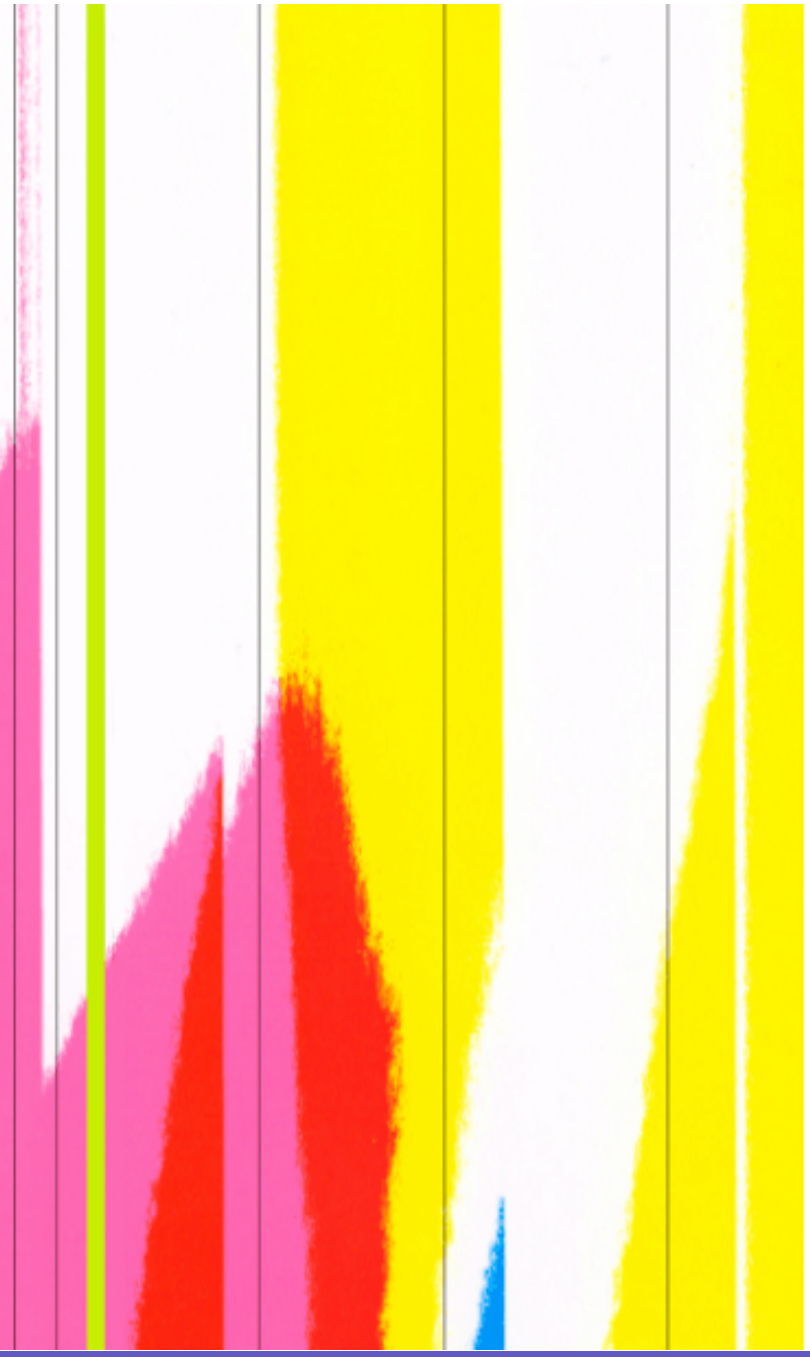


Collisions on SHA-0 in one hour

Stéphane Manuel
Thomas Peyrin

INRIA Rocquencourt, Team SECRET
Orange Labs - AIST

FSE
February 10-13, 2008
Lausanne

Outline

Outline

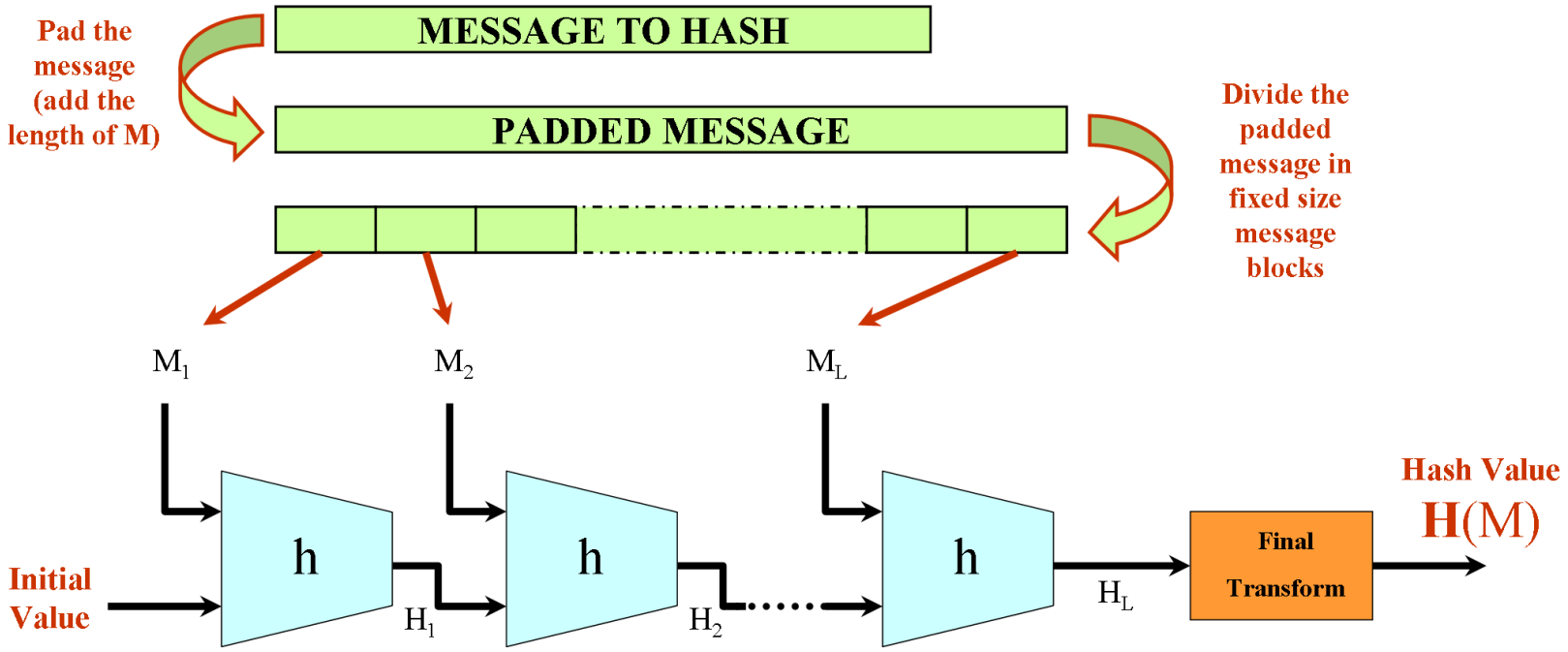
Cryptographic hash function

- An algorithm that maps input strings of arbitrary length to "short" fixed length output strings.

- Expected security properties:
  - Preimage resistance: given any specified output, it is computationally infeasible to find any input which hashes to this output.
  - Second preimage resistance: given any specified input, it is computationally infeasible to find another input which hashes to the same output.
  - Collision resistance: it is computationally infeasible to find two distinct input which hashes to the same output.

