Bash-f: another LRX sponge function

S. Agievich, V. Marchuk, A. Maslau, V. Semenov

Research Institite for Applied Problems of Math. and Informatics

Belarusian State University

CTCrypt-2016, Yaroslavl, Russia, 07 June 2016

1. Bash

$\texttt{Bash} = \overset{\texttt{B}}{\not{\texttt{h}}}\texttt{ash}$

$\texttt{Bash}_{2l} : \{0,1\}^* \to \{0,1\}^{2l}$, $l \in \{16, 32, \ldots, \mathbf{128}, \ldots, \mathbf{192}, \ldots, \mathbf{256}\}$

The platform: sponge + LRX

Influenced by: Keccak

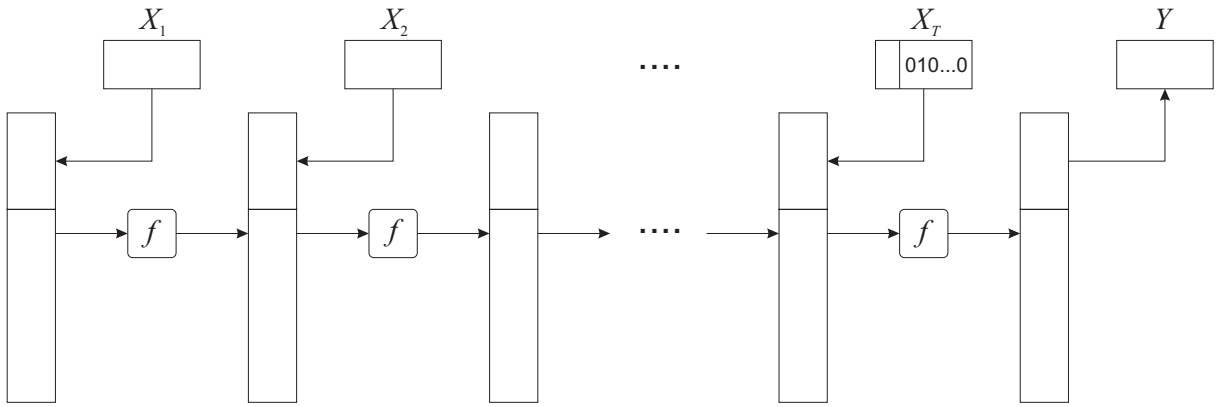
The core: `Bash-f` (another sponge LRX function)

2. Objectives

- **wide**: to support wide range of hash lengths
- **fast**: to be competitive in performance
- **strong**: to be competitive in security guarantees
- **lightweight**: to be suitable for low-resource hardware platforms
- **rigidity**: use reasonable quality criteria as detailed as possible ["With all the richness of choice there is no alternative"]

3. The sponge construction

The hashing process (the overwrite mode, $|X_t| = 1536 - 4l$):

