, 2019

\textsuperscript{1}University of Zurich, Switzerland
\textsuperscript{2}Horst Görtz Institute for IT Security, Ruhr-Universität Bochum, Germany
\textsuperscript{3}University of Hyogo, Japan
\textsuperscript{4}Cybercrypt, Denmark
AES-like Constructions are very popular

- **Block Ciphers:**
  - Deoxys-BC, Kuzneychik, LED, Midori, Prince, Skinny, ...

- **Hash Functions:**
  - Grøstl, Photon, Streebog, Whirlpool, ...

- **Permutations:**
  - AESQ, Haraka, Prøst, Simpira, ...
Building blocks:

- **SB**: S S S S S
- **$P_p$**: Arrows indicating the mixing process
- **$Mix_M$**: Swirls indicating the mixing process
AES-like Primitives

Building blocks:

- Apply S-box on each cell
- Only non-linear component
- Vast area of research
AES-like Primitives

Building blocks:

- Multiply each column with matrix
- Vast area of research

$Mix_M$
AES-like Primitives

Building blocks:

- Apply S-box on each cell
- Only non-linear component
- Vast area of research
- Multiply each column with matrix
- Vast area of research
Security of AES-like Primitives

Resistance against differential and linear cryptanalysis.

- S-box: Every *active* S-box has an effect on probability of differential trail.
- Mix: Gives a lower bound on active S-boxes in one round.
- Permute: Heavily influences bounds for multiple rounds.

Goal

Find a lower bound on the number of active S-boxes for a design.
Example AES

- MixColumns has *branch number* 5.
- Only constraint active input + output ≥ 5.
Security of AES-like Primitives

Example AES

- MixColumns has *branch number* 5.
- Only constraint active input + output $\geq 5$. 

![Diagram of AES operations]

SB SR MC SB SR MC
Example AES

- MixColumns has *branch number* 5.
- Only constraint active input + output ≥ 5.
Example AES

- MixColumns has *branch number* 5.
- Only constraint active input + output $\geq 5$. 
Example AES

- MixColumns has *branch number* 5.
- Only constraint active input + output ≥ 5.
Example AES

- MixColumns has branch number 5.
- Only constraint active input + output ≥ 5.
Security of AES-like Primitives

Can be much more complex for other choices:

- Midori (Branch number 4)
- but not possible to have $2 \rightarrow 3$ (or $3 \rightarrow 2$) transitions.
- Skinny (Branch number 2)
Known results on the permute layer

- \( M \) is MDS and \( n \times n \) state \( \rightarrow \) AES ShiftRows optimal
- *Linear Frameworks for Block Ciphers*, Daemen, Knudsen, Rijmen, DCC, 2001
AES-like Primitives

Known results on the permute layer

- $M$ is MDS and $n \times n$ state $\rightarrow$ AES ShiftRows optimal
- *Linear Frameworks for Block Ciphers*, Daemen, Knudsen, Rijmen, DCC, 2001

Problem we solve

Given an $n \times m$ state of $w$-bit words with a fixed SB and Mix layer. What is the optimal choice for permute w.r.t. security against differential/linear cryptanalysis?
How can we find the optimal choice for $p$?

- For a $4 \times 4$ state we already get $2^{44.25}$ choices.
- Need to evaluate cryptanalytical properties for all of them?
- How can we limit the search space?
First observation:

- Consider permutation $p$ and $\vartheta$.
- If $\text{Mix}_M \circ \text{Permute}_{\vartheta} = \text{Permute}_{\vartheta} \circ \text{Mix}_M$...
- ...then $\text{Permute}_p$ and $\text{Permute}_{\vartheta \circ \vartheta^{-1}}$ have the same cryptographic properties.
First observation:

- Consider permutation \( p \) and \( \psi \).
- If \( \text{Mix}_M \circ \text{Permute}_{\psi} = \text{Permute}_{\psi} \circ \text{Mix}_M \ldots \)
- ...then \( \text{Permute}_p \) and \( \text{Permute}_{\psi \circ \psi^{-1}} \) have the same cryptographic properties.
First observation:

- Consider permutation \( p \) and \( \vartheta \).
- If \( \text{Mix}_M \circ \text{Permute}_\vartheta = \text{Permute}_\vartheta \circ \text{Mix}_M \)...
- ...then \( \text{Permute}_p \) and \( \text{Permute}_{\vartheta \circ \vartheta^{-1}} \) have the same cryptographic properties.
Equivalence Relation:

- Two permutations $p, p'$ are $\mathbf{M}$-equivalent if there exists $\vartheta$ such that
  \[ p' = \vartheta \circ p \circ \vartheta^{-1}, \] \hspace{1cm} (1)
  and $\vartheta$ commutes with $\mathbf{M}$.

- $\mathbf{M}$-equivalent permutations will have same number of active S-boxes!

