

MegoSAT project

Differential cryptanalysis with SAT solvers

Today: “master thesis presentation”



Lukas Prokop
Advisors: Florian Mendel, Maria Eichlseder
Institute of Applied Information Processing and Communications

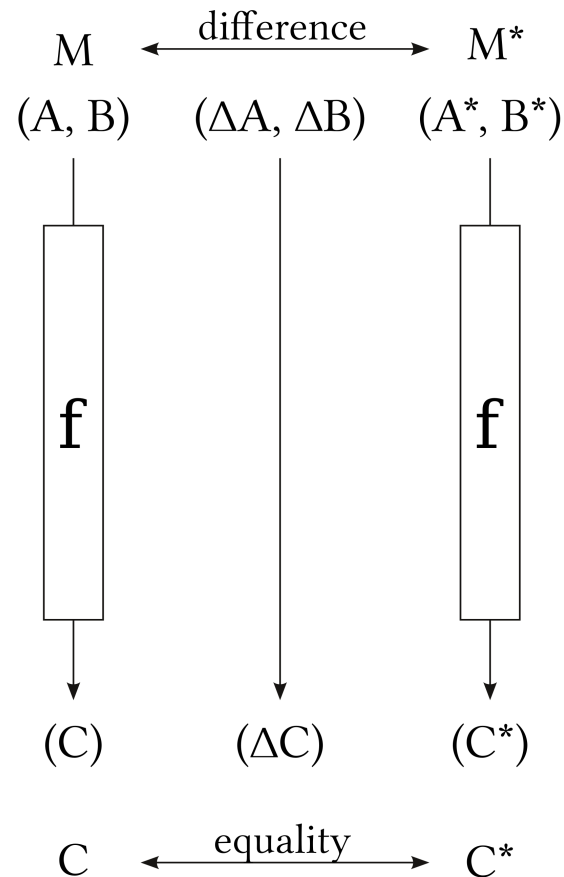
Goal

- Find hash collisions . . .
- for MD4 and SHA-256 . . .
- using SAT solvers

Outline

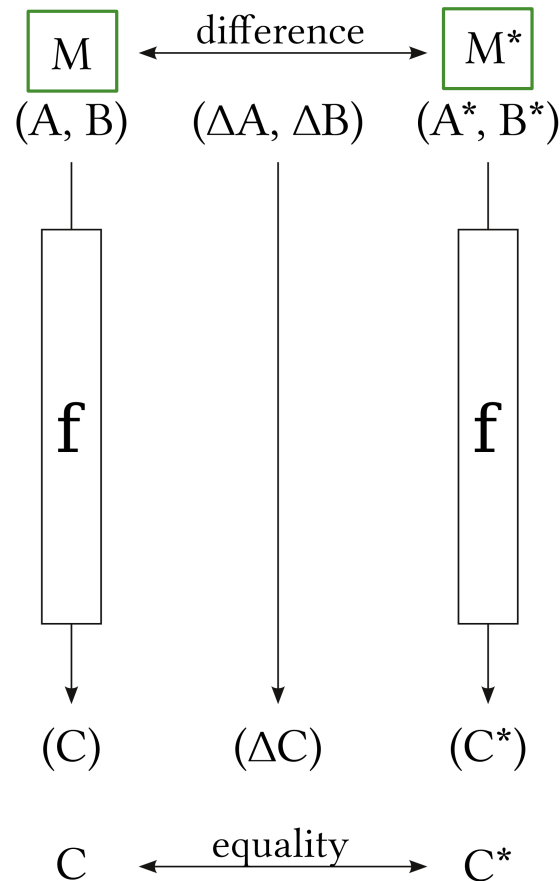
- Differential cryptanalysis
- SHA-256 and MD4
- Satisfiability
- Example hash collision (Wang et al.)
- MD4 testcases & results
- Tweaks to SAT encoding
- SHA-256 testcases & results

