

The LANE Hash Function

Sebastiaan Indesteege

`sebastiaan.indesteege@esat.kuleuven.be`

COSIC, ESAT/SCD, Katholieke Universiteit Leuven, Belgium

First SHA-3 Candidate Conference

Contributors:

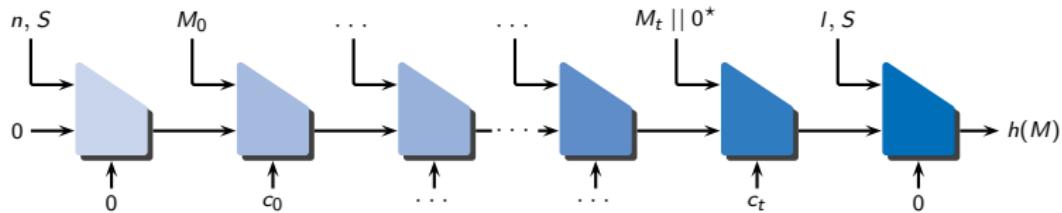
Elena Andreeva, Christophe De Cannière, Orr Dunkelman,
Emilia Käsper, Svetla Nikova, Bart Preneel, Elmar Tischhauser

LANE



- is **simple**, elegant, easy to understand and analyse.
- has a clear **design rationale**.
- has undergone an extensive **security analysis**.
- is **flexible** in implementation.

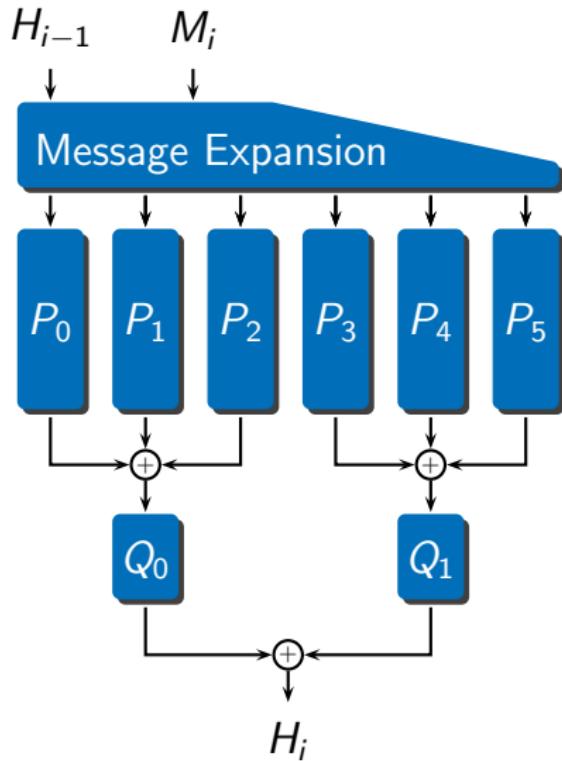
Description of LANE



Iteration Mode

- Very **simple** and lightweight
- **Features:** bit counter, output transformation, salt (*opt.*)
- **Security:** No length extension attacks, no long message second preimage attacks, indistinguishable from a random oracle (prefix-free encoding), PRF preserving (MACs), ...

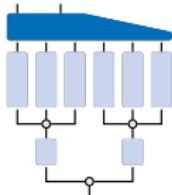
Description of LANE



Compression Function

- Simple structure
 - Message expansion
 - 6 **P**-lanes
(6 resp. 8 rounds)
 - 2 **Q**-lanes
(3 resp. 4 rounds)
 - XOR combiners
- Parallelism
(but low memory possible)

