A heuristic for finding compatible differential paths with application to HAS-160

Aleksandar Kircanski, Riham AlTawy, Amr M. Youssef

Concordia University
Concordia Institute for Information Systems Engineering
Montréal, Québec, Canada

ASIACRYPT 2013

Outline

- HAS-160 specification
- de Canniére and Rechberger (2006) differential path search
- Second-order collisions
- Searching for compatible/non-conflicting paths
  - Heuristic workflow
  - Propagation types
    - Single-path propagation
    - Quartet propagations
    - Quartet carry propagations
- Conclusion and future work

Some of the previous work on HAS-160

HAS-160: KISA (Korea Information Security Agency) + Academia, "Hash Function Standard (HAS-160),", TTA.IS-10118, 1998.

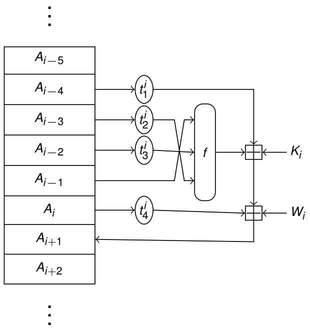
SHA-based hash, Merkle-Damgård construction, Davies-Meyer mode

- ICISC 2005, Yun *et al.*: Practical 45-step collision
- ICISC 2006, Cho *et al.*: 53-step collision in 2^{55}
- ICISC 2007, Mendel and Rijmen: Practical 65-step two-block collision
- ICISC 2011, Mendel *et al.*: Practical semi-freestart collision on 65 steps
- ICISC 2012, Sasaki *et al.*: Practical boomerang distinguisher for 75-step reduced compression function
  - Boomerang distinguisher for full HAS-160 with $2^{76.06}$

Our work

Is it possible to build a practical full 80-step distinguisher?

The HAS-160 hash function step update



Compression function (represented as a shift register):

$$\begin{aligned} A_{i+1} \quad &= A_{i-4} \lll t_1^i + K_i + f_i(A_{i-1}, A_{i-2} \lll t_3^i, A_{i-3} \lll t_2^i) \\ &\quad + W_i + A_i \lll t_4^i, \quad \text{where } i = 0, \ldots 79 \end{aligned}$$

Design very similar to SHA-1, except that the rotation constants change in every step.

Message expansion in HAS-160

i	Steps 1-20	Steps 21-40	Steps 41-60	Steps 61-80
0	$m_8 \oplus m_9$ $\oplus m_{10} \oplus m_{11}$	$m_{11} \oplus m_{14}$ $\oplus m_1 \oplus m_4$	$m_4 \oplus m_{13}$ $\oplus m_6 \oplus m_{15}$	$m_{15} \oplus m_{10}$ $\oplus m_5 \oplus m_0$
1	m_0	m_3	m_{12}	m_7
2	m_1	m_6	m_5	m_2
3	m_2	m_9	m_{14}	m_{13}
4	m_3	m_{12}	m_7	m_8
5	$m_{12} \oplus m_{13}$ $\oplus m_{14} \oplus m_{15}$	$m_7 \oplus m_{10}$ $\oplus m_{13} \oplus m_0$	$m_8 \oplus m_1$ $\oplus m_{10} \oplus m_3$	$m_{11} \oplus m_6$ $\oplus m_1 \oplus m_{12}$
6	m_4	m_{15}	m_0	m_3
7	m_5	m_2	m_9	m_{14}
8	m_6	m_5	m_2	m_9
9	m_7	m_8	m_{11}	m_4
10	$m_0 \oplus m_1$ $\oplus m_2 \oplus m_3$	$m_3 \oplus m_6$ $\oplus m_9 \oplus m_{12}$	$m_{12} \oplus m_5$ $\oplus m_{14} \oplus m_7$	$m_7 \oplus m_2$ $\oplus m_{13} \oplus m_8$
11	m_8	m_{11}	m_4	m_{15}
12	m_9	m_{14}	m_{13}	m_{10}
13	m_{10}	m_1	m_6	m_5
14	m_{11}	m_4	m_{15}	m_0
15	$m_4 \oplus m_5$ $\oplus m_6 \oplus m_7$	$m_{15} \oplus m_2$ $\oplus m_5 \oplus m_8$	$m_0 \oplus m_9$ $\oplus m_{12} \oplus m_{11}$	$m_3 \oplus m_{14}$ $\oplus m_9 \oplus m_4$
16	m_{12}	m_7	m_8	m_{11}
17	m_{13}	m_{10}	m_1	m_6
18	m_{14}	m_{13}	m_{10}	m_1
19	m_{15}	m_0	m_3	m_{12}

de Canniére and Rechberger heuristic (2006)

