Higher Order Differential Attack on Step-Reduced Variants of *Luffa* v1

@Fast Software Encryption, FSE 2010
9th February 2010

Dai Watanabe
Yasuo Hatano

Systems Development Laboratory, Hitachi, Ltd.

Tsuyoshi Yamada
Toshinobu Kaneko

Science University of Tokyo

Higher order differential attack

- 1994, Lai
  - Basic properties of higher order difference [10]
- 1994, Knudsen
  - Attack on block ciphers [9]
- 2008, Dinur and Shamir
  - Application to stream ciphers [8]
  - A new name given: Cube attack
- 2009, Aumasson et al. [1]
  - Cube tester
- 2009, Aumasson and Meier [2]
  - Zero-sum attack
- 2010, Watanabe et al.
  - This work

Outline

- Specification of *Luffa* v1
  - Chaining
  - Non-linear components
- Algebraic degree of the permutation Q_j
- Distinguishing attack on 7-steps *Luffa* v1
  - A way to ignore other components
  - Practicality of the attack
- Conclusion

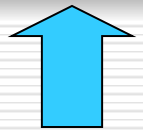
Motivation of our research

- What is *Luffa*?
  - One of the SHA-3 2nd round candidates
  - Thin step function
    - 64 4-bit Sboxes
      + Linear map consisting of XORs and Rotations
  - It changed the algorithm at the beginning of the Round 2
    - Our target is *Luffa* v1, not *Luffa* v2
- Evaluations
  - Differential attack: done
  - Algegraic attack: **none**

Designer's claim

Note that the number of monomials which appears in the polynomial representation is smaller than that of a randomly generated Sbox.

Though one might claim that this Sbox is weak in terms of algebraic attacks, we have not found any practical attack on *Luffa* using this property.



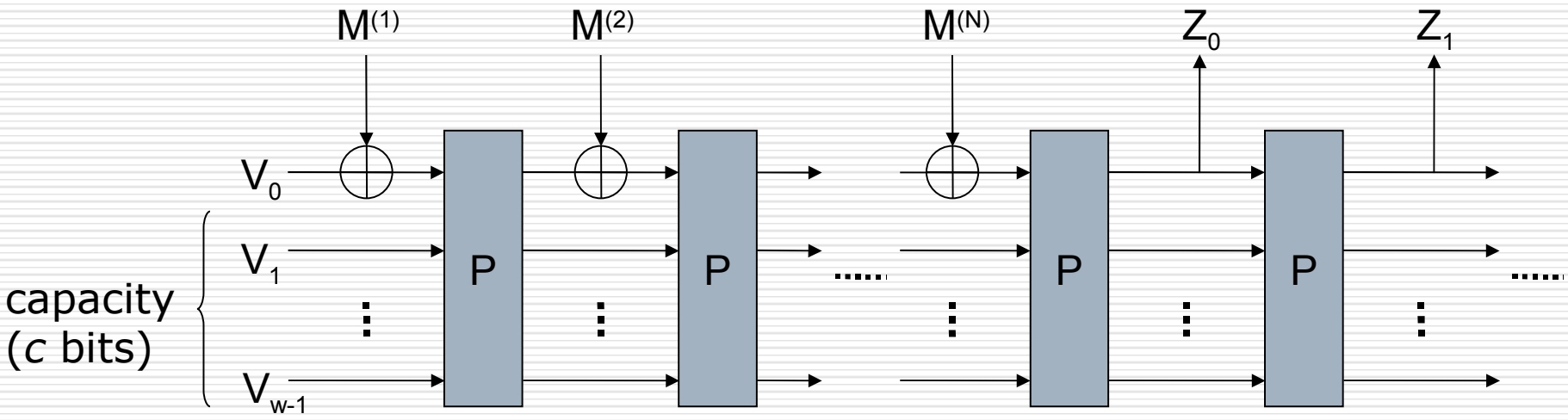
Should be investigated!

Our contribution

- The algebraic degree of the underlying non-linear permutation is investigated.
- Based on the investigation, a "distinguishing attack" on 7 steps *Luffa* v1 is proposed.
  - The XORing of 2^{216} messages is always zero.
  - If the function has 256-bit input and it is highly non-linear, this property is not expected.
  - The practicality of the attack is controversial. It will be discussed at the end of this talk.