Differential cryptanalysis



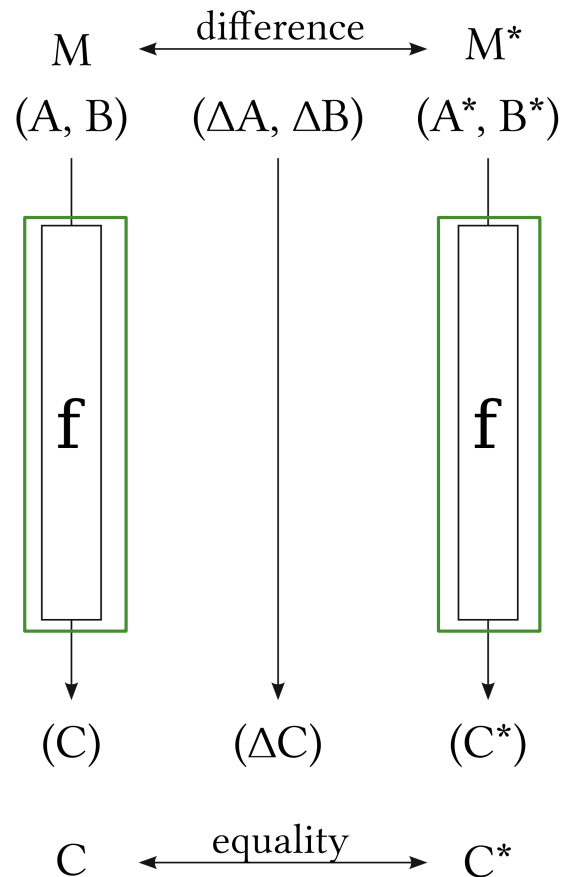
Differential cryptanalysis

We use two slightly different input messages

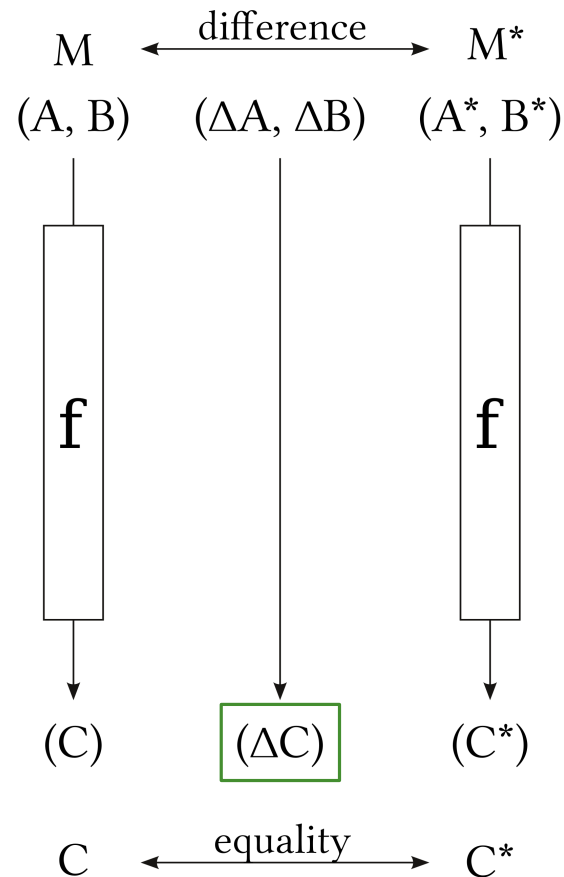


Differential cryptanalysis

We apply the cryptographic algorithm to both instances

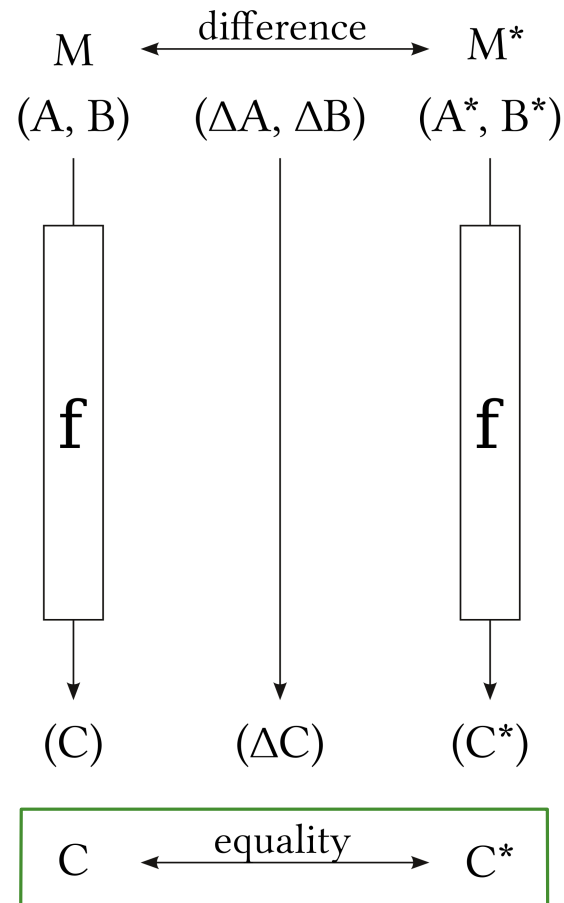


Differential cryptanalysis



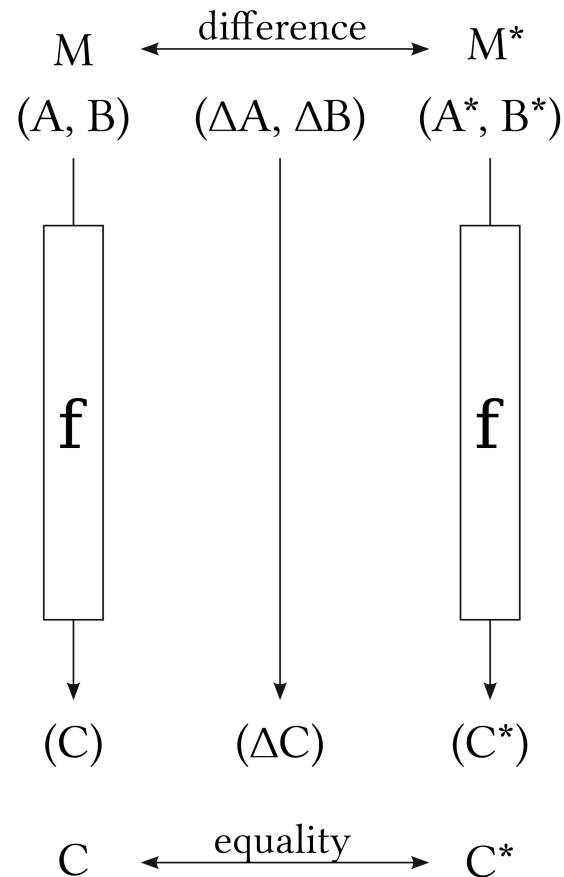
Differences cancel out

Differential cryptanalysis



Equality is given

Differential cryptanalysis



A *hash collision* is a pair (x_1, x_2) such that $x_1 \neq x_2$ with $f(x_1) = f(x_2)$.

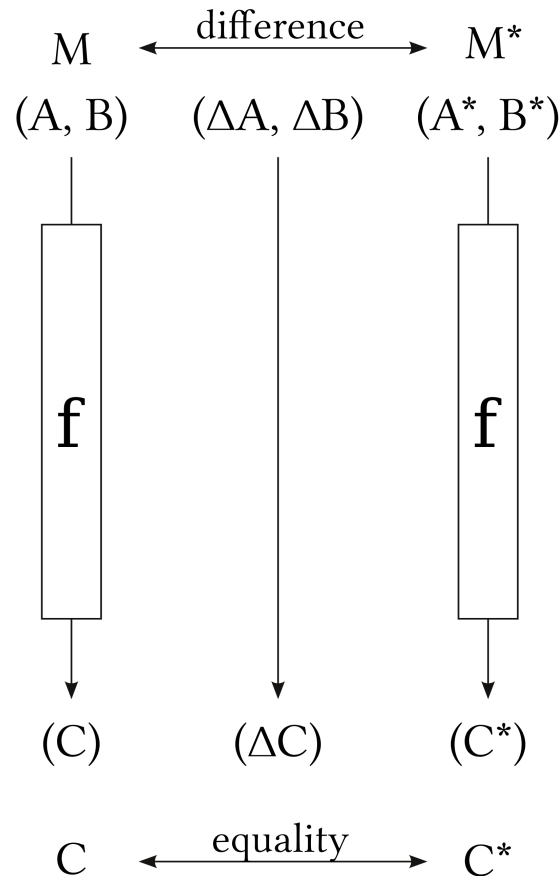
Differential cryptanalysis

We use two slightly different input messages

We apply the cryptographic algorithm to both instances

Differences cancel out

Equality is given



A *hash collision* is a pair (x_1, x_2) such that $x_1 \neq x_2$ with $f(x_1) = f(x_2)$.

Differential cryptanalysis focuses on the behavior of differences when f is progressing.

