SIMD in the SHA-3 Competition

Gaëtan Leurent, Charles Bouillaguet, Pierre-Alain Fouque

École Normale Supérieure
Paris, France

Second SHA-3 Conference

Outline

Speed vs Security

NIST wants SHA-3 to be faster and more secure than SHA-2.

- More secure: more operations
- Faster: less time

- We need to cheat (use the hardware more efficiently)
  - Use AES instructions (*e.g.* ECHO, SHAvite-3)
  - Use 64-bit integers (*e.g.* Skein, BMW-512)
  - Use vector instructions (*e.g.* Blake, CubeHash, Hamsi, JH, Keccak, Luffa, SIMD)

- Vector instructions are more widely available than 64-bit integers or AES instructions.
  - SSE2 on x86, AltiVec on PowerPC, IwMMXt or NEON on ARM, ...

Speed vs Security

NIST wants SHA-3 to be faster and more secure than SHA-2.

More secure: more operations

Faster: less time

We need to cheat (use the hardware more efficiently)

Use AES instructions	(*e.g.* ECHO, SHAvite-3)
Use 64-bit integers	(*e.g.* Skein, BMW-512)
Use vector instructions	(*e.g.* Blake, CubeHash, Hamsi, JH, Keccak, Luffa, SIMD)

Vector instructions are more widely available than 64-bit integers or AES instructions.

SSE2 on x86, AltiVec on PowerPC, IwMMXt or NEON on ARM, ...

Speed vs Security

NIST wants SHA-3 to be faster and more secure than SHA-2.

More secure: more operations

Faster: less time

We need to cheat (use the hardware more efficiently)

Use AES instructions	(*e.g.* ECHO, SHAvite-3)
Use 64-bit integers	(*e.g.* Skein, BMW-512)
Use vector instructions	(*e.g.* Blake, CubeHash, Hamsi, JH, Keccak, Luffa, SIMD)

Vector instructions are more widely available than 64-bit integers or AES instructions.

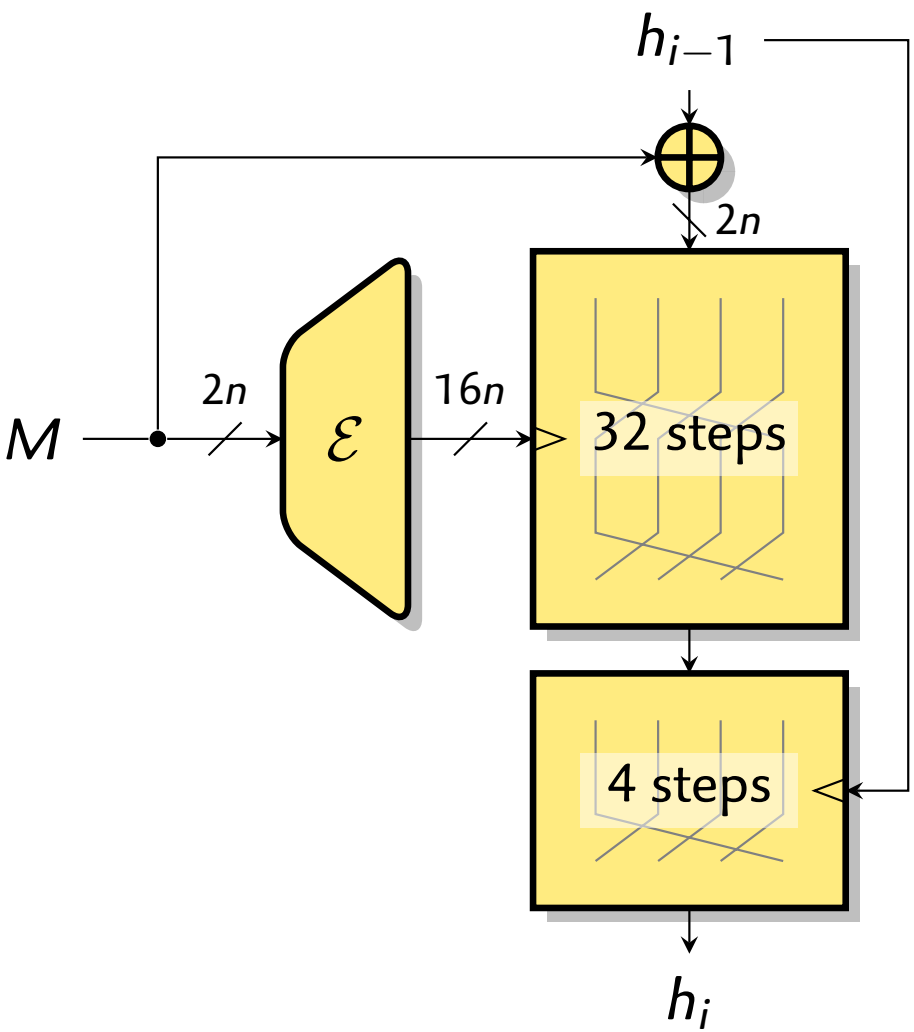
SSE2 on x86, AltiVec on PowerPC, IwMMXt or NEON on ARM, ...