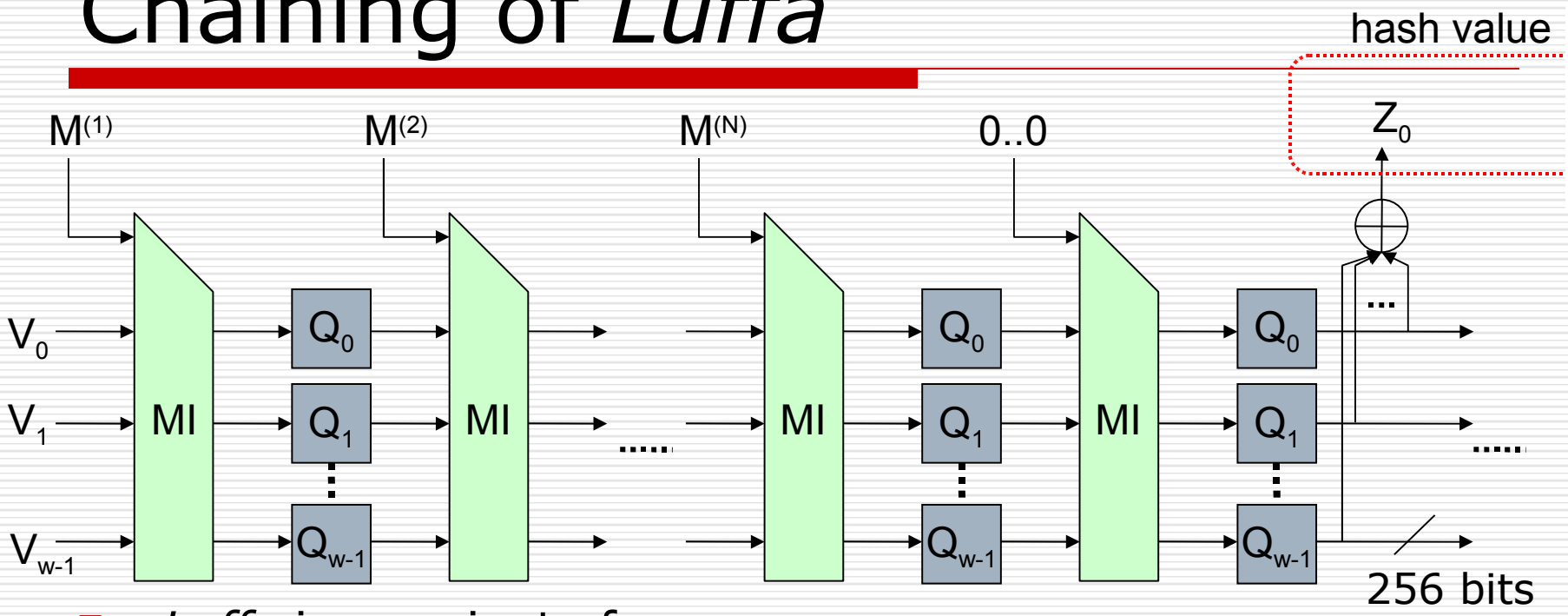
Specification of to *Luffa* v1

Cryptographic sponge function



- Novel construction of a hash function from a permutation
- It is proved to be indifferentiable from a RO

