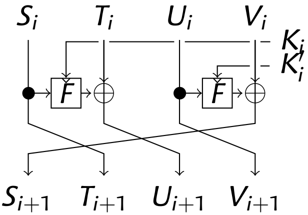
Pseudo-preimage attack against the full SHAvite-3_{512} compression function

aka: s-Pre attack against SHAvite-3_{512}-Compress

Praveen Gauravaram, Gaëtan Leurent, Florian Mendel,
María Naya-Plasencia, Thomas Peyrin,
Christian Rechberger, Martin Schläffer

FSE 2010 Rump Session

SHAvite-3_{512}



- 14 rounds
- Davies-Meyer (message is the key)
- $F_i(x) = AES(AES(AES(AES(x \oplus k_i^0) \oplus k_i^1) \oplus k_i^2) \oplus k_i^3)$

Eli Biham and Orr Dunkelman
The SHAvite-3 Hash Function
Submission to the NIST SHA-3 competition

Cancellation cryptanalysis on generalized Feistels

- $F_i(x) = F(k_i \oplus x)$ with a fixed F
- $\exists c_{i,j} : \forall x,\ F_i(x \oplus c_{i,j}) = F_j(x)$ $(c_{ij} = k_i \oplus k_j)$

- Cancel the effect of the non-linear components
  Using twice the same input pairs

- Fix some parts of the state to reduce the diffusion

Charles Bouillaguet, Orr Dunkelman, Gaëtan Leurent and Pierre-Alain Fouque
Attacks on Hash Functions based on Generalized Feistel
Application to Reduced-Round *Lesamnta* and *SHAvite-3*$_{512}$
ePrint Report 2009/634

Attack Overview

Basic algorithm

- Start from a state in the middle
- Fix some parts of the state to satisfy the cancellation conditions.
- One output word will have a relatively simple expression.
- Invert the expression to choose one word of the output.

- Choose one part of the output
  - Preimage and collision attacks.

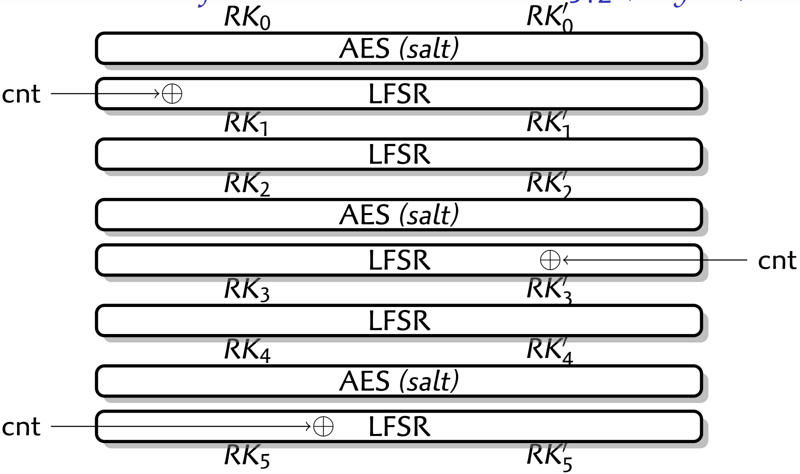
- Mostly generic in the round function.

Cancellation path for the full SHAvite-3_{512}

i	A_i	B_i	C_i	D_i	conditions
3	?	B_3	?	?	
4	?	?	B_3	D_4	
5	D_4	B_5	?	$B_3 + F'_4(D_4)$	$F_5(B_5) = 0$
6	$B_5 + F'_4(D_4)$	D_4	B_3	D_6	$RK_6 = RK'_4$
7	D_6	B_3	D_4	$B_5 + F'_6(D_6)$	$F_7(B_3) = 0$
8	$B_3 + F'_6(D_6)$	D_6	B_3	D_8	$RK_8 = RK'_6$
9	D_8	B_5	D_6	$B_3 + F'_8(D_8)$	$RK_9 = RK_5$
10	$B_5 + F'_8(D_8)$	D_8	B_5	D_{10}	$RK_{10} = RK'_8$
11	D_{10}	B_3	D_8	$B_5 + F'_{10}(D_{10})$	$RK_{11} = RK_7$