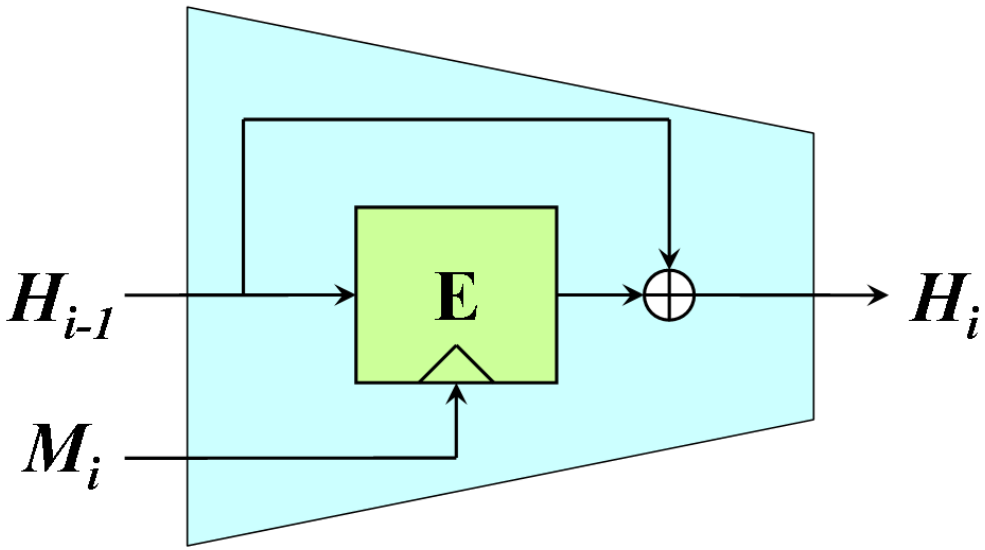
Domain extender

- The Merkle-Damgård algorithm:



