Distinguisher for full final round of Fugue-256

Jean-Philippe Aumasson and Raphael C.-W. Phan

Fugue-256

256-bit version of SHA-3 candidate Fugue

30$\times$32=960-bit internal state

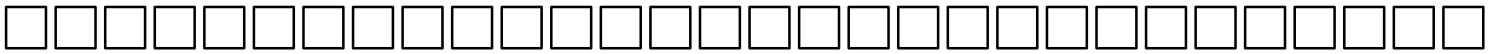
“Round transform” **R** processes 32-bit message chunks

“Final round” **G** takes the final state and returns a digest via a permute+truncate transform

Previous work (Khovratovich): internal collisions in 2^{352} time and space

Fugue-256: round transform **R**

30×32 = 960-bit internal state



32-bit message blocks integrated through **R** transform

R makes 2 AES-like rounds on 4-word windows

Trivial distinguishers (e.g., a block affects 11 state words)

⇒ **G** crucial to obtain random-looking digests

Fugue-256: final round **G**

30×32 = 960-bit internal state $S_0, \ldots, S_{29}$

Message-independent, permutate+truncate

18 double-AES-like rounds:

5 $G1$ rounds	**ROR3**; **CMIX**; **SMIX** **ROR3**; **CMIX**; **SMIX**
13 $G2$ rounds	$S_4 += S_0$; $S_{15} += S_0$; **ROR15**; **SMIX** $S_4 += S_0$; $S_{16} += S_0$; **ROR14**; **SMIX**

Returns

$$S_1, S_2, S_3, (S_4 + S_0), (S_{15} + S_0), S_{16}, S_{17}, S_{18}$$

SMIX

Transforms (S_0, S_1, S_2, S_3) with AES' Sbox followed by a linear transform using

$$\mathbf{N} = \begin{pmatrix}
1 & 4 & 7 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 1 & 4 & 7 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 7 & 1 & 1 & 4 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 4 & 7 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 4 & 7 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 4 & 7 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 7 & 1 & 0 & 4 \\
4 & 7 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 7 & 0 & 0 & 0 & 6 & 4 & 7 & 1 & 7 & 0 & 0 & 0 \\
0 & 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7 & 0 & 0 & 1 & 6 & 4 & 7 \\
7 & 1 & 6 & 4 & 0 & 0 & 7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7 & 0 \\
0 & 0 & 0 & 7 & 4 & 7 & 1 & 6 & 0 & 0 & 0 & 7 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 5 & 4 & 7 & 1 \\
1 & 5 & 4 & 7 & 0 & 0 & 0 & 0 & 0 & 4 & 0 & 0 & 0 & 4 & 0 & 0 \\
0 & 0 & 4 & 0 & 7 & 1 & 5 & 4 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \\
0 & 0 & 0 & 4 & 0 & 0 & 0 & 4 & 4 & 7 & 1 & 5 & 0 & 0 & 0 & 0
\end{pmatrix}$$

SMIX

Transforms (S_0, S_1, S_2, S_3) with AES' Sbox followed by a linear transform using

$$\mathbf{N} = \begin{pmatrix}
1 & 4 & 7 & 1 & 1 & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\
\cdot & 1 & \cdot & \cdot & 1 & 1 & 4 & 7 & \cdot & 1 & \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\
\cdot & \cdot & 1 & \cdot & \cdot & \cdot & 1 & \cdot & 7 & 1 & 1 & 4 & \cdot & \cdot & 1 & \cdot \\
\cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & 1 & 4 & 7 & 1 & 1 \\
\cdot & \cdot & \cdot & \cdot & \cdot & 4 & 7 & 1 & 1 & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\
\cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & 4 & 7 & \cdot & 1 & \cdot & \cdot \\
\cdot & \cdot & 1 & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & 7 & 1 & \cdot & 4 \\
4 & 7 & 1 & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & 7 & \cdot & \cdot & \cdot & 6 & 4 & 7 & 1 & 7 & \cdot & \cdot & \cdot \\
\cdot & 7 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 7 & \cdot & \cdot & 1 & 6 & 4 & 7 \\
7 & 1 & 6 & 4 & \cdot & \cdot & 7 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 7 & \cdot \\
\cdot & \cdot & \cdot & 7 & 4 & 7 & 1 & 6 & \cdot & \cdot & \cdot & 7 & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & 4 & \cdot & \cdot & \cdot & 4 & \cdot & \cdot & \cdot & 5 & 4 & 7 & 1 \\
1 & 5 & 4 & 7 & \cdot & \cdot & \cdot & \cdot & \cdot & 4 & \cdot & \cdot & \cdot & 4 & \cdot & \cdot \\
\cdot & \cdot & 4 & \cdot & 7 & 1 & 5 & 4 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 4 & \cdot \\
\cdot & \cdot & \cdot & 4 & \cdot & \cdot & \cdot & 4 & 4 & 7 & 1 & 5 & \cdot & \cdot & \cdot & \cdot
\end{pmatrix}$$

R: 2×AES

AES	AES

G: 36×AES

AES	AES
AES	AES
AES	AES
AES	AES
AES	AES
AES	AES
AES	AES
AES	AES
AES	AES
AES	AES
AES	AES
AES	AES
AES	AES
AES	AES
AES	AES
AES	AES
AES	AES
AES	AES

Our main results

Black-box distinguisher for **G** minus a linear layer

- Integral cryptanalysis
- Track propagation of multiset properties
- Exploit sparsity of the linear diffusion layer
- Need only 256 related but unknown inputs

White-box distinguisher for full **G**

- Start-in-the-middle strategy
- Exploit probability-1 differential characteristics
- Needs only two computations of **G**

Black-box distinguisher

256-element multiset of bytes characterized as

- P: permutation of GF(256)
- C: constant value
- B: values summing to zero

"Sbox(X) = X", for X in { P, C }, "Sbox(B) = ?"