- Applied on SHA-1, SHA-2, SM3, RIPEMD-160,...
- Switch from *bit-values* to *bit-constraints*
- Bit-constraints: a symbol for each bit pair configuration (b, b')
  - `'?'` if there is no constraint on (b, b')
  - `'x'` if $b \neq b'$
  - `'-'` if $b = b'$
  - `'u'` if $b = 0$ and $b' = 1$
  - `'n'` if $b = 1$ and $b' = 0$
  - ...

Workflow:

- *Guess*: select a `?` or `x` and replace by `-` or {`u`,`n`}, respectively.
- *Propagate*: propagate all new knowledge.

Boomerang distinguishers for hash functions

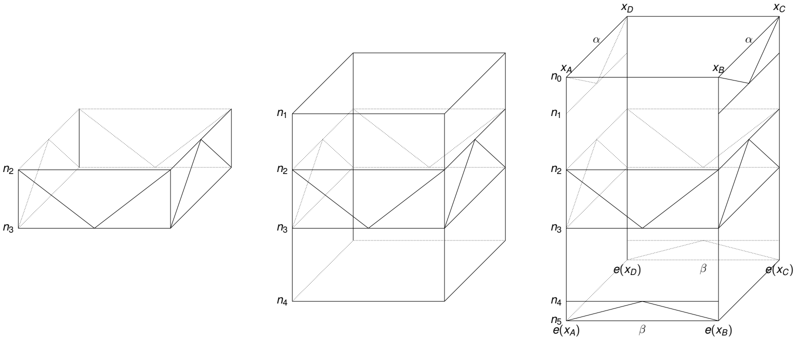
Definition

A *second order collision* for h is a set $\{x, \Delta, \nabla\}$ consisting of an input for h and two differences, such that

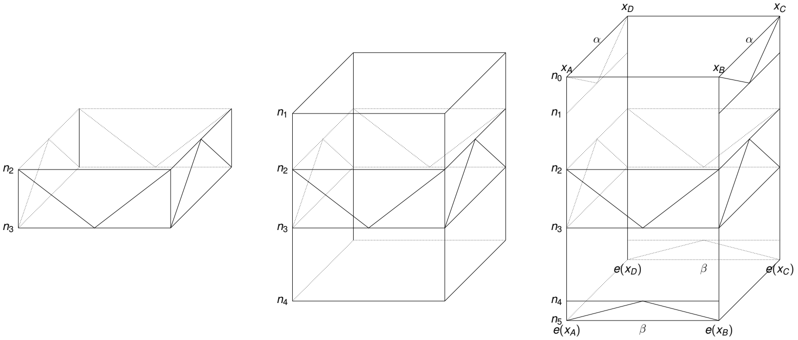
$$h(x + \Delta + \nabla) - h(x + \Delta) - h(x + \nabla) + h(x) = 0$$

Boomerang attack for the purpose of second order collisions:

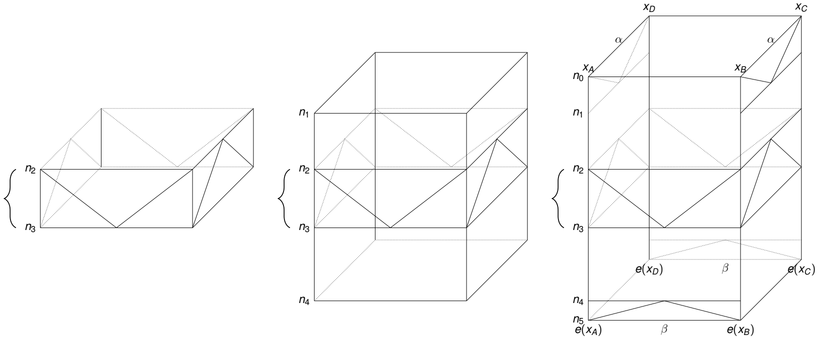
- Biryukov *et al.* in the context of BLAKE (2011)
- Lamberger and Mendel in the context of SHA-256 (2011)



