

The LANE hash function

Sebastiaan Indestege
sebastiaan.indestege@esat.kuleuven.be

COSIC, ESAT/SCD, K.U. Leuven, Belgium

Dagstuhl
12 January 2009

Contributors:

Elena Andreeva, Christophe De Cannière, Orr Dunkelman,
Emilia Käsper, Svetla Nikova, Bart Preneel,
Elmar Tischhauser.

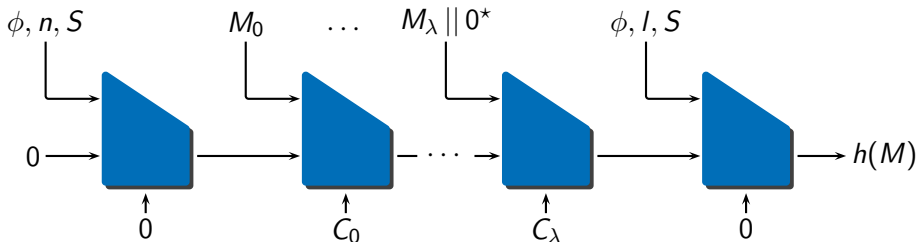
LANE



- is a candidate in the **NIST SHA-3 competition**
- is **simple**, elegant, easy to understand and analyse.
- reuses **AES**-rounds as a building block.
- can support a **salt** value, if desired.
- has a clear **design rationale**.
- has undergone an extensive **security analysis**.
- is flexible in **implementation**.
- ...

Description of LANE

Iteration mode

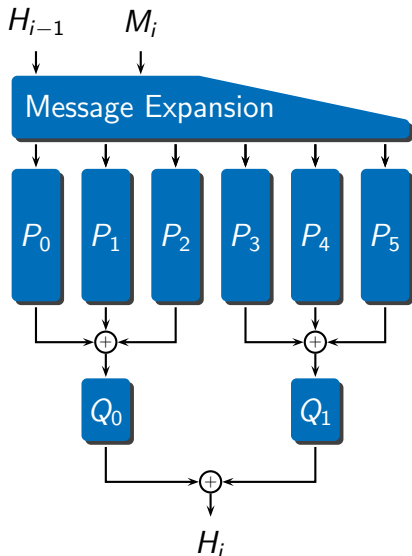


- Very simple and lightweight
- Supports salt value, if desired
- Bit counter (*HAIFA*)
- Output transformation
- IV derivation (*can be precomputed*)

ϕ flag byte
 n output length
 l message length
 S salt value
 M_i message block
 C_i bit counter

Description of LANE

Compression Function



- **Message expansion**
- 6 parallel **P-lanes**
(6 resp. 8 rounds)
- 2 parallel **Q-lanes**
(3 resp. 4 rounds)
- XOR combiners
- $|H_i| = 256$ resp. 512 bits
- $|M_i| = 512$ resp. 1024 bits

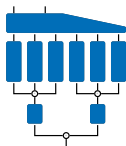
Description of LANE

Message expansion

- **Simple**, lightweight, linear

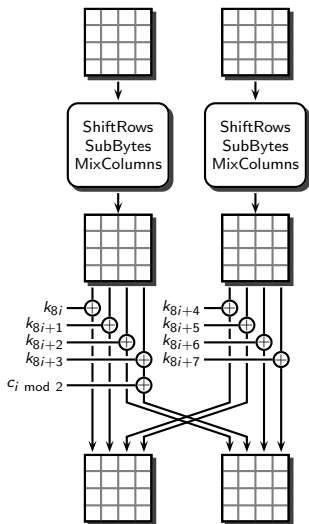
$$[W_0 \parallel \dots \parallel W_5] = [H \parallel M^h \parallel M^l] \cdot \begin{bmatrix} 1 & 0 & / & / & / & / & / & 0 & 0 & 0 & 0 & 0 \\ 0 & / & / & 0 & / & 0 & 0 & / & 0 & 0 & 0 & 0 \\ / & / & / & 0 & / & / & 0 & 0 & / & 0 & 0 & 0 \\ / & 0 & 0 & / & / & 0 & 0 & 0 & 0 & 0 & / & 0 \\ / & / & / & / & / & 0 & 0 & 0 & 0 & 0 & / & 0 \\ / & 0 & / & 0 & 0 & / & 0 & 0 & 0 & 0 & 0 & / \end{bmatrix}$$

- Easy and fast to implement
(*XOR of 128 resp. 256-bit blocks*)
- **Parallellisable**
- Ensures **minimum 4 active lanes**
(*linear code over GF(4) with minimum distance 4*)



Description of LANE

LANE-256 permutations



- **From AES:**

ShiftRows, SubBytes, MixColumns

- **New:**

AddConstants, AddCounter, SwapColumns

- Constants k_i generated using LFSR
- LANE-512 similar

Design rationale: Keep it simple!

- Use bit counter (HAIFA) and salt.
- Output transformation: small cost, helps protect against first preimage attacks, length extension and distinguishers.
- Reuse AES rounds
- Permutations, so no local collisions
- Message expansion: very light, ≥ 4 active lanes, stops straightforward inversion, stops MitM
- Ample parallelism, but also low-memory serial implementation possible
- Constants diversify the permutations
- ...



Security analysis

- Differential cryptanalysis
- Truncated differential cryptanalysis
- Higher order differential cryptanalysis
- Algebraic attacks
- Attacks based on reduced query complexity
- Generalised birthday attack
- Meet-in-the-middle attacks
- Long message second-preimage attacks
- Length-extension attacks
- Multicollision attacks
- ...

Refer to the [submission document](#)



Implementation of LANE

LANE is flexible in implementation

- Can reuse techniques for implementing **AES** (*table lookups, bitslicing, ...*)
- **LANE + AES** = code/ROM/hardware sharing!
- Current best speed on Intel Core2 (LANE-256): **25.7 cpb**
- **Intel AES-NI** instruction set: LANE-256 at ≈ 5 cpb?
- **embedded systems**: low memory possible
only 108 bytes of RAM needed for LANE-256
- **Hardware** (LANE-256 0.13 μ m CMOS):
16 462 GE @ 23.3 Mbps — 243 486 GE @ 14.2 Gbps



The LANE hash function

<http://www.cosic.esat.kuleuven.be/lane/>

Designer: Sebastiaan Indestege

Contributors: Elena Andreeva, Christophe De Cannière,
Orr Dunkelman, Emilia Käsper, Svetla Nikova,
Bart Preneel, Elmar Tischhauser