$+$	P	C	B	?
P	B	P	B	?
C	P	C	B	?
B	B	B	B	?
?	?	?	?	?

Black-box distinguisher

SMIX($S_0 S_1 S_2 S_3$) = Super-Mix(Sbox($S_0 S_1 S_2 S_3$))

if $S_0 S_1 S_2 S_3$ is CCCC CCCC PCCC CCCC then

Sbox($S_0 S_1 S_2 S_3$) $=$ CCCC CCCC PCCC CCCC

and Super-Mix(Sbox($\cdots$)) is

PCPC PPCC PCCC PCCP

Track properties through 5.5 rounds...

???? P??? B??? B???

Black-box distinguisher

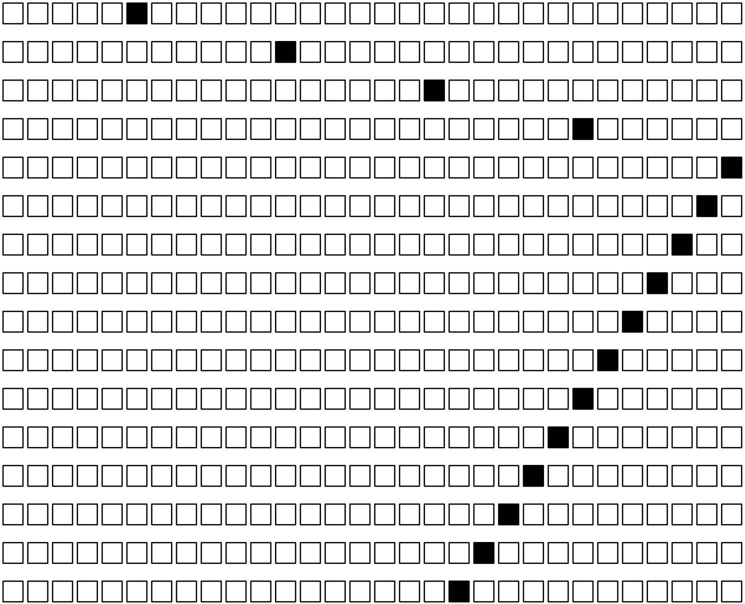
Omit "$S_{16} += S_0$" and Super-Mix at round 6, return S_{14}

$\Rightarrow$ After 18 rounds, S_{14} is ??P? (theory)

Distinguisher:

- Collect 256 outputs from distinct unknown inputs varying only S_5's first byte
- Check that S_{14}'s third byte is always unique

Probability-1 characteristic for 15 rounds of **G**



White-box distinguisher

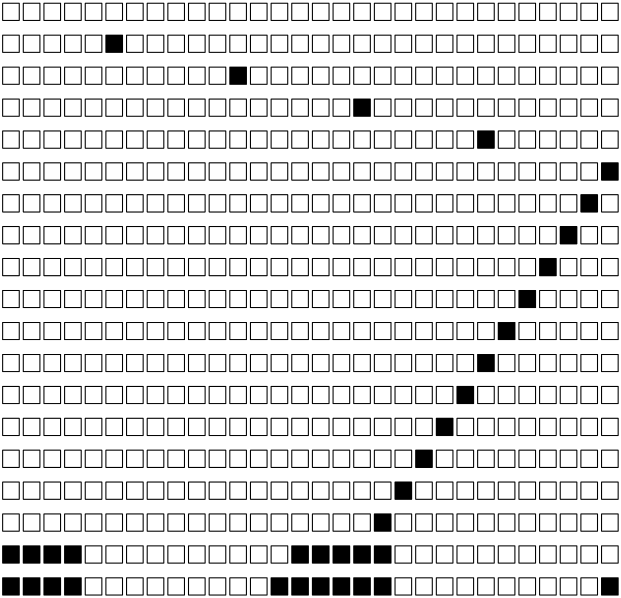
Choose two internal states before round 17 such that

- The proba-1 characteristics is followed backwards til round 2
- The two digests have fixed values

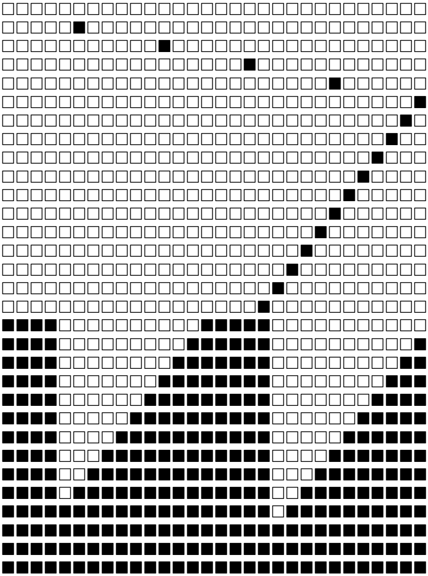
$\Rightarrow$ Find many pairs (S, S') such that

- $\mathbf{G}(S)$ and $\mathbf{G}(S')$ are fixed
- $S \oplus S'$ has Hamming weight ≈ 66

Probability-1 distinguisher on full 18-round **G**



And up to 30 rounds of untruncated **G**



Conclusions

Efficient distinguisher for **G** (and more), though not Fugue-256

Existence of high-probability characteristics previously conjectured by the designers; doesn't seem to assist attacks on the hash

Difficult to support RO-indifferentiability claims...
$\Rightarrow$ are Shabal-like relaxed proofs applicable ?