Building a permutation: comparing design approaches

Joan Daemen¹

based on joint work with Nicolas Bordes³, Daniël Kuijsters¹ and Gilles Van Assche²

Summer School on real-world crypto and privacy, June 17-21, 2019, Šibenik

¹Radboud University ²STMicroelectronics ³Université Grenoble Alpes
The sponge construction

Proven secure if $f$ is an ideal permutation
Keyed duplex

Proven secure if $f$ is an ideal permutation
Can likely be proven secure if $f$ is an ideal permutation
Security of these permutation-based constructions

- Build a permutation \( f \) that behaves like an ideal permutation!
- This cannot be formalized
- Assurance has to come from cryptanalytic evaluation of \( f \)
  - ...inside sponge, duplex or farfalle
- Requirements depend on the construction
- Deck functions (e.g., farfalle) are at level of block ciphers
  - PRP security of AES: distinguish \( \text{AES}[K] \) from random permutation
  - PRF security of \( \text{XOOffF} \): distinguish \( \text{XOOffF}[K] \) from random oracle
- Sponge: collision-resistance, preimage resistance, some distinguishing property ...
- This is about where security reductions stop

and cryptanalysis takes over
How to build a permutation? [Claude Shannon, 1949]

Substitution-Permutation Network (SPN): round with 2 layers:

- **non-linear** substitution layer: **S-boxes** applied in parallel
- permutation layer: **transposes** bits to different S-box positions

More rounds gives more security
There are many attack vectors in cryptanalysis

In this lecture: focus on difference propagation

Relevant in
- inner collisions: (partial) inputs leading to same state
- rebound attacks in hashing
- differential cryptanalysis in keyed constructions
- ...

Consider pairs of inputs $x$ and $x^*$ with $\Delta_{\text{in}} = x \oplus x^*$ and evaluate
- $\text{DP}(\Delta_{\text{in}}, \Delta_{\text{out}})$: probability that $f(x) \oplus f(x^*) = \Delta_{\text{out}}$
- effort to find a pair that satisfies differential $(\Delta_{\text{in}} \rightarrow \Delta_{\text{out}})$
Differences follow trails $Q$ with some probability

$$\text{DP}(Q) \approx \prod_i \text{DP}(\text{Sbox}_i)$$

Different trails may lead to same difference at output:

$$\text{DP}(\Delta_{in}, \Delta_{out}) = \sum_{\Delta_{in} \rightarrow Q \rightarrow \Delta_{out}} \text{DP}(Q)$$
SPN approach 2011 AD: Spongent

[Bogdanov, Knežević, Leander, Toz, Varici, Verbauwhede, 2011]

Table 1. 13 spongent variants.

<table>
<thead>
<tr>
<th>n</th>
<th>bcr</th>
<th>R</th>
<th>number security(bit)</th>
<th>of rounds pre. 2nd pre. col.</th>
</tr>
</thead>
<tbody>
<tr>
<td>88</td>
<td>80</td>
<td>8</td>
<td>80 40 40</td>
<td>80 40 40</td>
</tr>
<tr>
<td>88</td>
<td>176</td>
<td>88</td>
<td>135 88 44</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>128</td>
<td>8</td>
<td>70 120 64</td>
<td>120 64 64</td>
</tr>
<tr>
<td>128</td>
<td>256</td>
<td>128</td>
<td>195 128 64</td>
<td>195 128 64</td>
</tr>
<tr>
<td>160</td>
<td>160</td>
<td>16</td>
<td>90 144 80</td>
<td>90 144 80</td>
</tr>
<tr>
<td>160</td>
<td>160</td>
<td>80</td>
<td>120 80 80</td>
<td>120 80 80</td>
</tr>
<tr>
<td>160</td>
<td>320</td>
<td>160</td>
<td>240 160 80</td>
<td>240 160 80</td>
</tr>
<tr>
<td>224</td>
<td>224</td>
<td>16</td>
<td>120 208 112</td>
<td>120 208 112</td>
</tr>
<tr>
<td>224</td>
<td>224</td>
<td>112</td>
<td>170 112 112</td>
<td>170 112 112</td>
</tr>
<tr>
<td>224</td>
<td>448</td>
<td>224</td>
<td>340 224 112</td>
<td>340 224 112</td>
</tr>
<tr>
<td>256</td>
<td>256</td>
<td>16</td>
<td>140 240 128</td>
<td>140 240 128</td>
</tr>
<tr>
<td>256</td>
<td>256</td>
<td>128</td>
<td>195 128 128</td>
<td>195 128 128</td>
</tr>
<tr>
<td>256</td>
<td>512</td>
<td>256</td>
<td>385 256 128</td>
<td>385 256 128</td>
</tr>
<tr>
<td>224</td>
<td>224</td>
<td>16</td>
<td>120 208 112</td>
<td>120 208 112</td>
</tr>
<tr>
<td>256</td>
<td>256</td>
<td>128</td>
<td>195 128 128</td>
<td>195 128 128</td>
</tr>
<tr>
<td>256</td>
<td>512</td>
<td>256</td>
<td>385 256 128</td>
<td>385 256 128</td>
</tr>
</tbody>
</table>