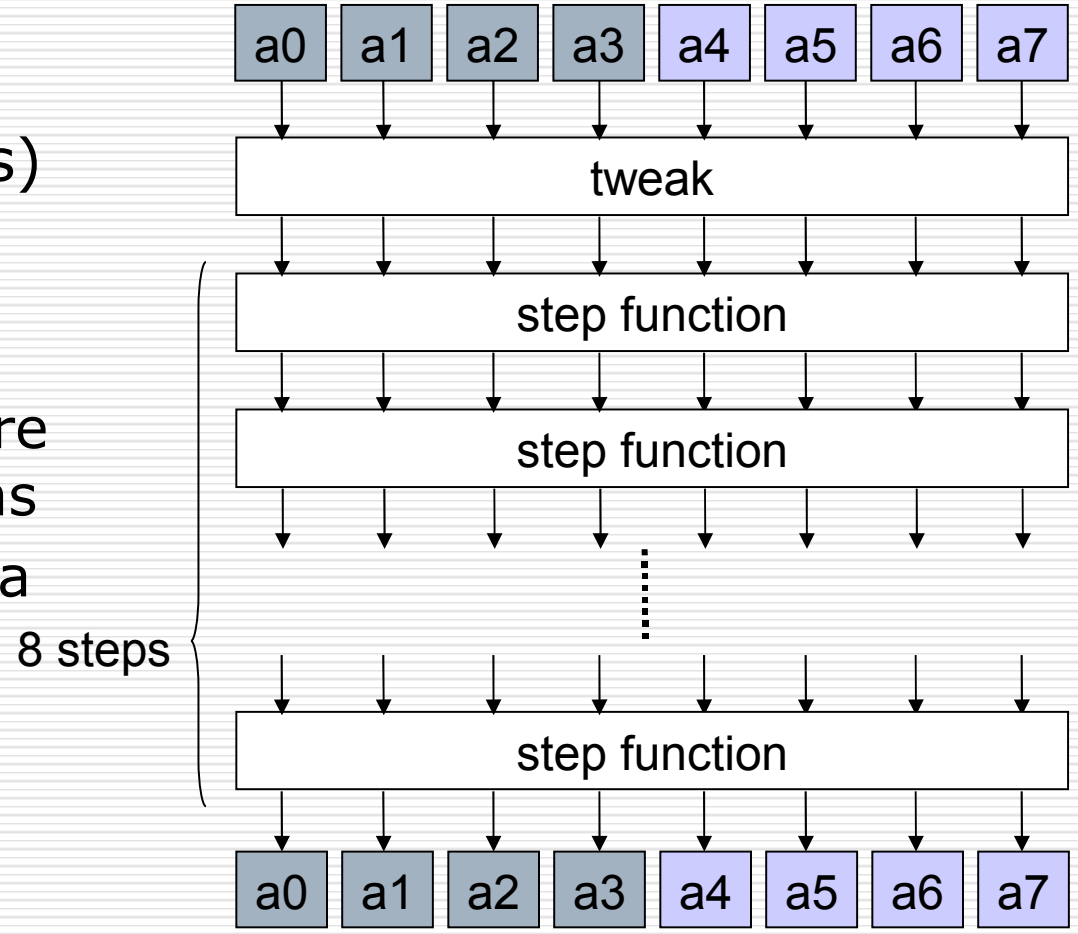
Chaining of *Luffa*



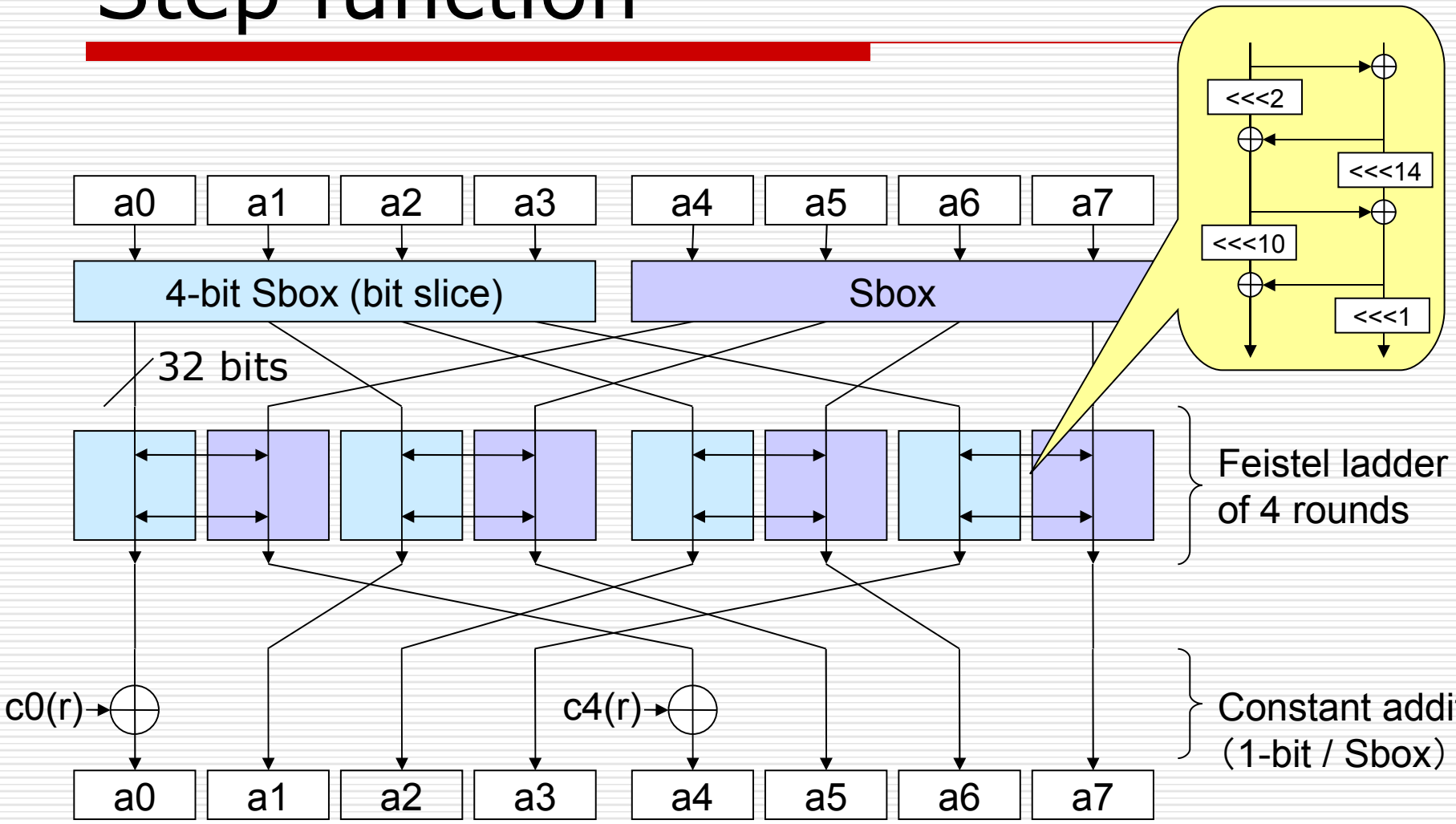
- *Luffa* is a variant of sponge
  - But, fixed length permutations for all hash length
    - The number of Qj increases if the hash length gets long (w=3, 4, 5 for hash_len=256, 384, 512)
  - Insert message and mix the state by the linear map MI
  - A blank round
  - The hash value is the XORing of the outputs of Qj