Differential cryptanalysis

Wang et al. defined signed differences.
De Cannière and Rechberger [01] defined *generalized bit conditions*.

(x_i, x_i^*)	(0,0)	(1,0)	(0,1)	(1,1)	(x_i, x_i^*)	(0,0)	(1,0)	(0,1)	(1,1)
?	✓	✓	✓	✓	3	✓	✓		
-	✓			✓	5	✓		✓	
x		✓	✓		7	✓	✓	✓	
0	✓				A		✓		✓
u		✓			B	✓	✓		✓
n			✓		C			✓	✓
1				✓	D	✓		✓	✓
#					E		✓	✓	✓

Differential cryptanalysis

addition $A + B = S$

A: 0100

B: 1x-1

S: ??-1

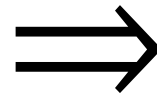
Differential cryptanalysis

addition $A + B = S$

A: 0100

B: 1x-1

S: ??-1



A:	0100	0100	0100	0100	
B:	1001	1011	1101	1001	...
S:	1101	1111	0001	1111	
A:	0100	0100	0100	0100	
B:	1101	1111	1001	1101	...
S:	0001	0011	1101	1011	

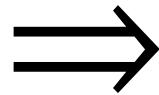
Differential cryptanalysis

addition $A + B = S$

A: 0100

B: 1x-1

S: ??-1



A:	0100	0100	0100	0100	
B:	1001	1011	1101	1001	...
S:	1101	1111	0001	1111	
<hr/>					
A:	0100	0100	0100	0100	
B:	1101	1111	1001	1101	...
S:	0001	0011	1101	1011	