Design Choices

SIMD is designed to be:

- Vectorisable
- With a Strong Message Expansion
- Wide-pipe

SIMD Compression Function



Block cipher based

- Well understood

Davies-Meyer

- Allows a strong message expansion

Add the message at the start

- Prevents some message modifications

Modified feed-forward: Feistel rounds instead of XOR

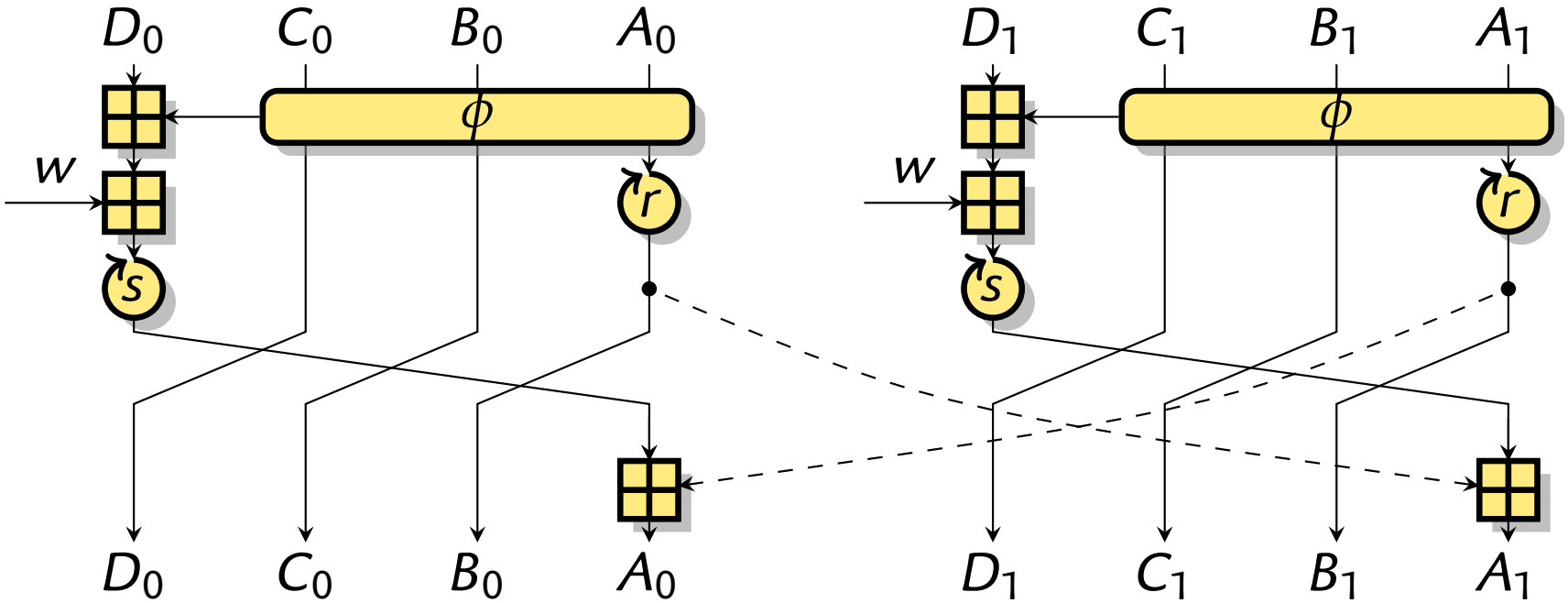
- Avoids some fixed point and multi-block attacks

Design Choices

SIMD is designed to be:

- Vectorisable
- With a Strong Message Expansion
- Wide-pipe

SIMD Feistel Rounds



Follows the SHA/MD legacy

- Additions, rotations, boolean functions
- Well understood

4 Parallel lanes for SIMD-256, 8 for SIMD-512

Parallel Feistel rounds allow vectorized implementation

Design Choices

SIMD is designed to be:

- Vectorisable
- With a Strong Message Expansion
- Wide-pipe

Strong Message Expansion

The inputs of a compression function have different roles:

- The message is controlled by the adversary
- The chaining value is only known

Use a strong transformation on the message.