Non-linear permutation Q_j

- Input/Output
  - 256 bits
    (8 32-bit words)
- Functions
  - tweak
    - Applied before step functions
    - Rotations in a word
  - Step functions
    - 8 steps



Step function



Algebraic degree of Q_j

ANFs of the Sbox of *Luffa* v1

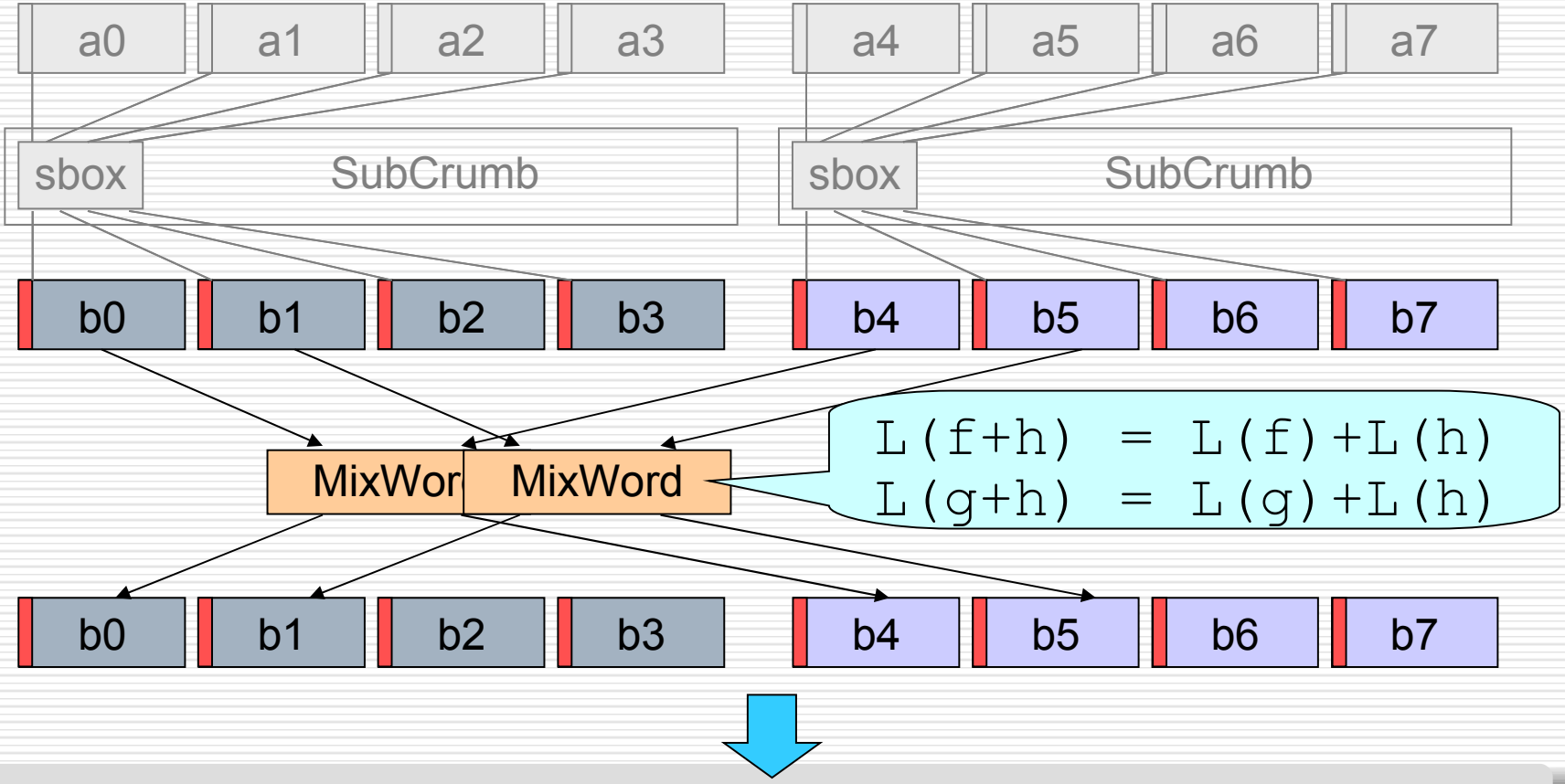
- $b_0 = 1 + a_2 + a_0a_1$
  $\mathbf{+ a_1a_3 + a_2a_3 + a_0a_1a_3}$
- $b_1 = 1 + a_0 + a_2 + a_0a_1 + a_0a_2 + a_3$
  $\mathbf{+ a_1a_3 + a_2a_3 + a_0a_1a_3}$
- $b_2 = 1 + a_1 \mathbf{+ a_1a_3 + a_2a_3 + a_0a_1a_3}$
- $b_3 = a_0 + a_1 + a_2$
  $+ a_0a_1 + a_1a_2 + a_0a_1a_2 + a_1a_3$

Most of the high degree terms are the same.