Differential cryptanalysis

addition $A + B = S$

A: 0100

B: 1x-1

S: ?x-1



A:	0100	0100	0100	0100	
B:	1001	1011	1101	1001	...
S:	1101	1111	0001	1111	
<hr/>					
A:	0100	0100	0100	0100	
B:	1101	1111	1001	1101	...
S:	0001	0011	1101	1011	

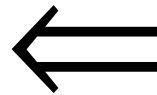
Differential cryptanalysis

addition $A + B = S$

A: 0100

B: 1x-1

S: ?x-1



A:	0100	0100	0100	0100	
B:	1001	1011	1101	1001	...
S:	1101	1111	0001	1111	...
A:	0100	0100	0100	0100	
B:	1101	1111	1001	1101	...
S:	0001	0011	1101	1011	

Valid 4-bit addition differential characteristics:

A: 0011
B: 0101
S: 1000

A: ---x
B: ---x
S: ????

A: ---x
B: ---x
S: ???-

A: ---x
B: ---x
S: x???

Invalid 4-bit addition differential characteristics:

A: 0011
B: 0101
S: 0000

A: ---x
B: ---x
S: ???x

A: ----
B: ---x
S: x-??

Differential cryptanalysis

We have constraints due to

- bit conditions (differential characteristic).
- the operation applied.

Our f is MD4 or SHA-256

MD4:

- Ronald Rivest, 1990
- RFC 1320
- broken since 1995
- 128 bits internal state size

Our f is MD4 or SHA-256

SHA-256:

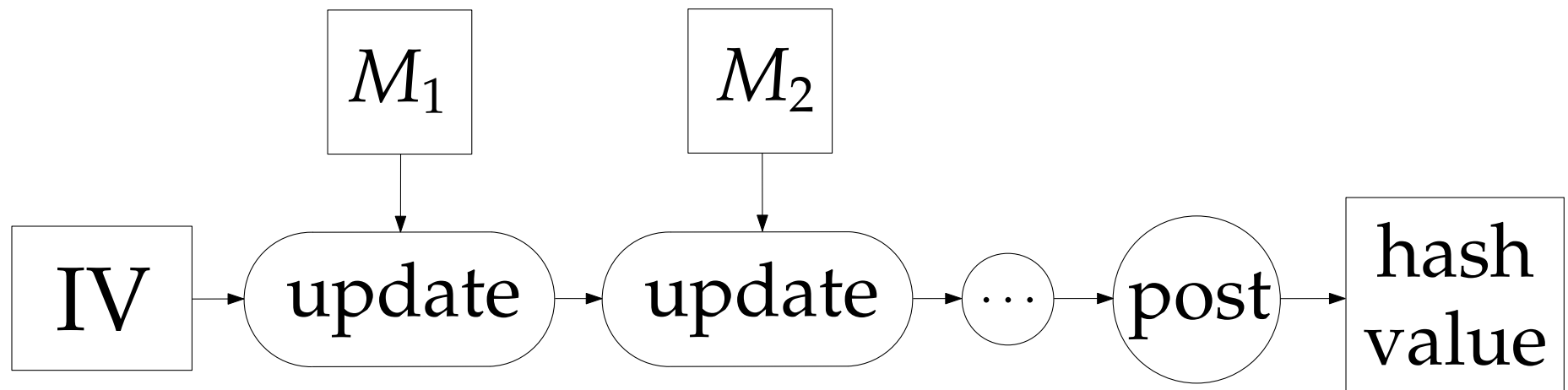
- NSA, 2001
- NIST publication 180-4
- best practical pseudo-collision attack breaks 38 rounds [02]
- 256 bits internal state size

Our f is MD4 or SHA-256

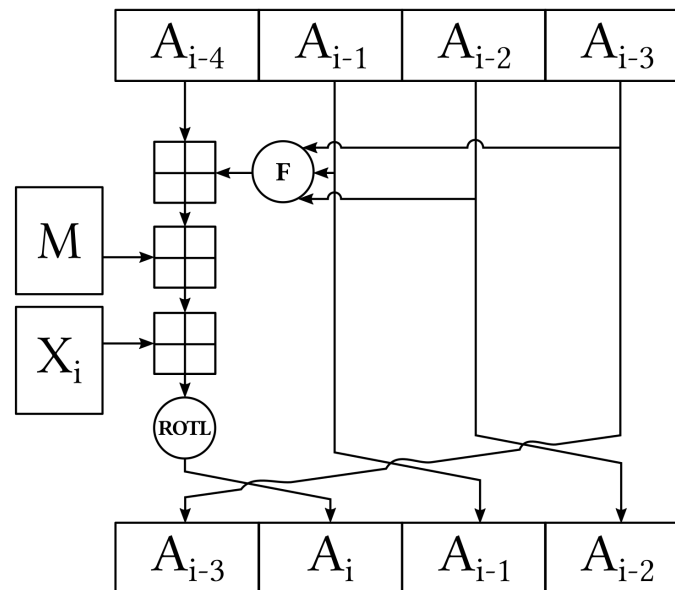
Merkle-Damgård constructions:

MD4: 48 steps

SHA-256: 64 steps

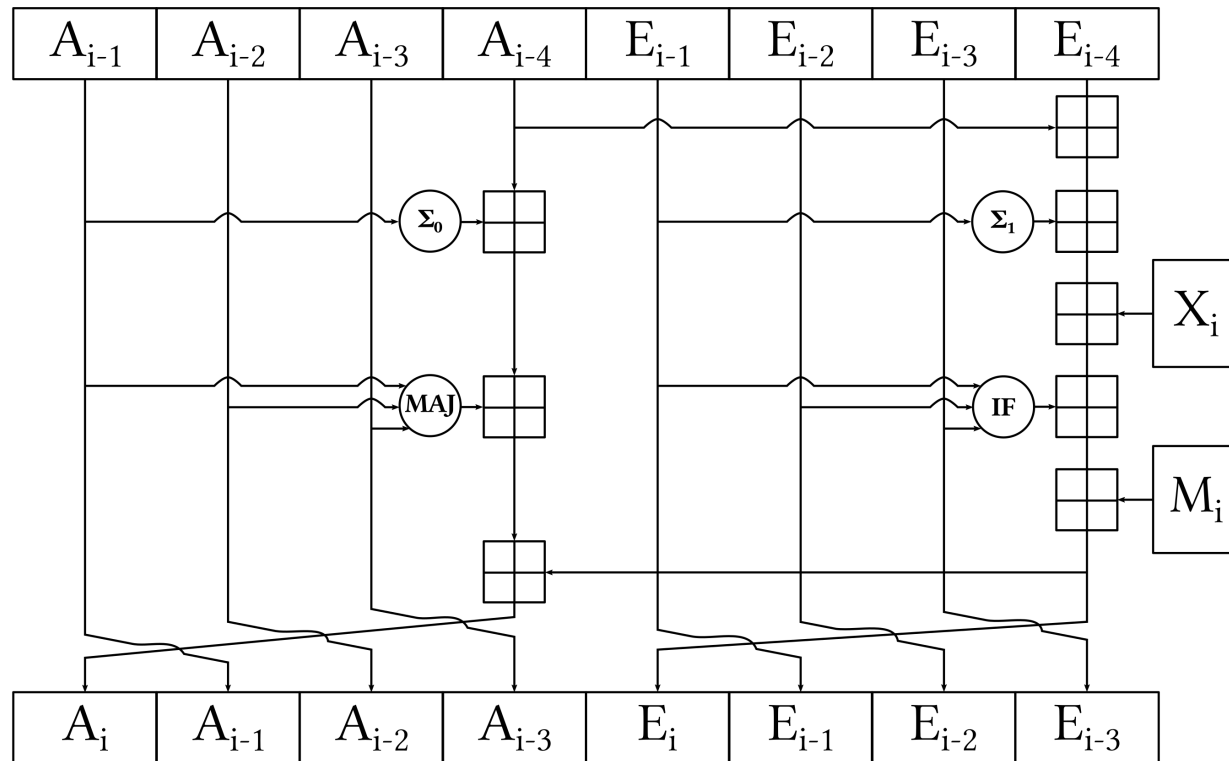


Our f is MD4 or SHA-256



MD4 update function

Our f is MD4 or SHA-256



SHA-256 update function

And now

for something completely different



Satisfiability

- A Boolean function is a mapping $h : X \rightarrow Y$ with $X = \{0, 1\}^n$ for $n \in \mathbb{N}_{\geq 1}$ and $Y = \{0, 1\}$.
- *AND* maps $X = \{0, 1\}^2$ to 1 iff $X = (1, 1)$.

x_1	x_2	$f(x_1, x_2)$
1	1	1
1	0	0
0	1	0
0	0	0

Satisfiability

A Boolean function f is *satisfiable* iff there exists some assignment m such that $f(m) = 1$.

$$f(x_1, x_2) = (x_1 \vee \neg x_2) \wedge (\neg x_1)$$

f is satisfiable because $m = (0, 0)$ evaluates to 1.

A *SAT solver* is a tool to evaluate whether a certain Boolean function is satisfiable or not.

