

# CAESAR candidate Marble

Jian Guo

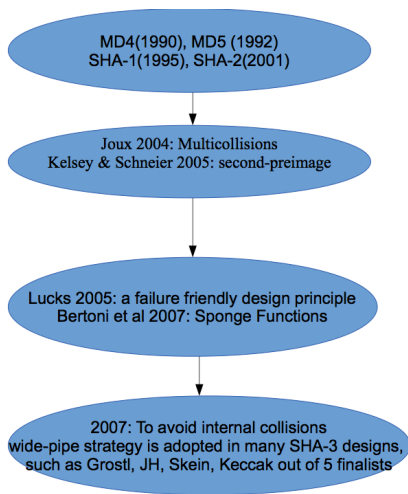


DIAC – 24 August 2014  
@Santa Barbara, CA, USA

# Design Goals

- ▶ Online
- ▶ Parallelizable
- ▶ Software oriented
- ▶ Decryption-misuse resistant, unverified plaintext release
- ▶ Nonce-misuse resistant, or nonce-free
- ▶ Low setup overhead
- ▶ Support of extreme usecases
- ▶ Full security

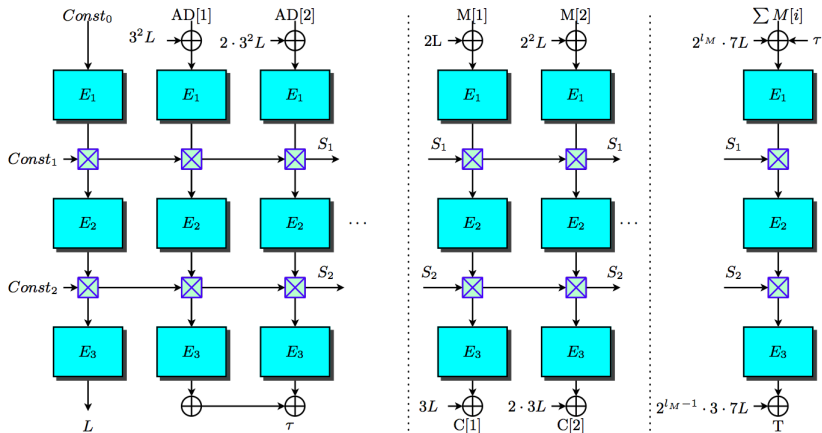
# The need of “wide-pipe”



## Lesson from hash function development

use double or even larger internal state to avoid internal collisions

# Design Overview



- ▶  $E_1, E_2, E_3$  are block-ciphers
- ▶  $\text{TRANS}(x, y)$ : a transition function with MDS property.
- ▶ ‘ $\cdot$ ’ multiplication is in  $\text{GF}(2^{128})$ .

# Recommended Parameters

**Choices** are made to optimize the software performance:

- ▶  $E_1, E_2, E_3$  are 4-round AES, every message block is processed by 12 AES rounds.
- ▶  $\text{TRANS}(x, y) = (x + y, 3 \cdot x + y)$ , **division-free** for the inverse computation.

# Recommended Parameters

**Choices** are made to optimize the software performance:

- ▶  $E_1, E_2, E_3$  are 4-round AES, every message block is processed by 12 AES rounds.
- ▶  $\text{TRANS}(x, y) = (x + y, 3 \cdot x + y)$ , **division-free** for the inverse computation.

achieve a speed of **1.6 cpb** for long message and **1.7 cpb** for 8KB message, tested on Intel(R) Core(TM) i5-4570 CPU @ 3.20GHz (Haswell Family), **12 rounds AES takes 0.6 cpb only, room to improve.**

# Recommended Parameters

**Choices** are made to optimize the software performance:

- ▶  $E_1, E_2, E_3$  are 4-round AES, every message block is processed by 12 AES rounds.
- ▶  $\text{TRANS}(x, y) = (x + y, 3 \cdot x + y)$ , **division-free** for the inverse computation.

achieve a speed of **1.6 cpb** for long message and **1.7 cpb** for 8KB message, tested on Intel(R) Core(TM) i5-4570 CPU @ 3.20GHz (Haswell Family), **12 rounds AES takes 0.6 cpb only, room to improve.**

## Options

- ▶ support the use of 128-bit nonce, by prepending it to the associated data.

# Recommended Parameters

**Choices** are made to optimize the software performance:

- ▶  $E_1, E_2, E_3$  are 4-round AES, every message block is processed by 12 AES rounds.
- ▶  $\text{TRANS}(x, y) = (x + y, 3 \cdot x + y)$ , **division-free** for the inverse computation.

achieve a speed of **1.6 cpb** for long message and **1.7 cpb** for 8KB message, tested on Intel(R) Core(TM) i5-4570 CPU @ 3.20GHz (Haswell Family), **12 rounds AES takes 0.6 cpb only, room to improve.**

## Options

- ▶ support the use of 128-bit nonce, by prepending it to the associated data.
- ▶ Better security margin with AES for  $E_1, E_2, E_3$ , yet with a speed of **3.0 cpb**.



In addition to the usual use, Marble supports many extreme usecases:

- ▶ Encryption/Decryption only (opting out the tag)
- ▶ Integrity of associated data only.
- ▶ Integrity of message — MAC only (opt out the ciphertext).

# Security Goals

$2^n$  security, not “birthday bound”, in **both** nonce-respecting and nonce-misuse scenarios.

Privacy	$2^{128}$
Authenticity	$2^{128}$

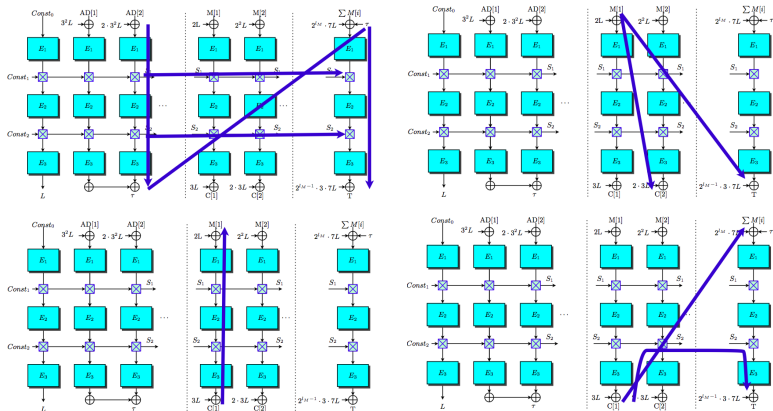
# Security Goals

$2^n$  security, not “birthday bound”, in **both** nonce-respecting and nonce-misuse scenarios.

Privacy	$2^{128}$
Authenticity	$2^{128}$

Privacy in nonce-misuse scenario: prefixed message blocks share the same ciphertext prefix.

# Security Evaluations



- ▶ Differential/Linear Cryptanalysis: any complete path will involve at least **12** rounds AES, with 75 active sboxes.
- ▶ Inner collisions: collision on single chain is NOT “detectable”; collision on double chains requires  $2^n$ .
- ▶ Nandi’s attack does not apply even with complexity  $2^n$  due to the  $2n$ -bit chain.

We welcome security proof of Marble mode, when the three block ciphers are idealized.

- ▶ Hardware implementations
- ▶ Improving the software implementations with AES-NI
- ▶ Implementations without AES-NI
- ▶ Implementations for Atmel AVR
- ▶ Security proof when the underlying blockciphers are ideal, extend tag-splitting to arbitrary-length message to avoid XLS.

Thank you!

Questions?