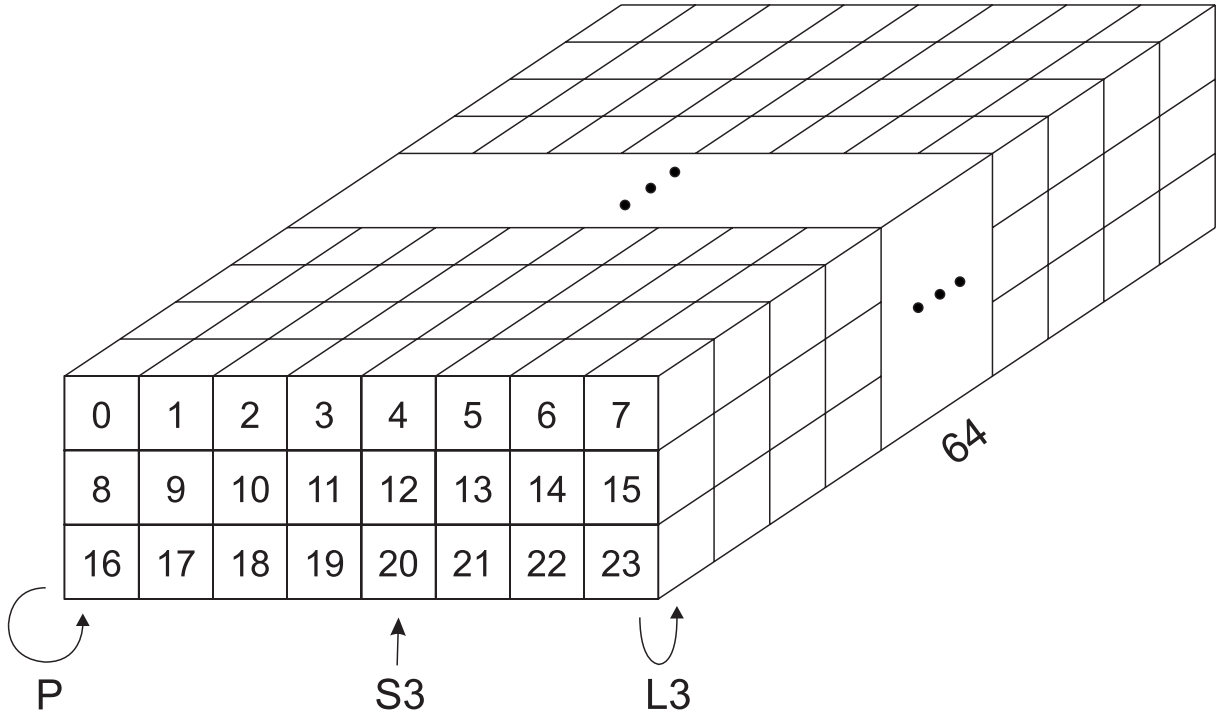


Applications (following to the Sponge team):

- tree hashing
- PRNG
- message authentication codes
- stream encryption
- secure sessions

4. The hash state

$S = S_0 \parallel S_1 \parallel \ldots \parallel S_{23}$
(1536 bits = 24 words × 64 bits)



Planes: vertical $(0, 8, 16)$, horizontal $(0, 1, \ldots, 7)$

5. LRX

Logical operations:

$\neg, \vee, \wedge$

Rotations (circular):

$\texttt{RotHi}^d$

Xors:

$\oplus$

Without $\boxplus$!

6. Bash-f

Mappings:

$L3$ — linear diffusion mappings (`RotHi`, $\oplus$)

$S3$ — a nonlinear mapping ($\neg, \wedge, \vee, \oplus$)

P — permutation of words

$L3$ and $S3$ sequentially transform separate vertical planes (potentially high parallelism!)

7. $S3$

$(W_0, W_1, W_2) \leftarrow (W_0 \oplus W_1 \vee \neg W_2, W_1 \oplus W_0 \vee W_2, W_2 \oplus W_0 \wedge W_1)$

The bitslice technique: simultaneosly apply an S-box $s\colon \{0,1\}^3 \to \{0,1\}^3$ to all vertical bit triples of W_i

The basic criteria:

- **R1** — $R_{\oplus\oplus}(s) = 4$
- **R2** — $Nl(s) = 4$
- **R3** — $\deg(s) = 2$
- **R4** — $(\star)$ holds for each nonzero in/out difference $(\alpha_0\alpha_1\alpha_2, \beta_0\beta_1\beta_2)$
- **R5** — $(\star)$ holds for each nonzero out/in mask $(\alpha_0\alpha_1\alpha_2, \beta_0\beta_1\beta_2)$

$$\alpha_0\beta_0 + \alpha_1\beta_1 + \alpha_2\beta_2 = 1 \qquad (\star)$$

R4 + **R5** $\Rightarrow$ analysis of the strength against differential and linear attacks are the same

8. $S3$-II

The extended criteria:

R6 — no fixed points

R7 — can be implemented with 7 operations

R8 — a complex description by polynomials over $\mathbb{F}_8$

$(W_0, W_1, W_2) \leftarrow S3(W_0, W_1, W_2)$:

$T_0 \leftarrow \neg W_2$,

$T_1 \leftarrow W_0 \vee W_2$,

$T_2 \leftarrow W_0 \wedge W_1$,

$T_0 \leftarrow T_0 \vee W_1$,

$W_1 \leftarrow W_1 \oplus T_1$,

$W_2 \leftarrow W_2 \oplus T_2$,

$W_0 \leftarrow W_0 \oplus T_0$.

Complexity: $1\neg + 2\vee + 1\wedge + 3\oplus$

9. $L3$

Parameters: $[m_1, n_1, m_2, n_2]$ — shift vaules

$(W_0, W_1, W_2) \leftarrow L3[m_1, n_1, m_2, n_2](w_0, w_1, w_2)$:

$$
\begin{aligned}
W_0 &\leftarrow w_0 \oplus w_1 \oplus w_2, \\
W_1 &\leftarrow w_1 \oplus \texttt{RotHi}^{m_1}(w_0) \oplus \texttt{RotHi}^{n_1}(W_0), \\
W_2 &\leftarrow w_2 \oplus \texttt{RotHi}^{m_2}(w_2) \oplus \texttt{RotHi}^{n_2}\left(w_1 \oplus \texttt{RotHi}^{n_1}(W_0)\right).
\end{aligned}
$$