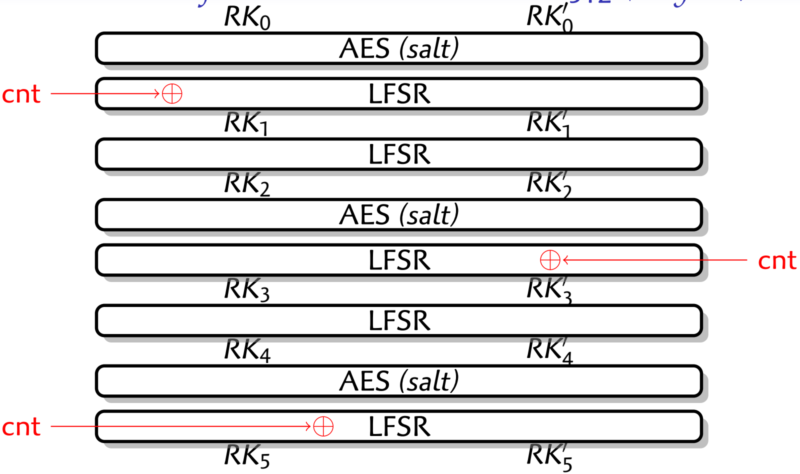
- Only two conditions on the state
- Many conditions on the key

Weak salt for Round-1 SHAvite-3$_{512}$ (Peyrin)



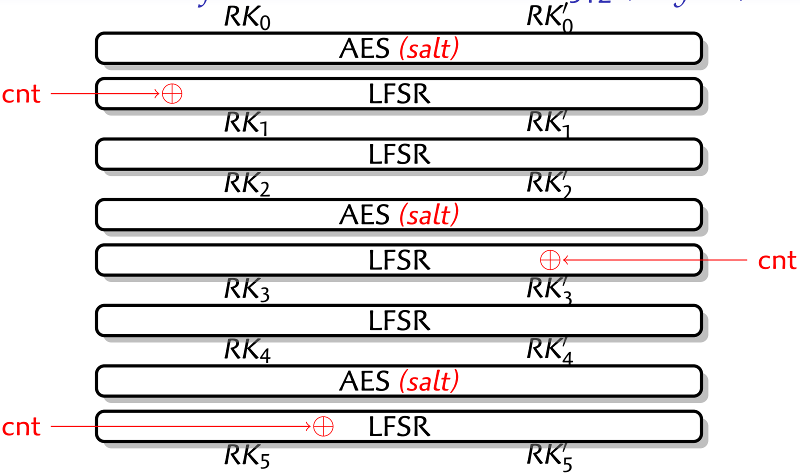
- Take the zero counter;
- Take the salt that sends zero to zero;
- Use the zero message: all the subkeys are zero.

Weak salt for Round-1 SHAvite-3_{512} (Peyrin)



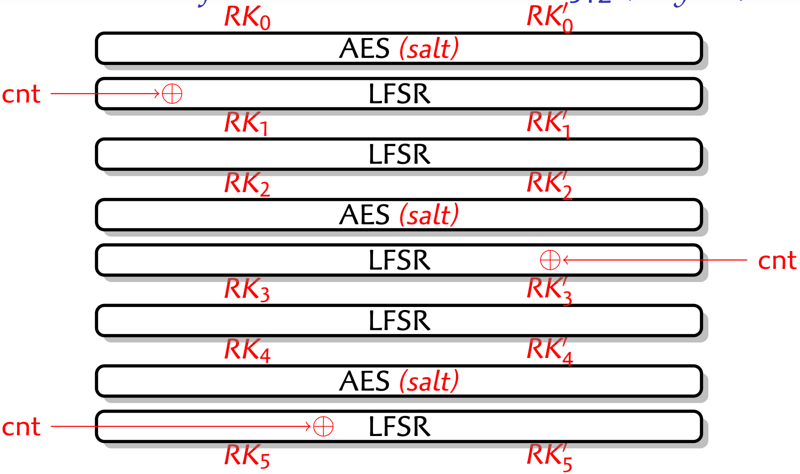
- Take the zero counter;
- Take the salt that sends zero to zero;
- Use the zero message: all the subkeys are zero.