Satisfiability

$$f(x_1, x_2) =$$

$$(x_1 \vee \neg x_2) \wedge$$

$$(\neg x_1)$$

p cnf 2 2

1 -2 0

-1 0

c Lingeling SAT Solver

c

s SATISFIABLE

v -1 -2 0

c

c 0 decisions, 0.0 decisions/sec

c 0 conflicts, 0.0 conflicts/sec

c 2 propagations, 0.0 megaprops/sec

c 0.0 seconds, 0.0 MB

$$m = (0, 0)$$

Satisfiability

Our strategy: We write a boolean equation system such that a hash collision is given iff the formula is satisfiable (constraints by bit conditions & operation).

Example hash collision

Recall . . .

- how bit conditions propagated
- how MD4 and SHA-256 worked



Example hash collision

Wang, Lai, Feng, Chen, Yu

“Cryptanalysis of the Hash Functions MD4 and RIPEMD”

EUROCRYPT 2005

i	A	W
-4	A: 011001 110100010 1001000110000001	
-3	A: 0001000000 110010010101000 11101 10	
-2	A: 1001100010 1110101101 11001 11111 10	
-1	A: 111011 111100110 11010 10111 00010 01	
0	A: 011010 1111 01010 01110 01000 0100 10	W: 0100110 10111 1010100111 001000011
1	A: 011101 100100111 11110 11100 1100 01	W: 10 101101100 10111 00100 10011 11010
2	A: 101010 110100000 00111 0 0 1 1 1100 10	W: 01 100 11101 01011 01001 01011 11000
3	A: 101011 100 11110 10101 00100 10100 01	W: 010 1011 11010 01111 01001 01111 01110
4	A: 001011 0001 10001 10101 01011 11100 10	W: 110 1111 00111 01001 00010 10001 11100
5	A: 000110 1001 10001 010 1 10100 00000 01	W: 110 1110 01100 00110 11001 10101 10011
6	A: 000110 1100 10001 10001 00000 11110 10	W: 101 1011 01000 00111 01000 00001 00000
7	A: 001010 1110 00000 10 0000 01100 10100 00	W: 001 1101 10010 10100 10111 01100 11111
8	A: 011100 1100 10001 1 1111 11111 01100 00	W: 110 0011 01001 11010 11100 01101 10011
9	A: 101011 1 01 10000 00 01111 10011 00111 11	W: 111 1100 11110 10011 00100 01100 11000
10	A: 1 0 001 001000010 10100 00001 01011 10	W: 110 1011 11001 11111 00000 00010 11110
11	A: 1 10001 1010 11011 00100 10101 11111 11	W: 101 0011 00011 10111 01100 10111 01000
12	A: 010101 1 00 10101 11111 11000 11110 11	W: 010 0010 11101 110 1 00011 10001 10001
13	A: 1 0 1 1 0100 10100 10100 00011 11001 01	W: 100 1011 11110 00110 00111 11111 00101
14	A: 000010 1001 01000 11000 10001 10101 10	W: 001 0011 11001 01001 01111 11000 01000
15	A: 000111 1010 101 1 0 10110 01101 10101 00	W: 101 1100 11110 10001 10000 11111 01001
16	A: 1 00 0 1 011 01001 01001 10110 10111 11	
17	A: 000111 1100 11101 00001 00100 00111 10	
18	A: 010101 1100 00110 10000 00001 00101 00	
19	A: 1 1 100 0000 01011 11001 10101 10001 00	
20	A: 1 1 100 100111 11110 11101 00000 01101 00	
21	A: 1 11100 1110 11000 00101 11111 10101 00	
22	A: 010111 0111 00110 10011 00110 01110 10	
23	A: 010100 0011 10111 01100 01111 00011 11	
24	A: 000000 1000 01001 00011 01110 00110 10	
25	A: 1 01100 0010 01011 00001 01001 11010 10	
26	A: 000010 1010 00100 10111 01110 10000 01	
27	A: 000001 1011 10111 10101 10101 01100 11	
28	A: 1 01101 1001 01110 10110 11000 01001 01	
29	A: 1 01000 1000 00110 10100 10000 11010 01	
30	A: 001010 0111 01011 11100 01110 11000 11	
31	A: 1 11111 0010 01001 01101 01111 01101 10	
32	A: 010011 1111 01001 00110 10000 01011 11	
33	A: 001110 0000 11110 10110 11101 11001 00	
34	A: 001000 0001 11010 11110 10000 00101 01	
35	A: 1 01000 0000 11001 10000 01000 11100 10	
36	A: 1 00001 1111 10101 11101 11100 10110 01	
37	A: 1 10010 0000 01101 00100 00110 00011 00	
38	A: 1 01100 0001 10011 11110 10011 01011 00	
39	A: 000100 1000 00101 00001 10110 00111 00	
40	A: 1 10000 0001 00100 00111 00011 00001 01	
41	A: 000001 1010 00011 01111 01010 01001 10	
42	A: 010011 1011 01110 11111 11101 00001 10	
43	A: 010100 0001 10001 11101 00000 11011 01	
44	A: 1 11110 0000 01011 01111 01110 00011 00	
45	A: 1 00010 1011 01101 10010 11000 00001 00	
46	A: 1 00000 1010 01100 10101 10001 10111 00	
47	A: 1 00000 0111 10010 11011 01001 01111 01	