The following building blocks are generalizations of the present structure to larger b-bit widths:

1. sBoxLayer: This denotes the use of a 4-bit to 4-bit S-box $S:F_4 \rightarrow F_4$ which is applied $b/4$ times in parallel. The action of the S-box in hexadecimal notation is given by the following table:

<table>
<thead>
<tr>
<th>x</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>S[x]</td>
<td>EDB0214F7A859C36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. pLayer: This is an extension of the (inverse) present bit-permutation and moves bit $j$ of state to bit position $P_{b}(j)$, where $P_{b}(j) = \left\lfloor \frac{j \cdot b}{4} \right\rfloor \mod b$

and can be seen in Figure 2.

3. lCounter: This is one of the four $d\log_2 R$-bit LFSRs. The LFSR is clocked once every time its state has been used and its final value is all ones. If $\zeta$ is the root of unity in the corresponding binary finite field, the $n$-bit LFRSs defined by the polynomials given below are used for the spongent variants.

Defined for any width $b$ that is a multiple of 4

Disadvantages:

- requires many rounds $n_r$: for $b = 384$, $n_r = 195$
- transposition layer makes it unsuited for software
Doing better than an SPN

\[ DP(Q) = \prod_i DP_{Sbox}(a_i, b_i) \]

- Design goal: have no trails \( Q \) with high DP
- High DP if trail has few active S-boxes or S-boxes have high DP
- Wide trail strategy: ensure all trails have many active S-boxes
Mixing layer criterion: Branch number $B$

Choose the mixing layer $\lambda$ so that
- few active S-boxes in $A$ give many active S-boxes in $\lambda(A)$
- few active S-boxes in $B$ give many active S-boxes in $\lambda^{-1}(B)$

Branch number $B$: min. over $A$ of nr. of active S-boxes in $A|\lambda(A)$
- # of active S-boxes per two rounds is at least $B$
- $B \leq n + 1$ as an input can have a single active S-box
- If $B = n + 1$, we call $\lambda$ maximum-distance separable (MDS)
Rijndael (AES) [Daemen, Rijmen 1998]

- **Strong alignment**: operates on bytes instead of bits
- **MixColumns** matrix $M$ is MDS: branch number 5
ShiftRows and SubBytes commute
Rijndael (some more)

- Recursive structure with 8 super boxes
- # active super boxes ≥ 5 so # active S-boxes ≥ 25
- 8-bit S-box with $DP \leq 2^{-6}$, so for 4R trails $DP \leq 2^{-6 \times 25} = 2^{-150}$
Disadvantages of Rijndael

- Rijndael was software-oriented
  - T-tables: 1 TLU and 32-bit XOR per byte per round
  - for 8-bit CPU: similar but more XORs and smaller tables
- Performance independent of S-box specifics: we chose the best one known
- We did choose **MixColumns** matrix with 8-bit CPU in mind
- Problem: timing attacks based on cache misses
- Countermeasure: dedicated hardware [AES-NI, Intel] or bitsliced software [Käsper, Schwabe 2009]
- **Gate cost**: # binary operations per bit per round: 16 XOR and 4 AND
Strongly aligned approach 2019 AD: Saturnin

[Canteaut, Duval, Leurent, Naya-Plasencia, Perrin, Pornin, Schrottenloher]

- Block cipher with 256-bit block length submitted to NIST lightweight
- Gate cost only 3.875 XOR and 1.5 AND/OR
  - 4-bit S-box layer: 1.5 XOR and 1.5 AND/OR
  - MC matrix MDS $B = 5$ with cost 2.375 XOR
- AES square becomes $4 \times 4 \times 4$ cube
Recursive structure with 64-bit mega boxes

- Mega box has 16-bit super boxes, that have 4-bit S-boxes
- # active S-boxes is $5^3$ and the S-boxes have $\text{DP} \leq 2^{-2}$
- 8-round trails have $\text{DP} \leq 2^{-250}$
Disadvantages of Saturnin: ShiftRows

- There are three transposition mappings:
  - Identity in even-indexed rounds
  - SR\textsubscript{slice} if index is 1 modulo 4
  - SR\textsubscript{sheet} if index is 3 modulo 4