- Trade-off: spend more time were it matters.

In Davies-Meyer mode, we have a message expansion.

In SIMD, code with big minimal distance:

	Msg. block	Expanded msg.	Min. distance
SIMD-256	512 bits	4096 bits	520 bits
SIMD-512	1024 bits	8192 bits	1032 bits

We can derive bounds for differential paths. [See paper]

Strong Message Expansion

The inputs of a compression function have different roles:

- The message is controlled by the adversary
- The chaining value is only known

Use a strong transformation on the message.

- Trade-off: spend more time were it matters.

In Davies-Meyer mode, we have a message expansion.

In SIMD, code with big minimal distance:

	Msg. block	Expanded msg.	Min. distance
SIMD-256	512 bits	4096 bits	520 bits
SIMD-512	1024 bits	8192 bits	1032 bits

We can derive bounds for differential paths. [See paper]

Design Choices

SIMD is designed to be:

- Vectorisable
- With a Strong Message Expansion
- Wide-pipe

Wide pipe

Avoid generic attacks on SHA-2

- MAC forgery in $2^{n/2}$
- Multicollisions
- Nostradamus attack (herding)
- Second-preimage for long messages
- Length-extension attack
- Various theoretical weaknesses

Good degradation of security:

Most distinguishers on the compression function
do not weaken the iterated function.

Several results show that indifferentiability proofs are quite resilient
In a wide-pipe design, indifferentiability implies all security notions.
[See paper, Shabal paper]

Analysis

Differential paths

Florian Mendel and Tomislav Nad
A Distinguisher for the Compression Function of SIMD-512

Hongbo Yu and Xiaoyun Wang
Cryptanalysis of the Compression Function of SIMD

Symmetry based distinguisher

Charles Bouillaguet, Pierre-Alain Fouque, and Gaëtan Leurent
Security analysis of SIMD

Partial fixed points

Praveen Gauravaram and Nasour Bagheri

Rotational cryptanalysis

Ivica Nikolić, Josef Pieprzyk, Przemyslaw Sokolowski, Ron Steinfeld
Rotational Cryptanalysis of (Modified) Versions of BMW and SIMD

Analysis

Differential paths

Florian Mendel and Tomislav Nad
A Distinguisher for the Compression Function of SIMD-512

Hongbo Yu and Xiaoyun Wang
Cryptanalysis of the Compression Function of SIMD

Symmetry based distinguisher

Charles Bouillaguet, Pierre-Alain Fouque, and Gaëtan Leurent
Security analysis of SIMD

Partial fixed points

Praveen Gauravaram and Nasour Bagheri

Rotational cryptanalysis

Ivica Nikolić, Josef Pieprzyk, Przemyslaw Sokolowski, Ron Steinfeld
Rotational Cryptanalysis of (Modified) Versions of BMW and SIMD

Differential paths (SIMD-512)

1st round Differential path with $p = 2^{-507}$ [Mendel and Nad]
Semi-free-start distinguisher with complexity 2^{427}

2nd round Differential path with $p = 2^{-897}$ [Yu and Wang]
Free-start distinguisher with complexity 2^{398}
Several mistakes in the ePrint path!

Attacks using a fixed message and a difference in the IV

Avoids the message expansion

Does not affect the security of the hash function [See paper]

(Semi-)Free-start attacks on wide-pipe design

Does not contradict the design goals

Security guaranteed from the message expansion

Analysis

Differential paths

Florian Mendel and Tomislav Nad
A Distinguisher for the Compression Function of SIMD-512

Hongbo Yu and Xiaoyun Wang
Cryptanalysis of the Compression Function of SIMD

Symmetry based distinguisher

Charles Bouillaguet, Pierre-Alain Fouque, and Gaëtan Leurent
Security analysis of SIMD