Description of LANE

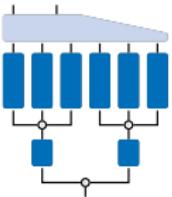
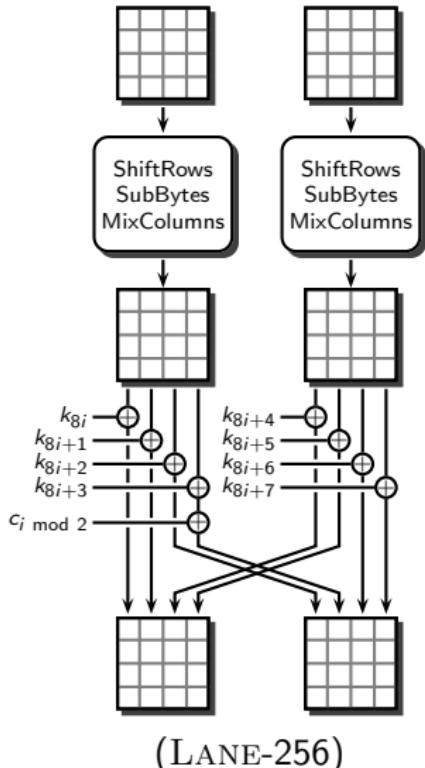


$$[W_0 \parallel \dots \parallel W_5] = [H \parallel M^h \parallel M^l] \cdot \begin{bmatrix} 1 & 0 & / & / & / & / & 0 & 0 & 0 & 0 \\ 0 & / & 1 & 0 & / & 0 & 0 & / & 0 & 0 & 0 \\ / & / & 1 & 0 & / & 1 & 0 & 0 & / & 0 & 0 \\ / & 0 & 0 & / & 1 & 0 & 0 & 0 & 0 & / & 0 \\ / & / & / & 1 & 1 & 0 & 0 & 0 &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

Message Expansion

- **Simple**, lightweight, parallelisable, linear
- Easy and fast to implement (*XOR of large blocks*)
- Ensures **minimum 4 active lanes**
(linear code over GF(4) with minimum distance 4)
- Stops straightforward inversion and meet-in-the-middle

Description of LANE



Permutation 'lanes'

- From AES:
ShiftRows, SubBytes, MixColumns
- New:
*AddConstants, AddCounter,
SwapColumns*
- Constants k_i generated by LFSR
- LANE-512 similar

Security of LANE

- LANE has undergone an **extensive 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 **supporting documentation:**
 -  S. Indesteege, E. Andreeva, C. De Cannière, O. Dunkelman, E. Käsper, S. Nikova, B. Preneel, E. Tischhauser
The LANE Hash Function



Security of LANE

Example: why standard differential cryptanalysis fails

- LANE-256
- Any differential characteristic Q
 - ≥ 4 active P -lanes
 - ≥ 45 active S-boxes per lane
 - $\Pr \leq 2^{-6}$ per active S-box
 - $\Rightarrow \Pr(Q) \leq 2^{-1080}$
- Assume perfect message modification
 - 832 degrees of freedom
 - $\Rightarrow \Pr(\langle m, m' \rangle \in Q) \leq 2^{-248}$
- Very unlikely that a right pair even exists for a given characteristic Q !



Implementation of LANE

- LANE is **flexible in implementation**
- Reuse techniques for implementing **AES**
- **LANE + AES** = share code/ROM/hardware
- Roughly **half** the speed of AES:

• {	AES-128	128 bits	10 AES rounds
	LANE-256	512 bits	84 AES rounds
• {	AES-256	128 bits	14 AES rounds
	LANE-512	1024 bits	224 AES rounds

×0.48

×0.50



Implementation of LANE



Performance results

- **Intel Core2**: 25.7 cpb (LANE-256)
- **Intel AES-NI**: LANE-256 at 5 cpb ?
- **Embedded systems**: 108 bytes of RAM (LANE-256)
- **Hardware** (LANE-256, 0.13 μ m CMOS):
16 kGE @ 23.3 Mbps — 243 kGE @ 14.2 Gbps

Conclusion

The LANE hash function

- is **simple**, elegant, easy to understand and analyse.
- has a clear **design rationale**.
- has undergone an extensive **security analysis**.
- is **flexible** in implementation.

Designer: Sebastiaan Indesteege

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

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