- Hardware: gives hassle in single-round combinatorial logic
- Not so efficient in software, e.g., on ARM Cortex M3
  - SR\textsubscript{sheet} costs more than MC step
  - SR\textsubscript{slice} costs more than MC + S-box layer
Weakly aligned approach 2017 AD: Xoodoo

- 384-bit permutation
- Size and *shape* inspired by Gimli [Bernstein, Kölbl, Lucks, Massolino, Mendel, Nawaz, Schneider, Schwabe, Standaert, Todo, Viguier, 2017]
- Design approach: that of the permutation in KECCAK
- **Weak alignment**: operates on bits rather than bytes (or nibbles)
Iterated: $n_r$ rounds that differ only by round constant
Nonlinear mapping $\chi$

Effect on one plane:

$\chi$ as in KECCAK-$p$, operating on 3-bit columns

Involution and same propagation differentially and linearly
Mixing layer $\theta$

- Column parity mixer: compute parity, fold and add to state
- Good average diffusion, identity for states in kernel
- Heavy inverse
Plane shift $\rho_{\text{east}}$

- After $\chi$ and before $\theta$
- Shifts planes $y = 1$ and $y = 2$ over different directions
After $\theta$ and before $\chi$.

Shifts planes $y = 1$ and $y = 2$ over different directions.
Xoodoo implementation aspects

- One single round function
- Gate cost: 3 XOR and 1 AND
  - 5 Xoodoo rounds cost less than 4 Saturnin rounds
- Transposition layer software-friendly, e.g., on ARM Cortex M3
  - $\rho$ steps have zero overhead
  - 7 Xoodoo rounds cost less than 4 Saturnin rounds
Disadvantage of XOODOO

▶ No simple proofs for trail DP values
▶ Because it does not have super-boxes
▶ Instead computer-assisted bounds
  • Scanning the space of trails with DP below some limit
  • Using techniques of [Mella, Daemen, Van Assche, ToSC 2016]

Comparison of minimum trail weights \( w(Q) = - \log_2 DP(Q) \)

<table>
<thead>
<tr>
<th># rounds:</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spongent</td>
<td>2</td>
<td>4</td>
<td>8</td>
<td>12</td>
<td>28</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Rijndael</td>
<td>6</td>
<td>30</td>
<td>54</td>
<td>150</td>
<td>( \geq 180 )</td>
<td>( \geq 300 )</td>
<td>( \geq 330 )</td>
</tr>
<tr>
<td>Saturnin</td>
<td>2</td>
<td>10</td>
<td>18</td>
<td>50</td>
<td>210</td>
<td>250</td>
<td>260</td>
</tr>
<tr>
<td>XOODOO</td>
<td>2</td>
<td>8</td>
<td>36</td>
<td>( \geq 74 )</td>
<td>( \geq 104 )</td>
<td>( \geq 148 )</td>
<td>( \geq 180 )</td>
</tr>
</tbody>
</table>
A closer look at difference propagation
What we compare

- Permutations we study have a round function with two layers
  - non-linear S-box layer
  - linear layer, that we will denote as $\lambda$
- We don’t consider other design approaches
  - ARX such as in ChaCha, ARX-like such as in NORX
  - no S-box layer, as Gimli, Subterranean or Friet
  - Feistel networks, ...
- These are harder to describe and compare, so future work
- We will limit our comparison to the following

<table>
<thead>
<tr>
<th></th>
<th>alignment</th>
<th>diffusion</th>
<th>S-box size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spongent</td>
<td>weak</td>
<td>weak</td>
<td>4</td>
</tr>
<tr>
<td>Rijndael</td>
<td>strong</td>
<td>strong</td>
<td>8</td>
</tr>
<tr>
<td>Saturnin</td>
<td>strong</td>
<td>strong</td>
<td>4</td>
</tr>
<tr>
<td>Xoodoo</td>
<td>weak</td>
<td>strong</td>
<td>3</td>
</tr>
</tbody>
</table>
Characterizing diffusion of the linear layer

- Currently mostly considered: branch number $B$
  
  \[ B = \min_{a \neq 0} (w(a) + w(\lambda(a))) \]

  with $w(x)$ the box weight: # active S-boxes in $x$

- More informative: histogram of # states per weight of $(a, \lambda(a))$
  
  - for all $2^b$ states $a$ compute $w(a) + w(\lambda(a))$
  - list them in a histogram
  - the tail of the histogram at the low-weight end says something about diffusion

