Observations on Prøst and Minalpher

Kazumaro Aoki NTT

January 15, 2015 @ ESC 2015

Motivation

- **CAESAR submissions**
- **Permutation-based**
  (Even-Mansour and not including Sponge construction)
- **4-bit sbox**
- **256-bit I/O**

$\Rightarrow$ **Prøst-128 and Minalpher-**P

Contents

- **Fixed linear spaces in sbox**
  (joint work with Mitsuru Matsui)
- **Implementation on Sandy Bridge/Ivy Bridge and Haswell/Broadwell**
  (joint work with Yosuke Todo)

Fixed Linear Spaces in sbox

Definition 1 V ***is a linear space and*** $S(V) = V$***, and then we call*** V ***is a fixed linear space with respect to*** S***.***

When designing Minalpher-P **sbox, we noticed that the property may harm the primitive.**

Related work: invariant subspace attack [LMR@EC15, LAAZ@C11]

Fixed Linear Space Examples

Ex. 1: When 0 is not a fixed point of S, $\{0, S(0)\}$ is a 1-dimensional fixed linear space.

Ex. 2: When 0 and x ($\neq 0$) is a fixed point, then $\{0, x\}$ is a 1-dimensional fixed linear space.

Case for Prøst-128 (1/3)

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(x)$	0	4	8	F	1	5	E	9	2	7	A	C	B	D	6	3

1-dim: $\{0,5\}$, $\{0,A\}$, $\{0,D\}$

2-dim: $\{0,1,4,5\}$, $\{0,2,8,A\}$

Let $V = \{0,1,4,5\}$, **cosets are** $\bar{0}$, $\bar{2}$, $\bar{8}$, **and** $\bar{A}$.

$S(\bar{x}) = \bar{y}$ **with probability**

$x \backslash y$	0	2	8	A
0	**1**	**0**	**0**	**0**
2	**0**	**0**	**1/2**	**1/2**
8	**0**	**1/2**	**1/4**	**1/4**
A	**0**	**1/2**	**1/4**	**1/4**

.

Case for Prøst-128 (2/3)

- $\bar{x} \oplus \bar{y} = \overline{x \oplus y}$ holds. That is, we can follow the characteristic with probablity 1 for XOR and AddConst.
- For MixSlice, $M = \begin{bmatrix} EABC \\ AECB \\ BCEA \\ CBAE \end{bmatrix}$ is applied, where $E, A, B, C \in \mathrm{GF}(2)^{4\times 4}$ and E is an identity map and $C = A \oplus B$.

Case for Prøst-128 (3/3)

$\overline{Bx} = \bar{y}$ with probability

$x \backslash y$	0	2	8	A
0	1/2	0	1/2	0
2	1/2	0	1/2	0
8	0	1/2	0	1/2
A	0	1/2	0	1/2

.

$\overline{Cx} = \bar{y}$ with probability

$x \backslash y$	0	2	8	A
0	1/2	1/2	0	0
2	0	0	1/2	1/2
8	1/2	1/2	0	0
A	0	0	1/2	1/2

.

Case for Prøst-128 (3/3)

$\overline{Bx} = \bar{y}$ with probability

$x \backslash y$	0	2	8	A
0	**1/2**	**0**	**1/2**	**0**
2	**1/2**	**0**	**1/2**	**0**
8	**0**	**1/2**	**0**	**1/2**
A	**0**	**1/2**	**0**	**1/2**

.

$\overline{Cx} = \bar{y}$ with probability

$x \backslash y$	0	2	8	A
0	**1/2**	**1/2**	**0**	**0**
2	**0**	**0**	**1/2**	**1/2**
8	**1/2**	**1/2**	**0**	**0**
A	**0**	**0**	**1/2**	**1/2**

.

$\overline{Ax} = \bar{y}$ with probability

$x \backslash y$	0	2	8	A
0	**1/4**	**1/4**	**1/4**	**1/4**
2	**1/4**	**1/4**	**1/4**	**1/4**
8	**1/4**	**1/4**	**1/4**	**1/4**
A	**1/4**	**1/4**	**1/4**	**1/4**

.

Comments

- There is no 3-dimensional fixed linear space for Prøst sbox, and there is no 2- and 3-dimensional fixed linear space for Minalpher-P sbox.
- Using $V = \{0, 5\}$ as a 1-dimensional fixed linear space for Prøst sbox, only 2 rounds keep non-uniform distribution with the assumption of independent probability distribution for XOR and S.
- Considering MixSlices$\circ$SubRows instead of independent MixSlices and SubRows, probability does not diverge.

Implementation on Recent Intel CPU

Timeline

2011.01 Release of Sandy Bridge
2012.04 Release of Ivy Bridge
2013.01 CAESAR Call
2013.06 Release of Haswell
2014.03 CAESAR Deadline
2015.01 Release of Broadwell

Characteristic Difference

- **Sandy Bridge and Ivy Bridge**
  — 128-bit SIMD integer
- **Haswell and Broadwell**
  — 256-bit SIMD integer

We hope that a SIMD implementation on $*$ Bridge is twice faster than $*$well.