Complexity: 4 $\texttt{RotHi}$ +6$\oplus$

10. $L3$-II

Conventions: $\oplus = +$, $\texttt{RotHi}^d(w) = w^d$

The structure:

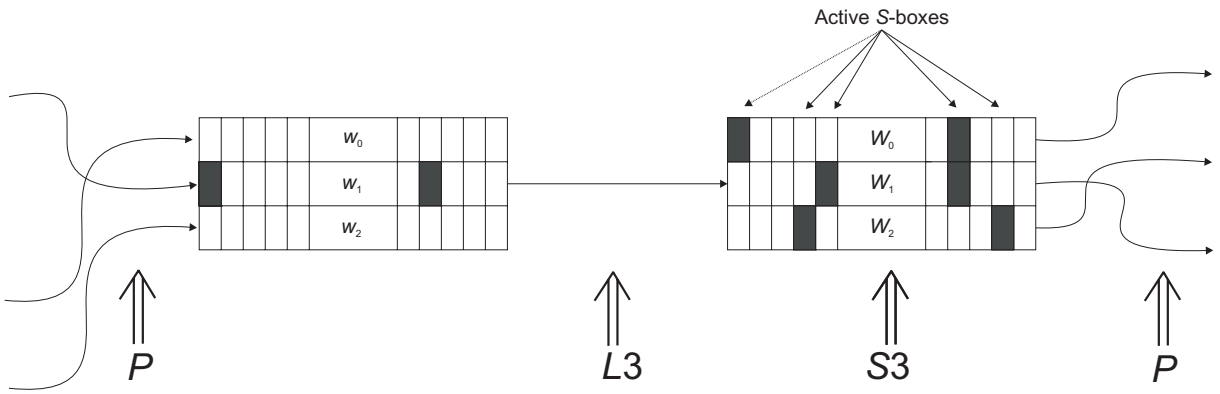
$$
\begin{aligned}
W_0 &\leftarrow w_0 + w_1 + w_2, \\
W_1 &\leftarrow w_1 + a^{m_1} + b^{n_1}, \\
W_2 &\leftarrow w_2 + c^{m_2} + d^{n_2},
\end{aligned}
$$

a, b, c, d are constructed from w_i

The basic criteria:

- **R1** — only 6 xors (a, b, c, d can be calculated without extra cost)
- **R2** — any bit of any w_i affects each W_j
- **R3** — any bit of any W_j affects each w_i (under $L3^{-1}$)
- **R4** — BranchNumber($L3$) $\geqslant 5$

11. $L3$: activation of S-boxes



- min[$i \rightarrow$?] — the minimal number of S-boxes activated by i errors
- min[? $\rightarrow j$] — the minimal number of errors to activate j S-boxes
- #[$i \rightarrow j$] — the number of i errors to activate j S-boxes

R5: min[1 $\rightarrow$?] = 4, min[2 $\rightarrow$?] = 3, min[3 $\rightarrow$?] = 3

R6: min[? $\rightarrow j$] as large as possible (for small j)

R7: #[$i \rightarrow j$] as small as possible (for small i and j)

12. $L3^{-1}$

To inverse the chosen $L3$ we have to solve the following equation

$$w_2 + w_2^{m_1} + w_2^{m_2} + w_2^{m_1+m_2} + w_2^{m_1+n_2} = F(W_0, W_1, W_2)$$

in w_2

The characteristic polynomial:

$$f(x) = 1 + x^{m_1} + x^{m_2} + x^{m_1+m_2} + x^{m_1+n_2}$$

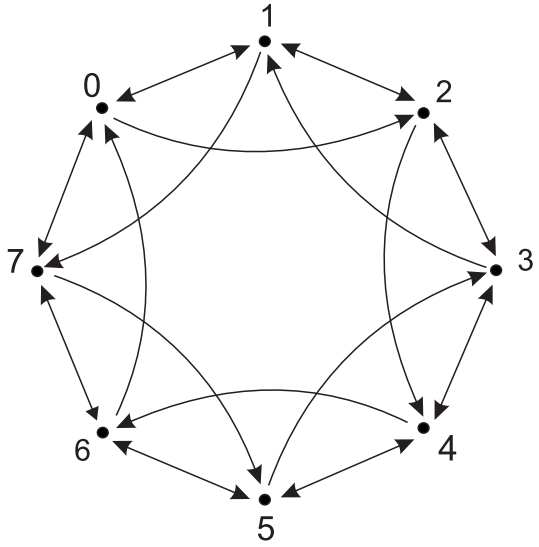
(pentanomial!)