Results lead by the property

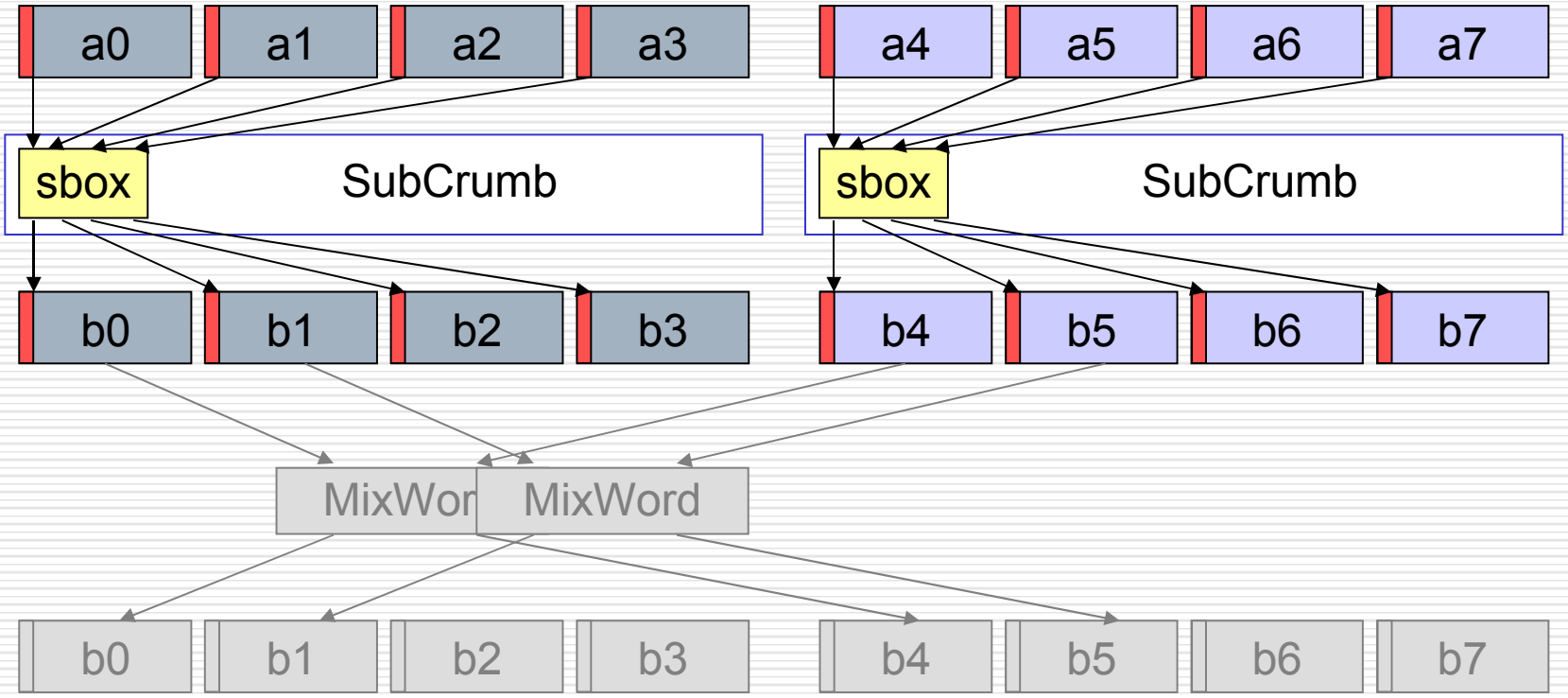
- The increase of the algebraic degree in the iteration of step functions is slower than expected.
- The XORing b_0+b_1 has lower degree than b_0 and b_1.

Property preservation in MixColumn()



The property, that the terms of high degree are the same, is preserved by MixWord().