- Hamming weight histogram: absolute diffusion power independent of S-box layer

- Box weight histogram: the 2-round differential trails with given # active S-boxes
Hamming weight histograms
From Hamming weight to box weight

\[
a \rightarrow \lambda(a)
\]
From box weight to trail weight

\[ \Delta_{\text{in}} \]
\[ a \]
\[ \lambda(a) \]
\[ \Delta_{\text{out}} \]
**\(\lambda\)-box partition**

- **\(\Delta_{in}\)**

- **\(\Delta_{out}\)**

- **\(\lambda(a)\)**

- **\(a\)**

- **Trail:** sequence of states (differences) \((\Delta_{in}, a, \lambda(a), \Delta_{out})\)

- **box pattern:** set of states with same pattern of passive boxes

- In the trail above:
  - \(\Delta_{in}\) and \(a\) are in same box pattern
  - \(\Delta_{out}\) and \(\lambda(a)\) are in same box pattern

- So differential \((\Delta_{in}, \Delta_{out})\) fixes box pattern of \(a\) and \(\lambda(a)\)

- **\(\lambda\)-box partition** of states \(a\) by box pattern of \((a, \lambda(a))\)

- Size of these subsets bound the number of trails in a differential
### λ-box partition: Rijndael and Saturnin

#### Rijndael super box:

<table>
<thead>
<tr>
<th>box weight</th>
<th># subsets × size</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>(56 × 255)</td>
</tr>
<tr>
<td>6</td>
<td>(28 × 64005)</td>
</tr>
<tr>
<td>7</td>
<td>(8 × 16323825)</td>
</tr>
<tr>
<td>8</td>
<td>(1 × 4162570275)</td>
</tr>
</tbody>
</table>

Quasi all differentials have multiple trails [Daemen Rijmen, 2008]

#### Saturnin super box:

<table>
<thead>
<tr>
<th>box weight</th>
<th># subsets × size</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>(56 × 15)</td>
</tr>
<tr>
<td>6</td>
<td>(28 × 165)</td>
</tr>
<tr>
<td>7</td>
<td>(8 × 2625)</td>
</tr>
<tr>
<td>8</td>
<td>(1 × 39075)</td>
</tr>
</tbody>
</table>

There is significant clustering of trails in differentials
Differential weight histogram

Saturnin Trails
Saturnin differentials
Xoodoo
### \( \lambda \)-box partition of Xoodoo (modulo horizontal translation)

<table>
<thead>
<tr>
<th>box weight</th>
<th>( # \text{ subsets} \times \text{size} )</th>
<th>( # \text{ subsets} \times \text{size} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>(3 \times 1)</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>(24 \times 1)</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>(600 \times 1)</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>(2 \times 1)</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>(442 \times 1)</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>(10062 \times 1)</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>(80218 \times 1)</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>(11676 \times 1)</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>(228531 \times 1)</td>
<td>(3 \times 2)</td>
</tr>
<tr>
<td>15</td>
<td>(2107864 \times 1)</td>
<td>(90 \times 2)</td>
</tr>
<tr>
<td>16</td>
<td>(8447176 \times 1)</td>
<td>(702 \times 2)</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

Differentials with more than a single trail are rare

For 3 rounds: all trails \( w(Q) < 50 \) are alone in their differential
Truncated differentials

\[ \Delta_{in} \]  
\[ \bar{a} \]  
\[ \lambda \]  
\[ \bar{b} \]  
\[ \Delta_{out} \]

- **Truncated trail**: sequence of box patterns \((\Delta_{in}, \bar{a}, \bar{b}, \Delta_{out})\)

\[ \text{DP}(\Delta_{in}, \Delta_{out}) = \Pr(y \in \Delta_{out}\mid x \in \Delta_{in}) \]

- S-box layers have probability 1 iff \(\Delta_{in} = \bar{a}\) and \(\bar{b} = \Delta_{out}\), so

\[ \text{DP}(\Delta_{in}, \Delta_{out}) = \Pr(\lambda(a) \in \Delta_{out}\mid a \in \Delta_{in}) \]
Truncated differentials in Rijndael and Saturnin

In Rijndael super box: \( \text{DP}(\Delta_{\text{in}}, \Delta_{\text{out}}) \approx 255^{w(\Delta_{\text{out}}) - 4} \)

- AES has 4-round super box truncated diff. with \( \text{DP} = 1 \)
- Saturnin has 8-round mega box truncated diff. with \( \text{DP} = 1 \)

These can be used as distinguisher in attacks

Can also be exploited in other attacks, e.g.,

- impossible differentials (most powerful AES attack)
- rebound attacks in hashing
In Xoodoo truncated trails have higher weight than trails.
Conclusions

Please make up your own conclusions

Thanks for listening!

Part of this work was sponsored by ERC grant ESCADA