13. P

P rotates horizontal planes (upward or downward) and simultaneously shuffles words in the planes using permutations π_i

The digraph $G = G(P)$:
the vertices $0, 1, \ldots, 7$
the arcs $\{(u, v) : v \in \{\pi_0(u), \pi_1(u), \pi_2(u)\}\}$

R1, **R2**: G is strongly regular

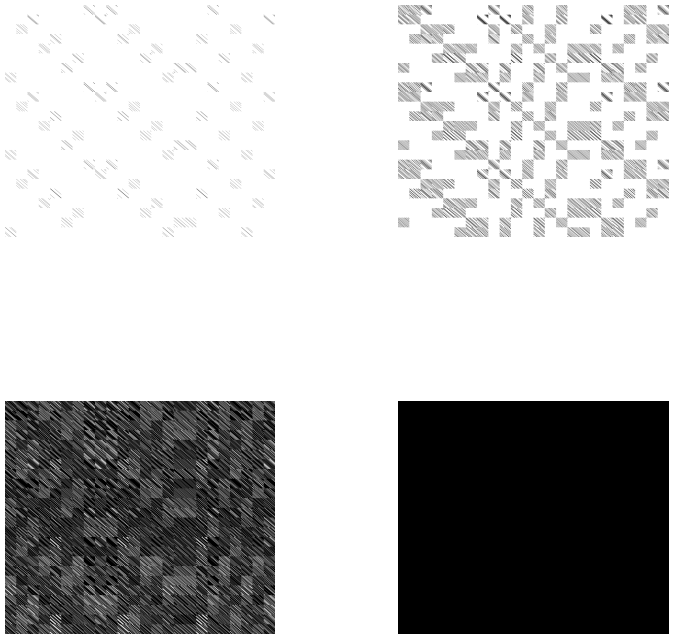


14. P-II

R3, **R4**: using symmetries to propagate errors as fast as possible

Shift tuples for ith vertical plane ($\operatorname{ord} g = 8$): $g^i[m_1, n_1, m_2, n_2]$ mod 64

The dependency matrices (after the first 4 rounds):



15. The algorithm

Algorithm `Bash-f`

Input: $S = S_0 \parallel S_1 \parallel \ldots \parallel S_{23}$, $S_i \in \{0,1\}^{64}$.

Output: the updated state S.

Steps:

1. $C \leftarrow \texttt{B194BAC80A08F53B}_{16}$.
2. For $i = 1, 2, \ldots, 24$:
   (a) $[m_1, n_1, m_2, n_2] \leftarrow [8, 53, 14, 1]$;
   (b) for $v = 0, 1, \ldots, 7$:
      i. $(S_v, S_{v+8}, S_{v+16}) \leftarrow L3[m_1, n_1, m_2, n_2](S_v, S_{v+8}, S_{v+16})$;
      ii. $(S_v, S_{v+8}, S_{v+16}) \leftarrow S3(S_v, S_{v+8}, S_{v+16})$;
      iii. $[m_1, n_1, m_2, n_2] \leftarrow 7[m_1, n_1, m_2, n_2] \bmod 64$;
   (c) $S \leftarrow S_{P(0)} \parallel S_{P(1)} \parallel \ldots \parallel S_{P(23)}$;
   (d) $S_{23} \leftarrow S_{23} \oplus C$;
   (e) if the lowest bit of C is zero, then $C \leftarrow \texttt{ShLo}(C)$;
      else $C \leftarrow \texttt{ShLo}(C) \oplus \texttt{AED8E07F99E12BDC}_{16}$.
3. Return S.

16. Performance

Operations for one round:

operation	Bash	Keccak
$\neg$	8^*	$5 \cdot 5 = 25^*$
$\oplus$	$8(6+3)+1 = 73$	$5(5+10)+1 = 76$
$\wedge$	8	$5 \cdot 5 = 25$
$\vee$	16	0
`RotHi`	$8 \cdot 4 = 32$	$5(5+1) = 30$
overall	137	156
block length	$1536 - 4l$	$1600 - 4l$

* — can be avoided if the instruction AND-NOT is available

The reference implementation: `https://github.com/agievich/bee2`

17. Differential analysis

A differential trail: a sequence of xor-differences between hash states

An active S-box: s gets nonzero input difference
(fork: 4 output differences are possible)

The minimal number of active S-boxes in d rounds: $s_A(d)$

The objective: prove that $s_A(d)$ is large

Current estimates:

I) $s_A(4) \geqslant 26$ (by hands) $\Rightarrow s_A(24) \geqslant 162$

II) $s_A(24) \geqslant 150$ (automated)

`Keccak` **(in compatible units)**: 148 active S-boxes