Increase of the degree in the Sbox



$$(f(x)+h(x)) \cdot (g(x)+h(x)) = f(x) \cdot g(x) + (f(x)+g(x)+1) \cdot h(x)$$

- $\deg((f+h) \cdot (g+h)) < 2 \deg(h)$

Increases of algebraic degrees

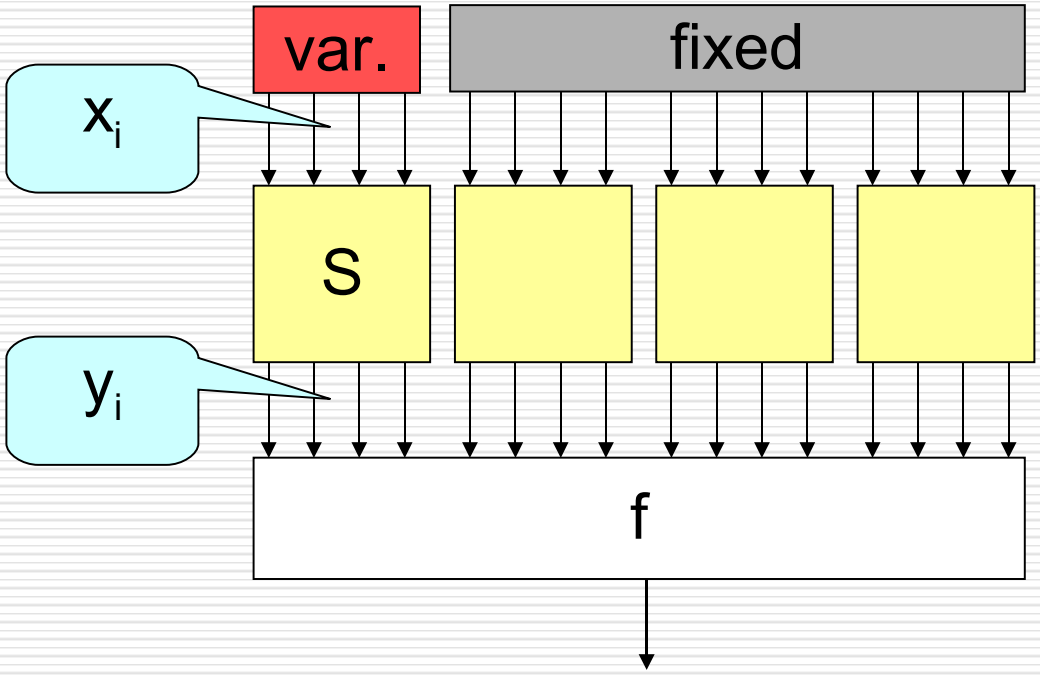
		Algebraic degrees			
		a0		a0+a1	
		estimate	experiment	estimate	experiment
# of steps	0	1	-	1	-
	1	3	1	2	2
	2	8	7	5	5
	3	20	18	13	12
	4	51	-	33	≥ 32
	5	130	-	84	-
	6	331	-	214	-
	7	843	-	545	-
	8	2147	-	1388	-

How to observe the algebraic degree

- Higher order differential characteristic
  - $f(x_1,\ldots, x_n) = x_1\, g(x_2,\ldots,x_n) + h(x_2,\ldots,x_n)$
  - $\Delta_1 f(x_1,\ldots, x_n)$
    $= g(x_2,\ldots, x_n)$
  - $= f(x_1+1, x_2, \ldots, x_n) + f(x_1, x_2, \ldots, x_n)$
  - $\Delta_{<x1,\ldots, xk>} f(x_1,\ldots, x_n)$
    $= \Delta_k \ldots \Delta_2 \Delta_1 f(x_1,\ldots, x_n)$
    $= \Sigma_{a\in<x1,\ldots, xk>} f(x_1+a_1,\ldots, x_k+a_k, x_{k+1}, \ldots, x_n)$
- Feature
  - $\deg(\Delta_i f) \leq \deg(f)-1$
  - $\Delta_{<xi1,\ldots, xik>} f(x_1,\ldots, x_n) = 0$ for all $\{x_{i1},\ldots, x_{ik}\}$
    $\Rightarrow \deg(f) = k-1$

Attack on reduced step variants of *Luffa*

A tip to skip a permutation



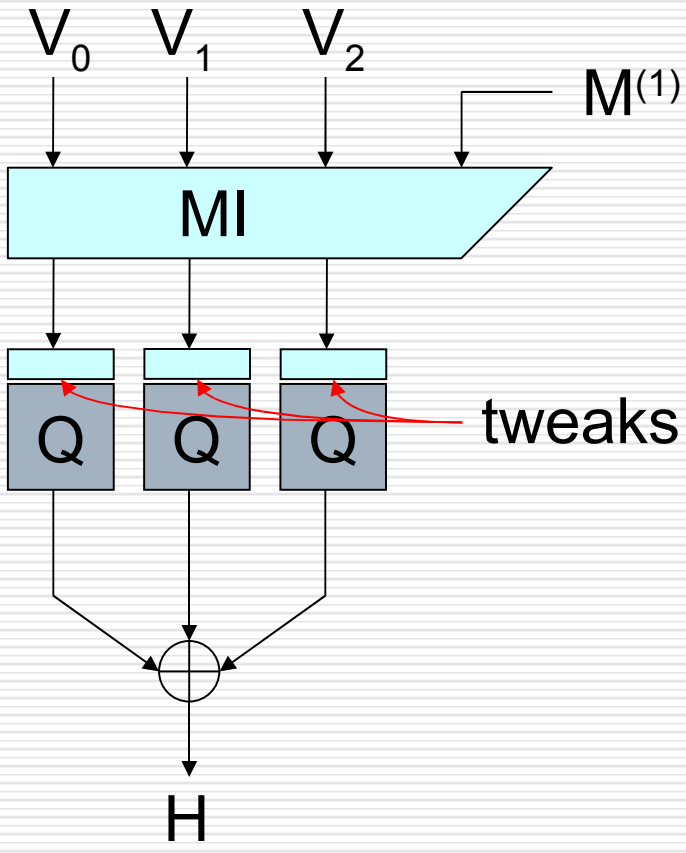
- For a permutation S, $\{x_i\}_i = \{y_i\}_i$
- $\Sigma_i f(y_i)=0 \Rightarrow \Sigma_i f(x_i)=0$

How many steps can be attacked?