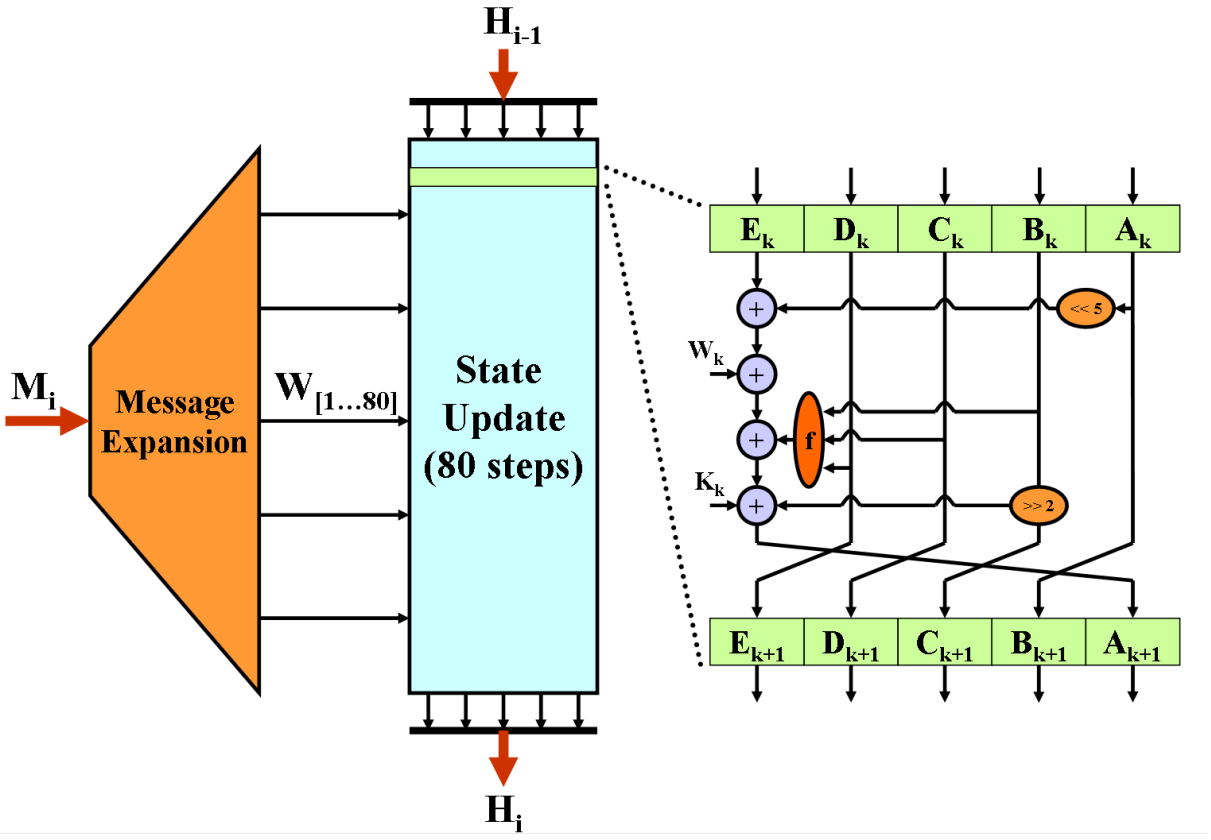
Compression function

- The Davies-Meyer construction:



The SHA-0 hash function

- Built in 1993, 160 bits output.



The SHA-0 hash function

- Message expansion:

$$W_k = \begin{cases} M_k, & \text{for } 0 \leq k \leq 15 \\ W_{k-16} \oplus W_{k-14} \oplus W_{k-8} \oplus W_{k-3}, & \text{for } 16 \leq k \leq 79 \end{cases}$$

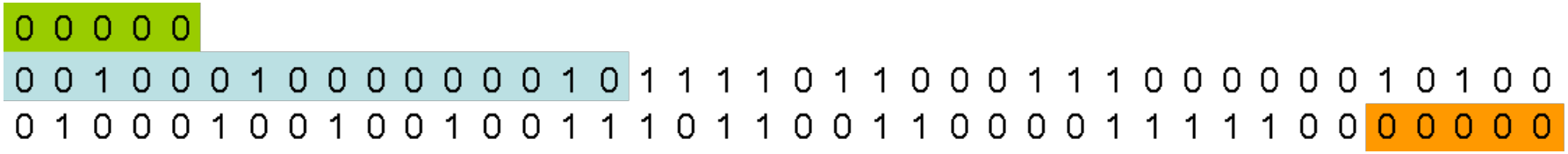
- Boolean functions:

step k	$f_k(B, C, D)$
$1 \leq k \leq 20$	$f_{IF} = (B \wedge C) \oplus (\overline{B} \wedge D)$
$21 \leq k \leq 40$	$f_{XOR} = B \oplus C \oplus D$
$41 \leq k \leq 60$	$f_{MAJ} = (B \wedge C) \oplus (B \wedge D) \oplus (C \wedge D)$
$61 \leq k \leq 80$	$f_{XOR} = B \oplus C \oplus D$

Outline

Chabaud and Joux [CRYPTO 98]

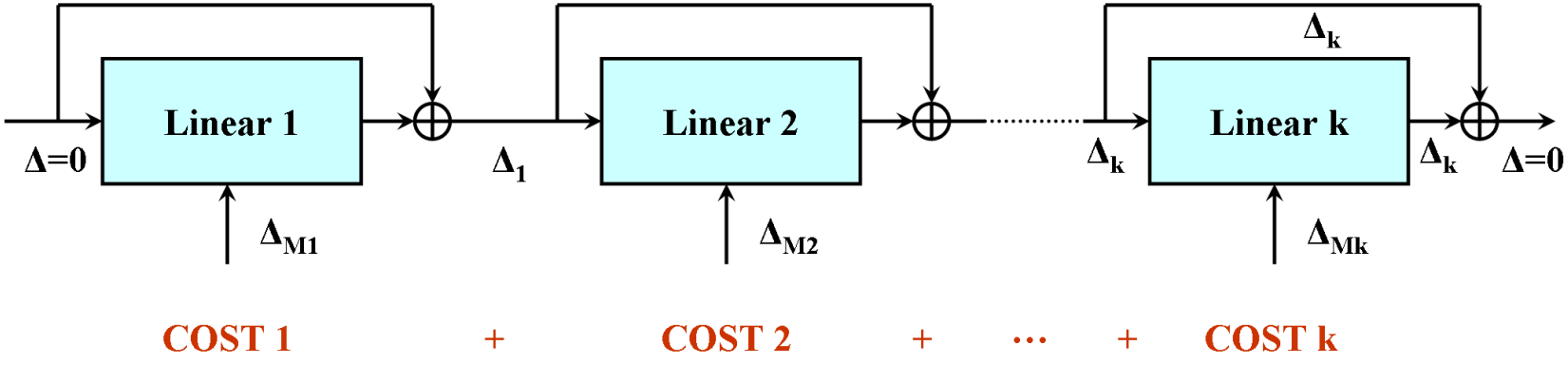
- Local collisions: insert a perturbation and correct it in the next 5 steps.
- Find linear differential path of interleaved local collisions with 3 constraints on the perturbation vector:
  - no truncated local collisions,
  - no consecutive perturbations in the first 16 steps,
  - no perturbation starting after step 74.