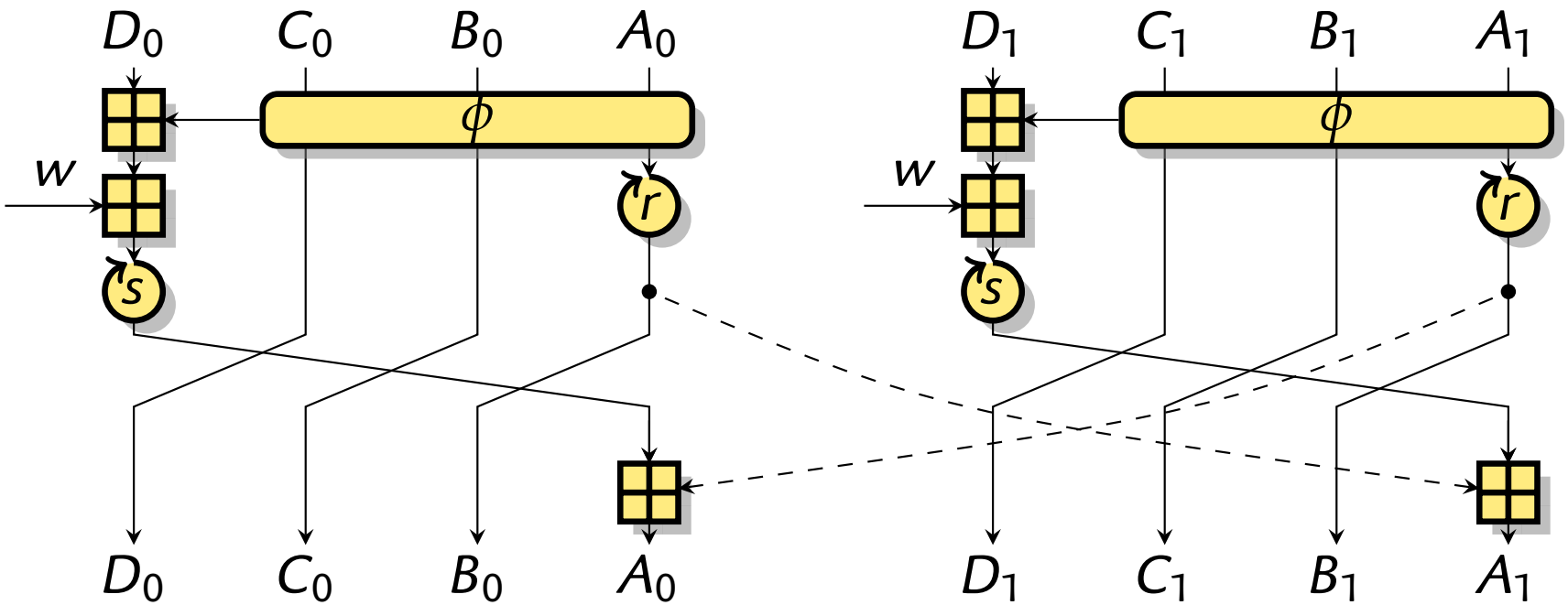
Partial fixed points

Praveen Gauravaram and Nasour Bagheri

Rotational cryptanalysis

Ivica Nikolić, Josef Pieprzyk, Przemyslaw Sokolowski, Ron Steinfeld
Rotational Cryptanalysis of (Modified) Versions of BMW and SIMD