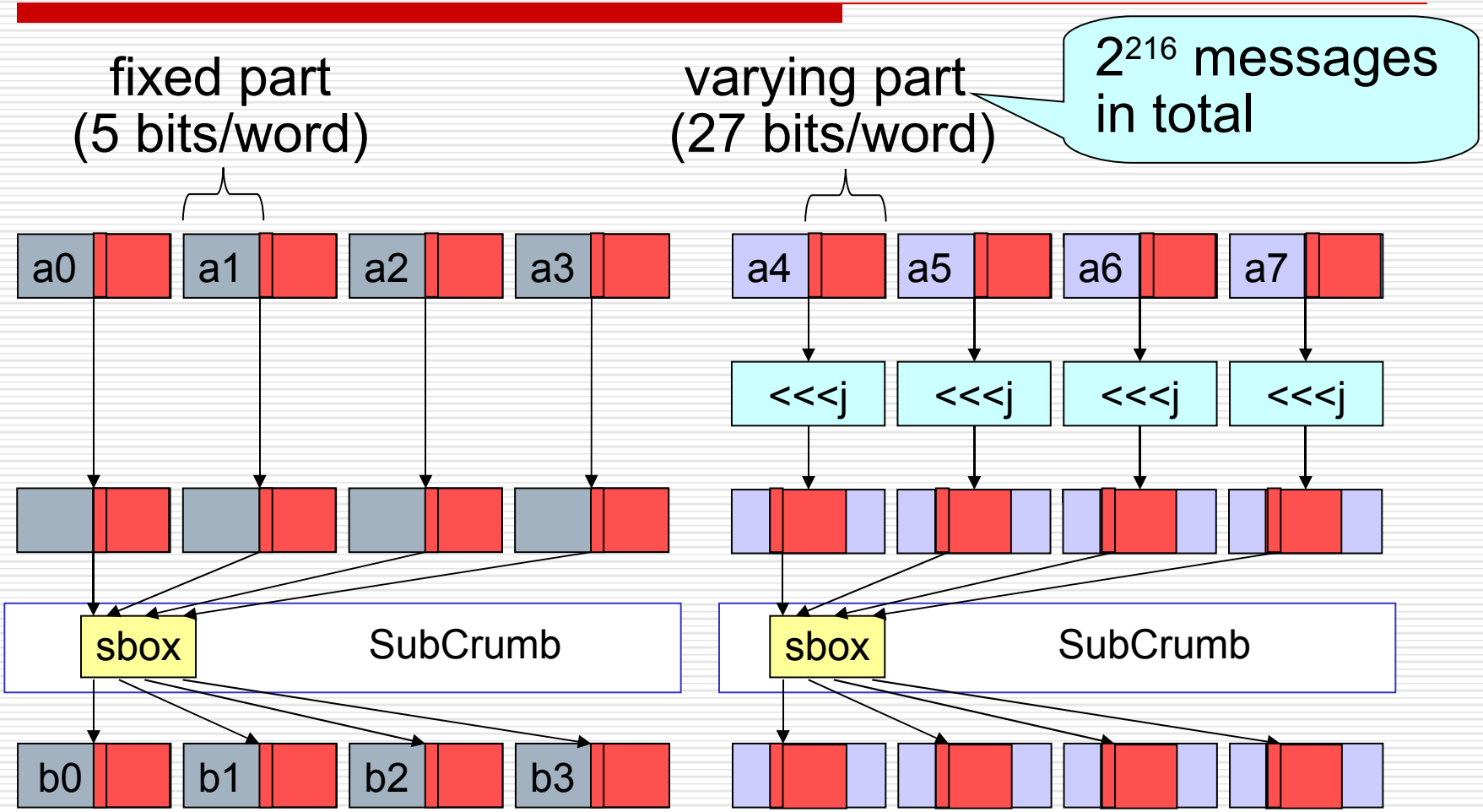
		Algebraic degrees			
		a0		a0+a1	
		estimate	experiment	estimate	experiment
# of steps	1	1	-	1	-
	2	3	1	2	2
	3	8	7	5	5
	4	20	18	13	12
	5	51	-	33	≥ 32
	6	130	-	84	-
	7	331	-	214	-
	8	843	-	545	-
	8	2147	-	1388	-

Luffa for a block message

- Different procedure for a block message
  - A blank round is not applied if the message length is less than 256 bits.
- Additional components
  - Message injection function MI
  - Tweaks