- Due to Davies-Meyer, the goal is to have:
  - $d(x_A, x_D) = d(x_B, x_C) = \alpha$
  - $d(e(x_A), e(x_B)) = d(e(x_D), e(x_C)) = \beta$
- Step notation: $0 \leq n_0, n_1, n_2, n_3, n_4, n_5 \leq n$
  - n_0, n_5: attacked steps
  - n_1, n_2, n_3, n_4: activation/deactivation steps



- Start from the middle: construct the quartet for steps n_2, n_3
- Extend the quartet to steps n_1, n_4
- Extend the quartet for some more steps n_0, n_5
- Randomize the quartet restarting from the first stage, until
  - $d(x_A, x_D) = d(x_B, x_C)$
  - $d(e(x_A), e(x_B)) = d(e(x_D), e(x_C))$



- Suboptimal number of middle steps
  - e.g., less than 16 steps
- Our work: improve the number of steps in the middle
- In case of HAS-160: 20 steps in the middle

Our proposal

A heuristic based on the path search heuristic by de Canniére and Rechberger that finds Compatible / non-conflicting / independent paths

step	Δ[A, B]	Δ[D, C]	Δ[B, C]	Δ[A, D]	step
9	????????????????????????????????	????????????????????????????????	--------------------------------	--------------------------------	9
10	????????????????????????????????	????????????????????????????????	--------------------------------	--------------------------------	10
11	????????????????????????????????	????????????????????????????????	--------------------------------	--------------------------------	11
12	????????????????????????????????	????????????????????????????????	--------------------------------	--------------------------------	12
13	????????????????????????????????	????????????????????????????????	--------------------------------	--------------------------------	13
⋮	⋮	⋮	⋮ [NO DIFFERENCE]	⋮	⋮
29	????????????????????????????????	????????????????????????????????	--------------------------------	--------------------------------	29
30	????????????????????????????????	????????????????????????????????	--------------------------------	--------------------------------	30
31	????????????????????????????????	????????????????????????????????	--------------------------------	--------------------------------	31
32	????????????????????????????????	????????????????????????????????	--------------------------------	--------------------------------	32
33	????????????????????????????????	????????????????????????????????	--------------------------------	--------------------------------	33
34	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	34
35	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	35
36	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	36
37	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	37
38	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	38
39	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	39
40	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	40
41	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	41
42	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	42
43	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	43
44	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	44
45	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	45
46	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	46
47	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	47
48	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	48
49	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	49
50	--------------------------------	--------------------------------	????????????????????????????????	????????????????????????????????	50
51	--------------------------------	--------------------------------	????????????????????????????????	????????????????????????????????	51
52	--------------------------------	--------------------------------	????????????????????????????????	????????????????????????????????	52
53	--------------------------------	--------------------------------	????????????????????????????????	????????????????????????????????	53
54	--------------------------------	--------------------------------	????????????????????????????????	????????????????????????????????	54
⋮	⋮ [NO DIFFERENCE]	⋮	⋮	⋮	⋮
76	--------------------------------	--------------------------------	????????????????????????????????	????????????????????????????????	76
77	--------------------------------	--------------------------------	????????????????????????????????	????????????????????????????????	77

Search heuristic

- Pick a random bit position in the quartet
- If applicable: perform substitution

1.	???? $\mapsto$ --??
2.	??-- $\mapsto$ ----
3.	??xx $\mapsto$ --xx
4.	xx?? $\mapsto$ {uu10,nn01}
5.	xx-- $\mapsto$ {uu10,nn01}
6.	xxxx $\mapsto$ {unnu,nuun}

1.	???? $\mapsto$??--
2.	--?? $\mapsto$ ----
3.	xx?? $\mapsto$ xx--
4.	??xx $\mapsto$ {01uu,10nn}
5.	--xx $\mapsto$ {01uu, 10nn}
6.	xxxx $\mapsto$ {unnu,nuun}

- Apply the following three types of propagation:
  - Single path propagations
  - Quartet propagations
  - Quartet addition propagations
- In case of contradiction, backtrack

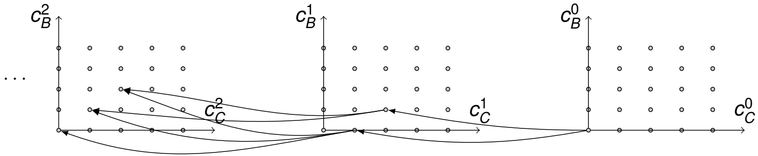
The substitution rules are a natural generalization of