- Complexity is evaluated in terms of probability for local collisions to hold.

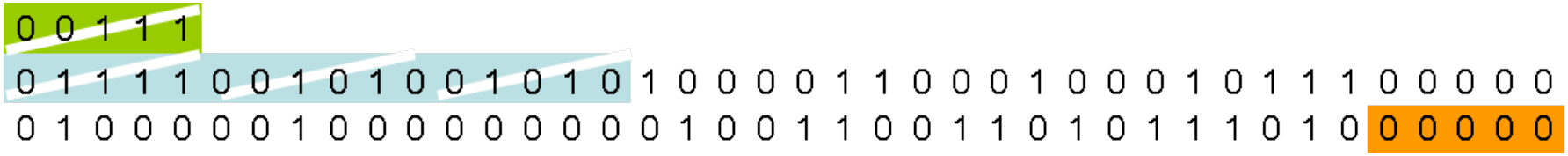
Biham *et al.*

- Biham and Chen [CRYPTO 04]
  - Speedup technique during collision search: using neutral bits, the conformance to the differential path is assured up to step 23.
- Biham *et al.* [EUROCRYPT 2005]
  - Multi-block technique: use several blocks to find a collision.



Wang *et al.* [CRYPTO 05]

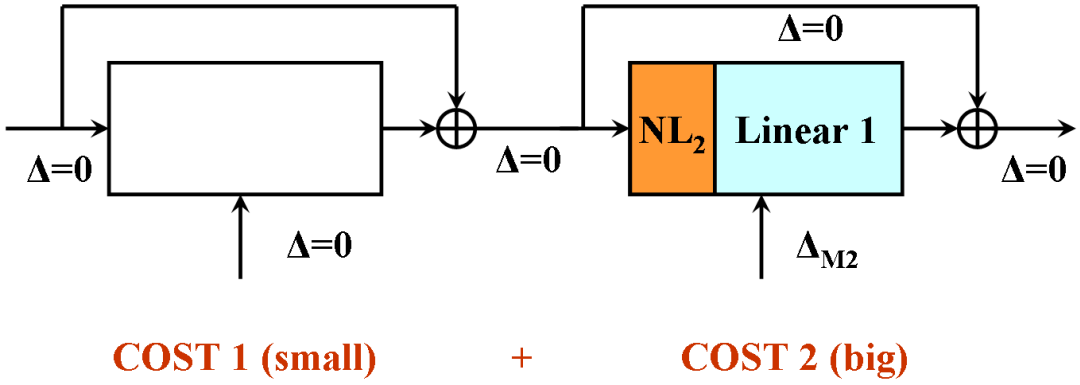
- Relax the first two constraints on the perturbation vector to find a better one.



- Modify (by hands) the first steps of the differential path to compensate truncated and consecutive local collisions, using different tools:
  - modular subtraction,
  - carry effect,
  - non-linearity of the boolean function f_{IF}.

Wang *et al.* [CRYPTO 05]

- Build from a random first block of message a chaining variable verifying specific conditions.