- Unclear how to efficiently determine $\mathbf{M}$-equivalence.
weak $\textbf{M}$-equivalence:

- $\theta = \pi \circ \phi$
- $\pi$ permutes whole columns of the state
- $\phi$ permutes inside columns individually
Classifying Cell Permutations

Structure matrix

Example

\[
\begin{bmatrix}
0 & 4 & 8 & 12 \\
1 & 5 & 9 & 13 \\
2 & 6 & 10 & 14 \\
3 & 7 & 11 & 15
\end{bmatrix}
\xrightleftharpoons{P}
\begin{bmatrix}
4 & 0 & 13 & 1 \\
5 & 6 & 14 & 2 \\
11 & 9 & 8 & 3 \\
15 & 12 & 7 & 10
\end{bmatrix}
, \quad A_P = \begin{pmatrix}
0 & 1 & 0 & 3 \\
2 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 \\
1 & 1 & 2 & 0
\end{pmatrix}
\]
Classifying Cell Permutations

Structure matrix

Example

\[
\begin{bmatrix}
0 & 4 & 8 & 12 \\
1 & 5 & 9 & 13 \\
2 & 6 & 10 & 14 \\
3 & 7 & 11 & 15 \\
\end{bmatrix}
\xrightarrow{P}
\begin{bmatrix}
4 & 0 & 13 & 1 \\
5 & 6 & 14 & 2 \\
11 & 9 & 8 & 3 \\
15 & 12 & 7 & 10 \\
\end{bmatrix}
\]

, 

\[
A_P = \begin{pmatrix}
0 & 1 & 0 & 3 \\
2 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 \\
1 & 1 & 2 & 0 \\
\end{pmatrix}
\]
Classifying Cell Permutations

Structure matrix

Example

\[
\begin{bmatrix}
0 & 4 & 8 & 12 \\
1 & 5 & 9 & 13 \\
2 & 6 & 10 & 14 \\
3 & 7 & 11 & 15 \\
\end{bmatrix}
\xrightarrow{P}
\begin{bmatrix}
4 & 0 & 13 & 1 \\
5 & 6 & 14 & 2 \\
11 & 9 & 8 & 3 \\
15 & 12 & 7 & 10 \\
\end{bmatrix}
\Rightarrow
\begin{pmatrix}
0 & 1 & 0 & 3 \\
2 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 \\
1 & 1 & 2 & 0 \\
\end{pmatrix}
\]
Classifying Cell Permutations

Structure matrix

<table>
<thead>
<tr>
<th>0</th>
<th>4</th>
<th>8</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>9</td>
<td>13</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>10</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>11</td>
<td>15</td>
</tr>
</tbody>
</table>

\[ \begin{pmatrix} 0 & 4 & 8 & 12 \\ 1 & 5 & 9 & 13 \\ 2 & 6 & 10 & 14 \\ 3 & 7 & 11 & 15 \end{pmatrix} \xrightarrow{P} \begin{pmatrix} 4 & 0 & 13 & 1 \\ 5 & 6 & 14 & 2 \\ 11 & 9 & 8 & 3 \\ 15 & 12 & 7 & 10 \end{pmatrix} \]

\[ A_P = \begin{pmatrix} 0 & 1 & 0 & 3 \\ 2 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 0 \end{pmatrix} \]
### Classifying Cell Permutations

**Structure matrix**

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>4</th>
<th>8</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>1</td>
<td>5</td>
<td>9</td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>6</td>
<td>10</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>7</td>
<td>11</td>
<td>15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>4</th>
<th>0</th>
<th>13</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>5</td>
<td>6</td>
<td>14</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>9</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>12</td>
<td>7</td>
<td>10</td>
</tr>
</tbody>
</table>

, $A_p = \begin{pmatrix} 0 & 1 & 0 & 3 \\ 2 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 0 \end{pmatrix}$
### Result

We provide an efficient algorithm to enumerate all permutations up to weak $M$-equivalence.

**Basic idea of the algorithm:**

- Enumerates all permutations up to **weak $M$** equivalence for given structure matrix.
- For example $4 \times 4$ state there are 10147 valid structure matrices.
- Find *smallest* representatives of each equivalence class.
When does weak $M$ imply $M$ equivalence?

- Consider the matrix $M$.
- Let $G$ be the directed graph corresponding to the adjacency matrix of $M$.
- If $G$ is strongly connected then $M$ coincides with weak $M$. 
Midori block cipher

- Energy efficient cipher
- $4 \times 4$ state
- Uses generic $p$
- MixColumns (Branch number 4, not all transitions possible)

$$
M = \begin{pmatrix}
0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 \\
1 & 1 & 1 & 0 \\
\end{pmatrix}.
$$
Takes a few days on a standard PC to find all permutations up to $\mathbf{M}$-equivalence.

- $2^{21.7}$ distinct equivalence classes.
- MILP (slow for larger number of rounds)
- Using branch and bound (Matsui’s algorithm) much faster

https://github.com/kste/matsui
Case Study: Midori

Midori64

Midori128
Conclusion

- Original permutation optimal for 1 to 12 rounds
- ...except for 9 rounds: 44 active S-boxes (instead of 41).
- For any higher number of rounds it is never optimal.
Proof in the paper

► If $p$, $p^2$ and $p^3$ have the structure matrix

$$
\mathbf{A}_p = \begin{pmatrix}
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
\end{pmatrix}
$$

then there are at least 28 active S-boxes for 6 rounds.
Case Study: Skinny

Skinny

- Lightweight Tweakable Block Cipher
- Uses AES ShiftRows
- MixColumns (Branch number 2)
Case Study: Skinny

Results using our algorithm

- weak $M$ also implies $M$ for Skinny MixColumns
- In total $2^{39.66}$ equivalence classes.
- Took 23.8 CPU days to find them.
We filter further:

- Only use permutations which give good diffusion
- Still 2,726,526 left...
- \( \approx 2937 \) CPU days to run Matsui’s for all variants
Summary

- Better theoretical understanding
- Useful tool for future designs
- Possible to evaluate the *best* choice for some designs