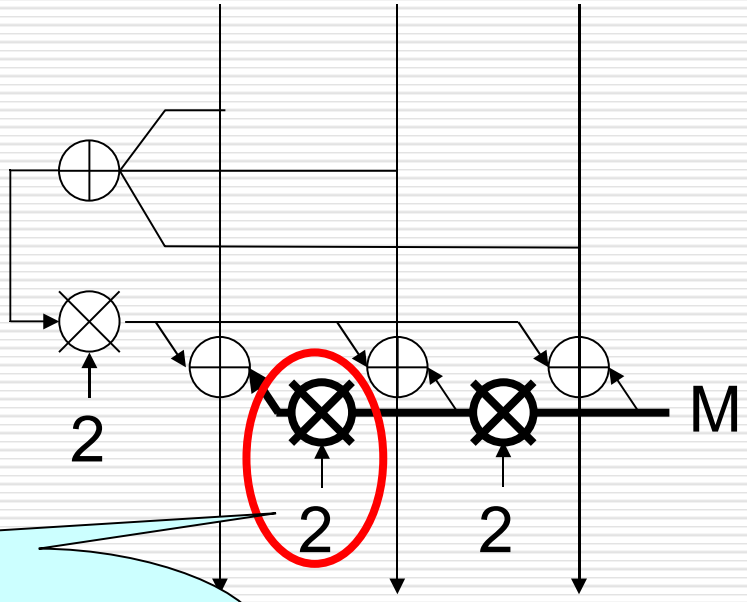


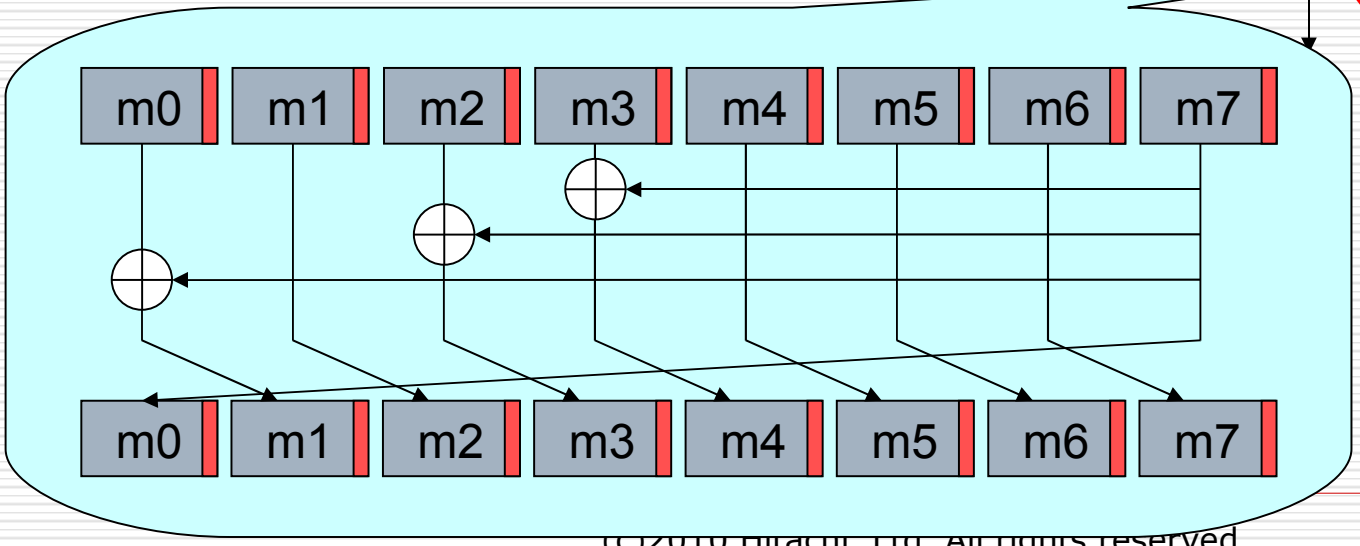
Choice of the set of inputs to Q_j



Skip Message injection *MI*

- The multiplication is defined over $GF(2^8)^{32}$
- It is surjective on the subset





How to use the “distinguisher”?

Q1. A hash function does not have a secret key. No “distinguisher” is possible (in general).
A1. Consider only the keyed applications like MACs.

Q2. The length of the message must be less than 256. The attacker has only 255-216=39 bits freedom for the key. Does 2^{216} complexity make sense?
A2. It may make sense if it is allowed to deal with the IV as a parameter.

Attack with 6-steps distinguisher

- Setting
  - Consider a MAC algorithm MAC(K,M)=h(K||M), it is distinguishable from a random function with 2^{84} chosen messages.
- Is it practical?
  - HMAC(K,M) = h((K⊕opad)||h(K ⊕ ipad)||M))
  - We can apply the distinguisher to h(K ⊕ ipad)||M)).
  - But the output transformation prevents the application of the attack.

Attack with 7-steps distinguisher

- Setting
  - If IV is dealt with as a parameter, the family of hash functions is distinguishable from a random function with 2^{216} chosen messages.
- Is it practical?
  - ISO9797-2 MAC Algorithm 1
  - In which h'(M)=h(g(K),M) is used, where the original IV is replaced by the key dependent constant g(K).
  - The output transformation prevents the application of the attack.

Attack on *Luffa* v2

- What are changed?
  - 1. Sbox
  - 2. Order of the inputs to SubCrumb()
  - 3. A blank round is **always** applied
- The result
  - 1 and 2 improve the property of Q_j.
  - 3 is the essential improvement more than 1 and 2. The number of rounds to be attacked becomes double.

Summary

The algebraic degree of the underlying non-linear permutation is investigated.

A distinguishing attack on 7 steps *Luffa* v1 is proposed. The attack requires 2^{216} messages.

The practical application of the attack has not been not found.

Extension of the attack to *Luffa* v2 seems difficult.

Thank you for attention!