- Message modifications: another speedup technique.
  - Complexity is given in terms of number of conditions to fulfill (starts from step 20).

Naito *et al.* [ASIACRYPT 06]

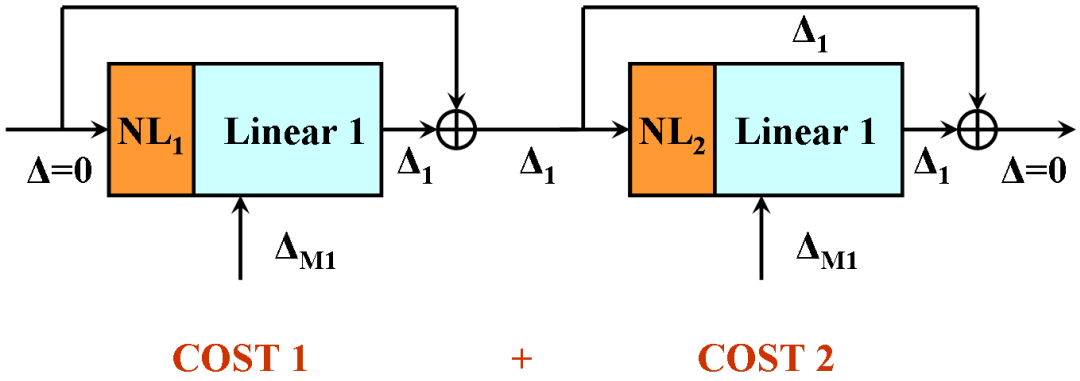
- Based on the linear and non-linear characteristics of Wang *et al.*
- Submarine modifications: condition counting starts from step 24.
- Complexity:
  - 2^{36} function calls theoretically ...
  - ... but requires 100 hours on average with a good PC.
  - Our estimation: $2^{40,5}$ function calls practically.

Complexity should be given in terms of function calls with an efficient implementation on the same computer (*i.e.* OpenSSL) according to De Cannière *et al.* proposal [Hash Workshop 2007].

Outline

Possible improvements

- Relax the last constraint to find better perturbation vectors:
  - no perturbation starting after step 74.
- Then we need:
  - multi-block technique,
  - adapted non-linear characteristics,
  - generic speedup technique.



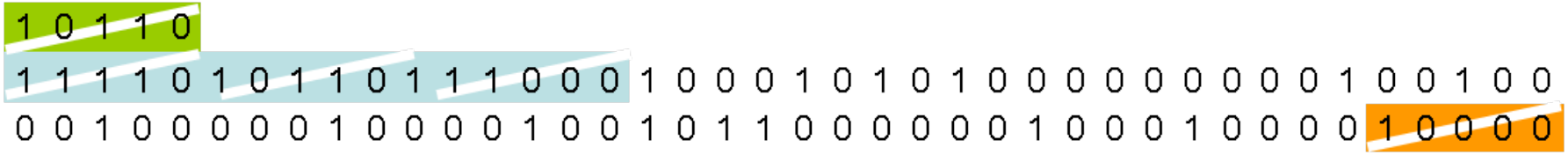
Possible improvements

Adapt the tools developed for the recent attacks against SHA-1.

- Non-linear characteristics:
  - the automated non-linear characteristic generator from De Cannière and Rechberger (2006).
- Speedup technique:
  - the boomerang attacks from Joux and Peyrin (2007).

New perturbation vector

- Criteria for vector search:
  - minimize the number of conditions between steps 16 and 80,
  - starting step for counting conditions depends on the speedup technique,
  - adaptability with the non-linear characteristic generator.
- Several good possible vectors found.
  - Our perturbation vector:



The boomerangs

- Boomerangs are a framework:
  - The attacker build auxiliary differentials that can be used under neutral bits or message modifications settings.
  - With the neutral bits setting they give a generic easy to use tool for collision search speedup.
  - Constraints are set to provide good neutral bits that would exist with very low probability on a random differential path.
- Our approach:
  - First find good generic auxiliary differentials.
  - Place them so that they do not interfere with the perturbation vector.
  - Then run the non-linear characteristic generator taking in account these auxiliary differentials.