Weak salt for Round-1 SHAvite-3_{512} (Peyrin)



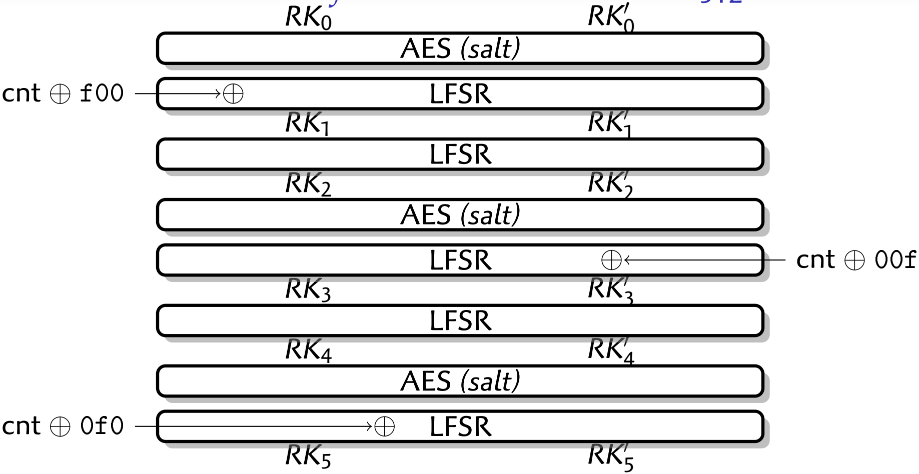
- Take the zero counter;
- Take the salt that sends zero to zero;
- Use the zero message: all the subkeys are zero.

Weak salt for Round-1 SHAvite-3_{512} (Peyrin)



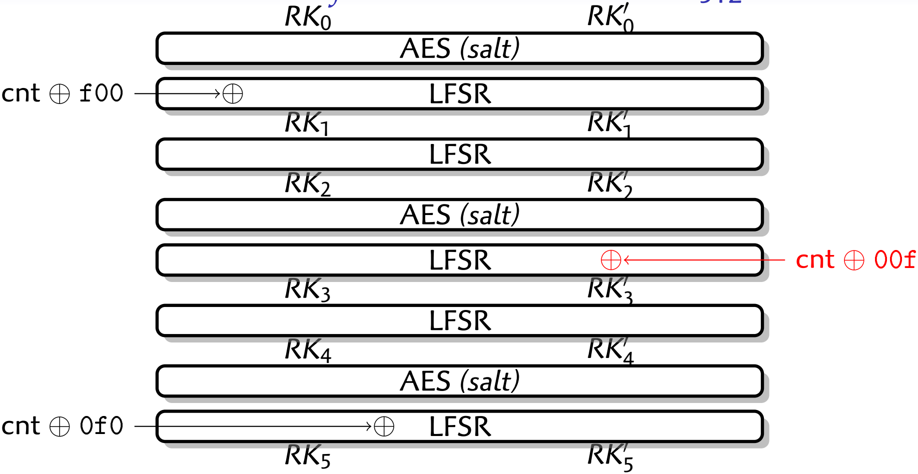
- Take the zero counter;
- Take the salt that sends zero to zero;
- Use the zero message: all the subkeys are zero.

Weak salt for Round-2 SHAvite-3$_{512}$



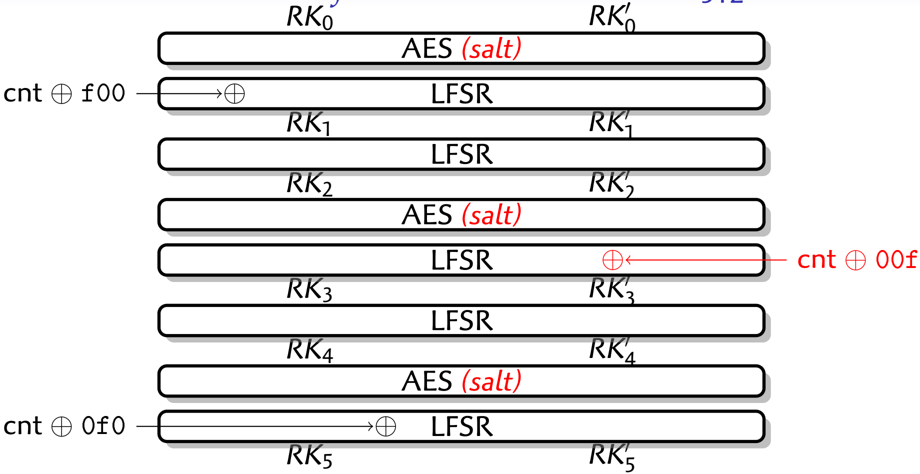
- Cancel one counter in the middle;
- Take the salt that sends zero to zero;
- Use the zero subkey in the middle.