- ? $\mapsto$ -,   x $\mapsto$ {u,n}

Single-path propagations

δK	01101110110110011110101110100001
$\delta[W_{B,41}, W_{C,41}]$	-----0---0----------------------
$\delta[B_{37}, C_{37}]$	11-0-u--00u-10n--10-0n1un---
$\delta[B_{38}, C_{38}]$	0u1000-100111uu011-0n11u0000-u1-
$\delta[B_{39}, C_{39}]$	01un-n0u010u-0-u00u-1--n0u-un00
$\delta[B_{40}, C_{40}]$	1-nu001uu01-0n-u01-1-0-u0-11-1
$\delta[B_{41}, C_{41}]$	1-n-00--0-u1-u-u1-001-0--1
$\delta[B_{42}, C_{42}]$	1--n-u1uun-n1u-00n0nn-0n0--n

δK	01101110110110011110101110100001
$\delta[W_{B,41}, W_{C,41}]$	-----0---0----------------------
$\delta[B_{37}, C_{37}]$	11-0-u--00u-10n--10-0n1un---
$\delta[B_{38}, C_{38}]$	0u1000-100111uu011-0n11u0000-u1-
$\delta[B_{39}, C_{39}]$	01un-n0u010u-0-u00u-1--n0u-un00
$\delta[B_{40}, C_{40}]$	1-nu001uu01-0n-u01-1-0-u0-11-1
$\delta[B_{41}, C_{41}]$	1-n-00-0-u1-u-u1-001-0--1
$\delta[B_{42}, C_{42}]$	1--n-u1uun-n1u-00n0nn-0n0--n



- Conditions: propagate bits that affect the LSB
- Carries: propagate new carry configurations
- Edges represent carry transitions
- Knowledge propagation can be mapped in edge removal
- Perform propagations at *all* affected bit-positions

Quartet propagations

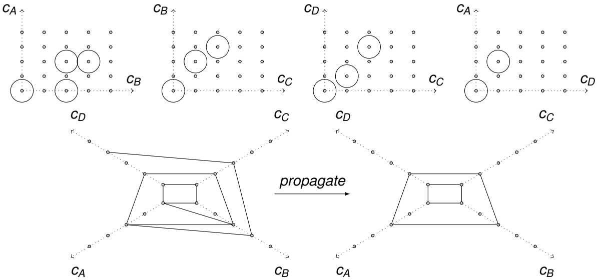
Simplest type of propagations.
Do not directly influence/depend on carry graphs.
Example:

- Let $\delta[A, B]$, $\delta[D, C]$, $\delta[B, C]$, and $\delta[A, D]$ at bit position (i, j) follow `(ux-?)`
- Then $A_i^j = 0$, $B_i^j = 1$, $C_i^j = 1$ and $D_i^j = 0$
- Propagate: `(ux-?)` $\mapsto$ `(uu10)`

Rationale:

- Four bit-constraints influence each bit-value twice
- Take the minimal constraint describing the possible configurations
- Can be placed in a pre-computed table

Quartet addition propagations

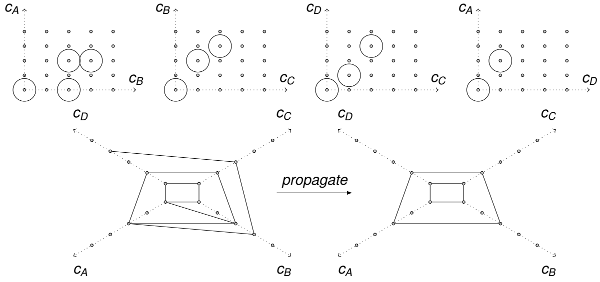


Introduce: 4-graphs or *quartet carry graphs*

Natural expression of quartet addition propagation rules

- Each bit-position: four "single-path" carry graphs
- Each execution branch: two "single-path" carry graphs
- Two "single-path" carry graphs: contradictory constraints?