Example hash collision

i	A	W
-4	A: 01100111010001010010001100000001	
-3	A: 00010000001100100101010001110110	
-2	A: 10011000101110101101110011111110	
-1	A: 1110111111001101101010110001001	
0	A: 011010111101010011100100000010010	W: 01001101011110101001110010000011
1	A: 0111011001001111111011100110001	W: 101011011001011100100100100111010
2	A: 101010110100000001110001110010	W: 1011001110101010101001010111000
3	A: 1010111001111010101001001010001	W: 0101011101001111010001011101110
4	A: 0010110001100011010101011110010	W: 1101111001101001000101000111100
5	A: 000110100110001010110100000001	W: 1101110011000011011001101010011
6	A: 00011011001000110001000001111010	W: 101101101000001101000000100000
7	A: 0010101110000010000110010101000	W: 00111011001010100101110110011111
8	A: 011100110010001111111110110000	W: 1100011010011101011000110110011
9	A: 101011101000000111110011001111	W: 11111001111010011001000110011000
10	A: 10001001000010100000010101110	W: 110101111001111100000001011110
11	A: 100011010110110010010101111111	W: 1010011000110111011001011101000
12	A: 00101110011010111111110001111011	W: 0100010111011101000111000110001
13	A: 1001101001100010100000111100101	W: 100101111110001100011111100101
14	A: 0000101001010001100010001101010	W: 0010011110010100101111100010000
15	A: 000111010101010110101101010100	W: 10111001111010001100001111101000
16	A: 000001011010010100101101011111	
17	A: 000111100111010000100100001110	
18	A: 0101011100001101000000010010100	
19	A: 1000000101111001101011000100	
20	A: 10010011111011101000000110100	
21	A: 11110011101100000101111111010100	
22	A: 01011101100110011001100111010	
23	A: 010100001110110110001111000111	
24	A: 000000100010001000110111000110	
25	A: 101100001010100000100011101010	
26	A: 000010101000100101101110100001	
27	A: 000001101110111010101010110011	
28	A: 10110110010111010110110001000101	
29	A: 10100010000011010100100001101001	
30	A: 0010100111010111100011101100011	
31	A: 11111100101000101010111110101010	
32	A: 0100111111010010011010000101111	
33	A: 0011100000111101011011011100100	
34	A: 0010000011101011110100000010101	
35	A: 01000000011001100000100001110010	
36	A: 0000011111101011110111001011001	
37	A: 11001000000110100100001100001100	
38	A: 1011000001100111110100110101100	
39	A: 00010010000010100001101100011100	
40	A: 1100000001001000011000110000101	
41	A: 00000110100001101110101010010	
42	A: 010011101101110111111101000010	
43	A: 01010000011000111101000001101101	
44	A: 1111100000101010111011100001100	
45	A: 1000101011010110010110000000100	
46	A: 10000010100110010100011011100	
47	A: 100000111100101101010010111101	

Example hash collision

i	A	W
-4	A: 01100111010001010010001100000001	
-3	A: 00010000001100100101010001110110	
-2	A: 10011000101110101101110011111110	
-1	A: 111011111100110110101110001001	
0	A: 01101011110101001110010000010010	W: 01001101011110101001110010000011
1	A: 0111011001001111111011100110001	W: 11010110110010111001001001111010
2	A: 101010110100000001110001110010	W: 10111001110101011010010101110000
3	A: 1010111001111010101001001010001	W: 0101011110100111101001011101110
4	A: 0010110001100011010101011110010	W: 11011110011101001000101000111000
5	A: 000110100110001010110100000001	W: 1101110011000011011001101010011
6	A: 00011011001000110001000001111010	W: 1011011010000011101000000100000
7	A: 00101011100000100001100101010000	W: 00111011001010100101110110011111
8	A: 0111001100100011111111101100000	W: 11000110100111010111000110110011
9	A: 1010111010000000111110011001111	W: 11111001111010011001000110011000
10	A: 100010010000101010000010101110	W: 11010111100111110000000101110
11	A: 100011010110110010010101111111	W: 1010011000110111011001011101000
12	A: 00101110011010111111110001111011	W: 010001011101110100011000110001
13	A: 100010100110001010000011100101	W: 100101111110001100011111100101
14	A: 000010100100011000100011010110	W: 001001111001010010111110001000
15	A: 000111010101010110101101010100	W: 10111001111010001100001111101001
16	A: 000001011010010100110110101111	
17	A: 000111100111010000100100001110	
18	A: 0101011100001101000000010010100	
19	A: 1000000101111001101011000100	
20	A: 10010011111011101000000110100	
21	A: 1111001110110000010111111010100	
22	A: 01011101100110011001100111010	
23	A: 010100001110110110001111000111	
24	A: 000000100010001000110111000110	
25	A: 101100001010100001010011101010	
26	A: 00001010100100101101101000001	
27	A: 000001101110111010101010110011	
28	A: 1011011001011101011010001000101	
29	A: 101000100000110101000001101001	
30	A: 0010100111010111100011101100011	
31	A: 11111100101000101010111110101010	
32	A: 0100111111010010011010000101111	
33	A: 0011100000111101011011011100100	
34	A: 001000001110101111010000010101	
35	A: 0100000001100110000010001110010	
36	A: 0000011111101011110111001011001	
37	A: 11001000000110100100001100001100	
38	A: 1011000001100111110100110101100	
39	A: 0001001000001000001101100011100	
40	A: 11000000010010000111000110000101	
41	A: 0000011010000110111010101000110	
42	A: 0100111011011101111111010000110	
43	A: 01010000011000111101000001101101	
44	A: 11111000001010111101110100001100	
45	A: 1000101011010110010110000000100	
46	A: 10000010100110010100011011100	
47	A: 100000111100101101010010111101	