`vperm` Implementation

- `vperm` **(vector permute) instruction efficiently look up 16 or 32 4-bit sboxes in parallel.**
- **Intel** `pshufb` **realizes** `vperm`**.**

`vpshufb xmm2, xmm0, xmm1`

$S(x)$	**0 4 8 F 1 5 E 9 2 7 A C B D 6 3**	**xmm0**
input	**0 0 1 3 C F D 2 3 D 6 E A A 3 6**	**xmm1**
output	**0 0 4 F B 3 D 8 F D E 6 A A F E**	**xmm2**

Hamburg, "Accelerating AES with Vector Permute Instructions," CHES 2009

Round Function

States are represented by 4×16 nibbles.

Prøst-128: SubRows + MixSlices + ShiftPlanes + AddConstants

Minalpher-P: SubNibbles + ShuffleRows + SwapMatrices + XorMatrix + MixColumns + $\oplus$ of Round Constant

`vperm` **implementation of 1 round:**

Sub, Shuft/Shuffle: `pshufb`
Mix, Const: `pxor`

1-round Implementatio on $*$ Bridge

Prøst-128: pshufb$\times 4 \times 4$ + pxor$\times 3 \times 4$ +
pshufb$\times 4$ + pxor$\times 4$
= **shuffle**$\times 20$ + **logical**$\times 16$

Minalpher-P: pshufb$\times 4$ + pshufb$\times 4$ +
psrldq$\times 4$ + pxor$\times 4$ + pxor$\times 6$ +
pxor$\times 4$
= **shuffle**$\times 8$ + **shift**$\times 4$ + **logical**$\times 14$

1-round Implementation on ∗well

Prøst-128:

vperm2i128 × 2 + pshufb × 4 × 2 +
pxor × 3 × 2 + pshufb × 2 + pxor × 2
= **shuffle** × 12 + **logical** × 8

Minalpher-P: pshufb × 2 + pshufb × 2 +
psrldq × 2 + pxor × 2 + vperm2i128 +
pxor × 3 + pxor × 2
= **shuffle** × 5 + **shift** × 2 + **logical** × 7

Summary of 1-round Implementation

	shuffle	shift	logical
Prøst-128			
∗ Bridge	20	0	16
∗well	12	0	8
Minalpher-P			
∗ Bridge	8	4	14
∗well	5	2	7

∗ Bridge vs ∗well

SIMD operation

	∗ Bridge	∗well
port 0	logical	logical/shift
port 1	logical/shuffle/shift	logical
port 5	logical/shuffle/shift	logical/shuffle

Therefore, shift or shuffle instruction on ∗well has the same performance of ∗ Bridge.

Parallel Implementation on $*$well

Prøst-128 16-way:

$16 \times$(logical $\times (\leq 8.5)$ + shift $\times 2$)

(with bitsliced SubRows)

Minalpher-P 4-way:

$4 \times$(logical $\times 5$ + shuffle $\times 4$)

	Prøst-128	Minalpher-P
1	SLL	SLL
2	SLL	SLL
3	LLL	SL
4	L	S

S: shuffle/shift, L: logical

Minalpher-P on $*$well (4-way)

	current round			next round	
	p5	p0	p1	p0	p1
1	S_0			C_3	
2	R_0			X_{37}	
3	S_1	C_0			M_{23}
4	R_1			M_1	M_0
5	S_4	C_1			M_3
6	R_4				
7	S_5	X_{04}			
8	R_5				
9	S_2	X_{15}	M_{45}		
10	R_2	M_{01}			
11	S_6	C_2			
12	R_6				
13	S_7	X_{26}	M_6		
14	R_7	M_2			
15	S_3	M_{67}	M_7		
16	R_3	M_4	M_5		

S_i: sbox
R_i: SR_i
C_i: $\oplus$RC
X_{ij}: XM
M_{ij}: MC
M_i: MC

Pros and Cons

	Prøst-128	Minalpher-P
#inst / #ports	3+1/2	4
#regs	16	8
#rounds	16	17.5
format conversion	?	?

Comparison with Rate-1/2 Modes

		∗ Bridge	
Prøst-COPA	10.6 cpb	8-way	DIAC 2014
Minalpher	8.9 cpb	2-way	

Other Numbers of Minalpher (cpb)

μ		message length in bytes				
arch		31	63	1K-1	8K-1	64K-1
Ivy	1	25.21	19.58	14.40	14.05	14.05
Bridge	2	26.97	17.97	9.63	8.94	8.85
Has	1	25.47	19.66	14.00	13.69	13.65
-well	2	26.12	16.97	8.80	8.30	8.23
	4	26.85	17.27	6.33	5.76	5.69

Comments on Skylake

- **Release year 2015-2016 before the announcement of CAESAR final portfolio.**
- **AVX-512**
  - **32 512-bit SIMD registers (`zmm0-zmm31`).**
  - **Ternary logic (`vpternlogd`).**
- **Gather instruction (`vpgather*`) is hoped to be faster.**