Quartet addition propagations

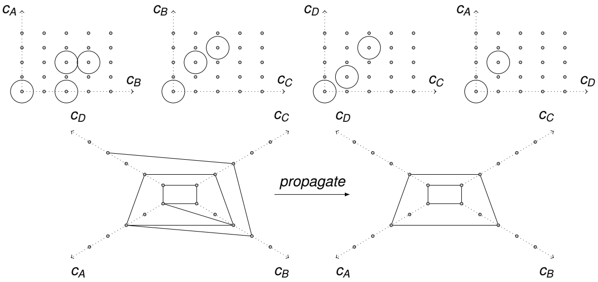


- Active carry graph nodes $\mapsto$ edges in quartet carry graphs
- A "dead-end" QCG edge:
  - One corresp. CG: a particular carry value *possible*
  - The other corresp. CG: a particular carry value *impossible*

Rule (1)

Remove all the "dead-end" edges recursively

Quartet addition propagations



- A “QCG edge not participating in any cycle”
  - Allows a certain carry configuration on the two branches
  - However, it cannot be realized
  - Cannot connect the carry configurations on the other end

Rule (2)

Remove all the edges not participating in “cycles”

Propagation rules related to QCG:

- (R1) Remove all the “dead-end” edges recursively
- (R2) Remove all the edges not participating in “cycles”

Now, “propagation” amounts to recursive application of:

- Single-path propagations
- Quartet propagations
- Quartet addition propagations

Implementing rule (R1): sufficient in case of HAS-160.

step	Δ[A, B]	Δ[D, C]	Δ[B, C]	Δ[A, D]	step
9	????????????????????????????????	????????????????????????????????	---------------------	---------------------	(9)
10	????????????????????????????????	????????????????????????????????	---------------------	---------------------	10
11	????????????????????????????????	????????????????????????????????	---------------------	---------------------	11
12	????????????????????????????????	????????????????????????????????	---------------------	---------------------	12
13	????????????????????????????????	????????????????????????????????	---------------------	---------------------	13
⋮	⋮	⋮	⋮ [NO DIFFERENCE]	⋮	⋮
29	????????????????????????????????	????????????????????????????????	---------------------	---------------------	29
30	????????????????????????????????	????????????????????????????????	---------------------	---------------------	30
31	????????????????????????????????	????????????????????????????????	---------------------	---------------------	31
32	????????????????????????????????	????????????????????????????????	---------------------	---------------------	32
33	????????????????????????????????	????????????????????????????????	---------------------	---------------------	(33)
34	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	(34)
35	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	35
36	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	(36)
37	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	37
38	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	38
39	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	(39)
[40]	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	(40)
[41]	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	41
42	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	42
43	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	43
44	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	(44)
45	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	(45)
46	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	(46)
47	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	47
48	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	48
49	????????????????????????????????	????????????????????????????????	????????????????????????????????	????????????????????????????????	49
[50]	---------------------	---------------------	????????????????????????????????	????????????????????????????????	(50)
51	---------------------	---------------------	????????????????????????????????	????????????????????????????????	(51)
52	---------------------	---------------------	????????????????????????????????	????????????????????????????????	52
[53]	---------------------	---------------------	????????????????????????????????	????????????????????????????????	53
54	---------------------	---------------------	????????????????????????????????	????????????????????????????????	54
⋮	⋮ [NO DIFFERENCE]	⋮	⋮	⋮	⋮
76	---------------------	---------------------	????????????????????????????????	????????????????????????????????	76
[77]	---------------------	---------------------	????????????????????????????????	????????????????????????????????	77