Symmetry based distinguisher

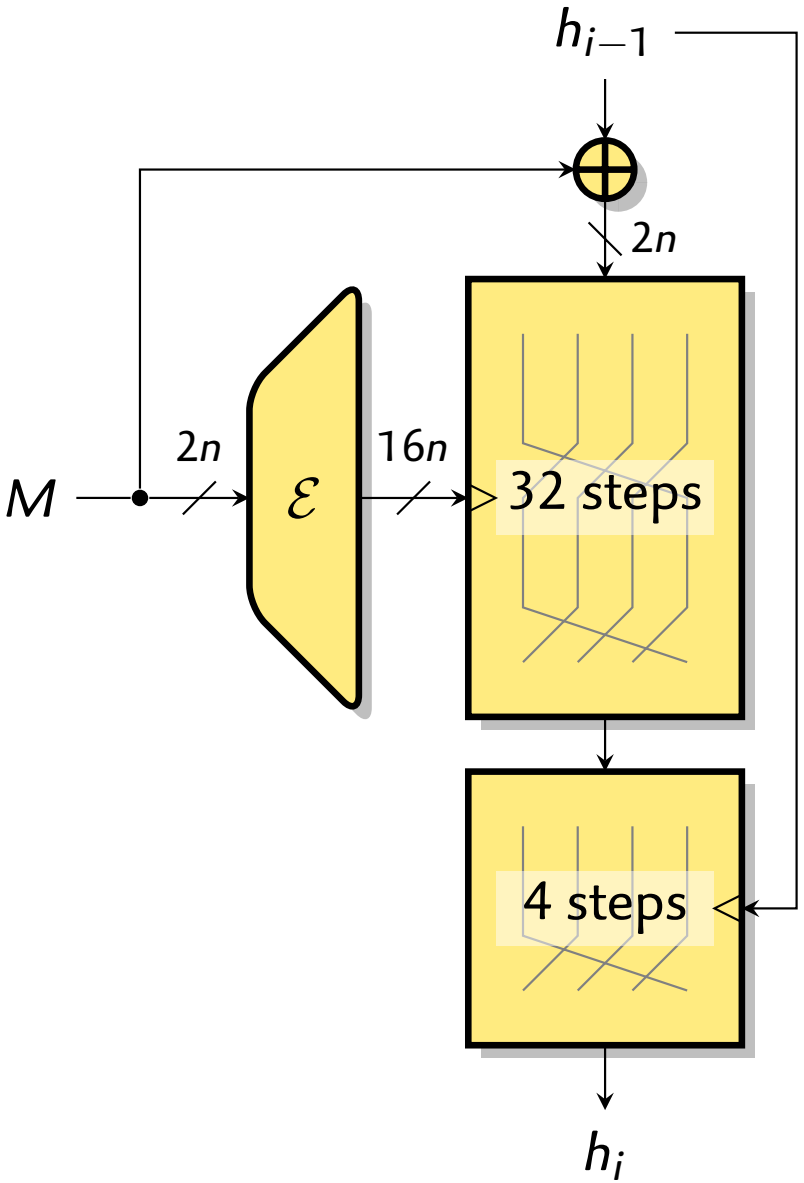


Put the same values in two lanes

Put the same message

Need a special message...

Application to the Compression Function



There are a few message giving a symmetric expanded message

Symmetric expanded message

Symmetric state in the Feistel

Message not symmetric

Almost symmetric input

Somewhat symmetric output

Tweak: add constants in the message expansion

Important properties

2^{256+16} weak states (2^{512+32} for SIMD-512)

Each state can be paired with 2^{32} states (2^{64} for SIMD-512)

Wide-pipe: It is hard to get into a symmetric state / pair of states

There is no intersection between the symmetry classes

Each pair only works with a single message pair

An output pair can not be used as input pair

It cannot be used in the final transform

Getting into a symmetric state is not really useful...

Proof of Security

Two results:

- Some distinguisher do not affect wide-pipe designs
  - Includes the symmetry-based distinguisher
  - Includes differential paths with a difference in the chaining value
  - Basic idea: reaching a weak state is too expensive
  - Shabal paper: results for a wider class of distinguisher
  - Indifferentiability results are quite resilient
- Differential path with a difference in the message are unlikely
  - Bounds using a modeling as an Integer Linear Program
    - *SIMD-256* $p \leq 2^{-132}$
    - *SIMD-512* $p \leq 2^{-253}$
  - Better results with heuristic assumptions

Main features of SIMD

Security

- Strong message expansion
- Security proof against differential attacks
- Some improvements of the standard
- Merkle-Damgård / Davies-Meyer

Security margin

- Few attacks so far
- The message expansion is doing its job well
- Guesstimate: around half the number of rounds

Performance

- Parallelism / Vectorisable
- Efficient use of modern processors
- Can also use two cores

Performance

OS	Processor *machine*	Core i5 *hi5*	Core 2 (45 nm) *jos*	Core 2 (65 nm) *katana*
64-bit	SIMD-256	7.78 cpb	9.70 cpb	11.34 cpb
	SIMD-512	8.85 cpb	10.48 cpb	12.05 cpb
32-bit	SIMD-256	8.75 cpb	10.03 cpb	11.98 cpb
	SIMD-512	10.43 cpb	10.99 cpb	13.77 cpb

[Taken from eBASH]

Performance

Vectorized implementations for SSE2, Altivec, and IwMMXt

Gives an idea of performances for a generic CPU with SIMD unit

Processor	Core 2	Atom	PowerPC G4	ARM Xscale
SHA-1	1	1	1	1
SHA-256	0.55	0.55	0.55	0.60
SHA-512	0.70	0.20	0.15	0.15
SIMD256	0.85	0.95	0.75	0.45
SIMD512	0.75	0.75	0.55	

Normalized speed

Vector units are available on all desktop/laptop/netbook and becoming available on embedded machines

They will get more powerful:

AVX on Intel (Q4 2010), AVX+XOP on AMD (2011)