Weak salt for Round-2 SHAvite-3_{512}



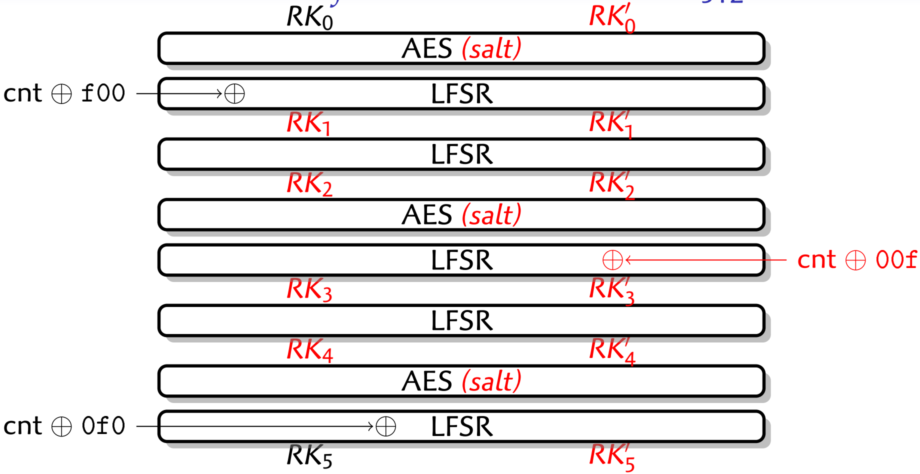
- Cancel one counter in the middle;
- Take the salt that sends zero to zero;
- Use the zero subkey in the middle.

Weak salt for Round-2 SHAvite-3_{512}



- Cancel one counter in the middle;
- Take the salt that sends zero to zero;
- Use the zero subkey in the middle.

Weak salt for Round-2 SHAvite-3_{512}



- Cancel one counter in the middle;
- Take the salt that sends zero to zero;
- Use the zero subkey in the middle.

Weak salt for Round-2 SHAvite-3$_{512}$

i	RK_i				RK'_i				r
	$k^0_{0,i}$	$k^1_{0,i}$	$k^2_{0,i}$	$k^3_{0,i}$	$k^0_{1,i}$	$k^1_{1,i}$	$k^2_{1,i}$	$k^3_{1,i}$	
0	?	?	?	?	?	?	?	?	M
1	?★	?	?	?	?	?	?	0	1
2	0	?	?	?	?	0	0	0	
3	0	?	?	?	0	0	0	0	2
4	0	?	0	0	0	0	0	0	
5	0	0★	0	0	0	0	0	0	3
6	0	0	0	0	0	0	0	0	
7	0	0	0	0	0	0	0	0	4
8	0	0	0	0	0	0	0	0	
9	0	0	0	0★	0	0	0	0	5
10	0	0	0	0	0	0	0	0	
11	0	0	0	0	0	0	0	0	6
12	0	0	0	0	0	0	0	0	
13	0	0	0	0	0	0	?★	?	7

The attack on the full SHAvite-3_{512}

```
Input: Target value H
Output: message, chaining value, salt, counter
repeat
    Take a random weak salt, and the corresponding message
    Compute $2^{128}$ states with 128 chosen output bits
until a full preimage is found ($2^{256}$ iterations)
```

- Pseudo-preimage attack: complexity 2^{384} and 2^{128} memory
- Pseudo-preimage attack: complexity 2^{448} without memory
- Pseudo-collision attack: complexity 2^{192} and 2^{128} memory.

The attack on the full SHAvite-3_{512}

```
Input: Target value H
Output: message, chaining value, salt, counter
repeat
    Take a random weak salt, and the corresponding message
    Compute $2^{128}$ states with 128 chosen output bits
until a full preimage is found ($2^{256}$ iterations)
```

- Pseudo-preimage attack: complexity 2^{384} and 2^{128} memory
- Pseudo-preimage attack: complexity 2^{448} without memory
- Pseudo-collision attack: complexity 2^{192} and 2^{128} memory.