step	Δ[A, B]	Δ[D, C]	Δ[B, C]	Δ[A, D]	step
29	????????????????????????????????	????????????????????????????????	---------------------	---------------------	29
30	????????????????????????????????	????????????????????????????????	---------------------	---------------------	30
31	????????????????????????????????	????????????????????????????????	---------------------	---------------------	31
32	????????????????????????????????	????????????????????????????????	---------------------	---------------------	32
33	????????????????????????????????	????????????????????????????????	---------------------	---------------------	(33)
34	0???????????????????????????????	1???????????????????????????????	u---------------------	u---------------------	(34)
35	0??????????u???????x0???x-0?????	1??????????u???????x0???x-1?????	u-------1-----0---u---	u-------0-----0---u---	35
36	1x????????xu?-01B?-0Bx-u0D????	0x????????xu?-11B?-1Bx-u0D????	n-------1-u1---u---10---	n-------0-u1---u---00---	(36)
37	11-0D0B??0n0?101-x-10-01u01C???x	11-0D1B??0n1?100-x-10-00u10C???x	11-0-u--00u-10n--10-0n1un---	11-0-u--01u-10n--10-0n0un1---	37
38	00u0nn-1n01uu000uu-011u00nnn-01-	01u0nn-1n01uu110uu-001u10nnn-11-	0u1000-100111uu011-0n11u0000-u1-	0u0011-110100uu000-0n10u0111-u1-	38
39	n101-1000100-0-0000-1--100-010n	n110-0010101-0-1001-1--001-100n	01un-n0u010u-0-u00u-1--n0u-un00	11un-n0u010u-0-u00u-1--n0u-un01	(39)
[40]	1-100010001-01-0n1-u-0-00-11-1	1-010011101-00-1n1-u-0-10-11-1	1-nu001uu01-0n-u01-1-0-u0-11-1	1-nu001uu01-0n-u11-0-0-u0-11-1	(40)
[41]	u-1-00-0-01-0-0u-001-0--1	u-0-00-0-11-1-1u-001-0--1	1-n-00-0-u1-u-u1-001-0--1	0-n-00-0-u1-u-u0-001-0--1	41
42	u--1-01001-110-n01011-n10--1	u--0-11110-011-n00000-n00--0	1--n-u1uun-n1u-00n0nn-0n0--n	0--n-u1uun-n1u-10n0nn-1n0--n	42
43	n----01---0----u----00-un	n----00---0----u----01-un	0????-0nD???0x?????1x??x-0u-10	1????-0nD???0x?????0x??x-0u-01	43
44	0---10-----------1u----	0----0-----------1u----	0?????C0????????????????11?????x	0?????C0????????????????10?????x	(44)
45	----00--------u----1---	----00--------u----1---	??????00????????????1??????1????	??????00????????????0??????1????	(45)
46	u--------------------	u--------------------	1???????????????????????????????	0???????????????????????????????	(46)
47	---------------------	---------------------	????????????????????????????????	????????????????????????????????	47
48	-----u----------------	-----u----------------	???????1????????????????????????	???????0????????????????????????	48
49	-----n----------------	-----n----------------	???????0????????????????????????	???????1????????????????????????	49
[50]	---------------------	---------------------	????????????????????????????????	????????????????????????????????	(50)
51	---------------------	---------------------	????????????????????????????????	????????????????????????????????	(51)
52	---------------------	---------------------	????????????????????????????????	????????????????????????????????	52
[53]	---------------------	---------------------	????????????????????????????????	????????????????????????????????	53
54	---------------------	---------------------	????????????????????????????????	????????????????????????????????	54

Second order collision for the full HAS-160 compression function

Message quartet								
M_A	F6513317	810F1084	FFB71009	78CC955E	C3C09F18	5379FC99	435586DA	9C9AD3B4
	00440C80	E174316A	006D1670	2B5CF68A	AB3DE600	02C9E9D3	5FE95AFF	E351DE04
M_B	F6513317	810F1084	FFB71009	78CC955E	C3C09f18	5379FC99	435786DA	9C9AD3B4
	00440C80	E174316A	006D1670	2B5CF68A	AB3FE600	02C9E9D3	5FE95AFF	E351DE04
M_C	76513317	010F1084	FFB71009	78CC955E	43C09F18	5379FC99	435786DA	1C9AD3B4
	00440C80	E174316A	006D1670	2B5CF68A	AB3FE600	02C9E9D3	5FE95AFF	E351DE04
M_D	76513317	010F1084	FFB71009	78CC955E	43C09f18	5379FC99	435586DA	1C9AD3B4
	00440C80	E174316A	006D1670	2B5CF68A	AB3DE600	02C9E9D3	5FE95AFF	E351DE04
Chaining values quartet								
IV_A	1143BE75	9A9CA381	85B3F526	DA6ABE66	70EBE920			
IV_B	3AF7BD99	D08E2E63	245C2AF0	C4456954	CAC046EA			
IV_C	3AF7B599	D08E2E63	B45C2AF0	C425694C	3BE146F2			
IV_D	1143B675	9A9CA381	15B3F526	DA4ABE5E	E20CE928			

Conclusion

- A heuristic for searching for compatible/non-conflicting diff. paths was proposed
- A generalization of the previous path search heuristic
- HAS-160: Second-order collision for the full 80-step function.
- How do 1-bit constraints and three proposed propagation types work with more complex functions (SHA-2, SM3, ..)?

Thank you!