The boomerangs

- We build two types of auxiliary differentials:
  - a light but short one (few constraints but low range),
  - and a heavy but long one (long range but lot of constraints).
- These auxiliary differentials are used as neutral bits for steps 23 and 28 respectively.
- On average, we can set 5 auxiliary differentials (7 for the first block):
  - improvement of a factor 2^5 on the raw attack.

First auxiliary differential

i	A_i	W_i
-1:	----------	
00:	----------	----------
01:	----------	----------
02:	----------	----------
03:	----------	----------
04:	----------	----------
05:	----------b--	----------
06:	----------b--	----------a-
07:	----------a-	----------$\bar{a}$----
08:	----------0	----------
09:	----------1	----------
10:	----------	----------
11:	----------	----------$\bar{a}$
12:	----------	----------
13:	----------	----------
14:	----------	----------
15:	----------	----------

Second auxiliary differential

i	A_i	W_i
-1:	----------d--	
00:	----------d--	----------a-
01:	----------e-a-	----------$\bar{a}$----
02:	----------e--1	----------b-
03:	----------b-0	----------$\bar{b}$---$\bar{a}$
04:	----------0	----------$\bar{a}$
05:	----------0	----------$\bar{a}$
06:	----------	----------$\bar{b}$
07:	----------	----------$\bar{b}$
08:	----------	----------
09:	----------f--	----------
10:	----------f--	----------c-
11:	----------c-	----------$\bar{c}$----
12:	----------0	----------
13:	----------0	----------
14:	----------	----------$\bar{c}$
15:	----------	----------$\bar{c}$

Collision example

	1^{st} block		2^{nd} block	
	M_1	M'_1	M_2	M'_2
W_0	0x4643450b	0x46434549	0x9a74cf70	0x9a74cf32
W_1	0x41d35081	0x41d350c1	0x04f9957d	0x04f9953d
W_2	0xfe16dd9b	0xfe16dddb	0xee26223d	0xee26227d
W_3	0x3ba36244	0x3ba36204	0x9a06e4b5	0x9a06e4f5
W_4	0xe6424055	0x66424017	0xb8408af6	0x38408ab4
W_5	0x16ca44a0	0x96ca44a0	0xb8608612	0x38608612
W_6	0x20f62444	0xa0f62404	0x8b7e0fea	0x0b7e0faa
W_7	0x10f7465a	0x10f7465a	0xe17e363c	0xe17e363c
W_8	0x5a711887	0x5a7118c5	0xa2f1b8e5	0xa2f1b8a7
W_9	0x51479678	0xd147963a	0xca079936	0x4a079974
W_{10}	0x726a0718	0x726a0718	0x02f2a7cb	0x02f2a7cb
W_{11}	0x703f5bfb	0x703f5bb9	0xf724e838	0xf724e87a
W_{12}	0xb7d61841	0xb7d61801	0x37ffc03a	0x37ffc07a
W_{13}	0xa5280003	0xa5280041	0x53aa8c43	0x53aa8c01
W_{14}	0x6b08d26e	0x6b08d26c	0x90811819	0x9081181b
W_{15}	0x2e4df0d8	0xae4df0d8	0x312d423e	0xb12d423e

A_2	B_2	C_2	D_2	E_2
0x6f84b892	0x1f9f2aae	0x0dbab75c	0x0afe56f5	0xa7974c90

Outline

Complexity comparison

Team	Theoritical	Practical	Time on a PC
Chabaud and Joux (1998)	2^{61}		
Biham *et al.* (2004)	2^{51}	2^{51}	20 years
Wang *et al.* (2005)	2^{39}		
Naito *et al.* (2006)	2^{36}	$2^{40.3}$	100 hours
Our results	2^{33}	$2^{33.6}$	1 hour

Complexity comparison

Thank you!