input
message

Example hash collision

initial vectors

i	A	W
-4	A: 01100111010001010010001100000001	
-3	A: 00010000001100100101010001110110	
-2	A: 00011000101110101101110011111110	
-1	A: 11011111100110110101110001001	
0	A: 01101011110101001110010000010010	W: 010011010111010100111001000001
1	A: 0111011001001111111011100110001	W: 1101011011001011100100100111010
2	A: 101010110100000001110001110010	W: 101100111010101101001010111000
3	A: 1010111001111010101001001010001	W: 01010111010011110100101110110
4	A: 0010110001100011010101011110010	W: 1101111001110100100010100011100
5	A: 000110100110001010110100000001	W: 110111001100001101001101010011
6	A: 00011011001000110001000001111010	W: 10110110100000110100000010000
7	A: 001010111000001000011001010000	W: 00111011001010101011011011111
8	A: 011100110010001111111110110000	W: 1100011010011101011100011010011
9	A: 10101110110000000111110011001111	W: 11111001111010011001000110011000
10	A: 100010010000101010000010101110	W: 11010111100111110000000101110
11	A: 100011010110110010010101111111	W: 1010011000110111011001011101000
12	A: 00101110011010111111110001111011	W: 010001011101110100011000110001
13	A: 100110100110001010000011100101	W: 100101111110001100011111100101
14	A: 000010100101000110001001010101	W: 001001111001010010111110001000
15	A: 000111010101010110101101010100	W: 10111001111010001100001111101001
16	A: 000001011010010100101101011111	
17	A: 000111100111010000100100001110	
18	A: 0101011100001101000000010010100	
19	A: 1000000101111001101011000100	
20	A: 10010011111011101000000110100	
21	A: 1111001110110000010111111010100	
22	A: 01011101100110001001100111010	
23	A: 010100001110110110001111000111	
24	A: 00000010001001000110111000110	
25	A: 101100001010100001010011101010	
26	A: 00001010100100101101101000001	
27	A: 00000110111011010101010101100	
28	A: 1011011001011101011010001000101	
29	A: 101000100000110101000001101001	
30	A: 0010100111010111100011101100011	
31	A: 111111001010010101010111101010	
32	A: 0100111110100100110100000101111	
33	A: 0011100000111101011011011100100	
34	A: 00100000111010111101000000101	
35	A: 0100000001100110000010001110010	
36	A: 0000011111101011110111001011001	
37	A: 11001000000110100100001100001100	
38	A: 1011000001100111110100110101100	
39	A: 000100100000100001101100011100	
40	A: 1100000001001000011000110000101	
41	A: 000001101000011011101010100110	
42	A: 0100111011011101111111010000110	
43	A: 01010000011000111101000001101101	
44	A: 1111100000101010111011100001100	
45	A: 100010101101010010110000000100	
46	A: 10000010100110010100011011100	
47	A: 100000111100101101010010111101	

input message

Example hash collision

initial vectors

i	A	W
-4	A: 01100111010001010010001100000001	
-3	A: 00010000001100100101010001110110	
-2	A: 00011000101110101101110011111110	
-1	A: 1101111100110110101110001001	
0	A: 011010111101010011100100000010010	W: 010011010111010100110010000011
1	A: 0111011001001111110111000110001	W: 01010110110010111001001000111010
2	A: 101010110100000001110001110010	W: 001100111010101101001010111000
3	A: 10101101001111010101001001010001	W: 010101110100111101001011101101
4	A: 0010110001100011010101011110010	W: 1101111001110100100010100011100
5	A: 000110100110001010110100000001	W: 1101110011000011011001101010011
6	A: 00011011000000110001000001111010	W: 101101101000001101000000100000
7	A: 00101011100000100000110010101000	W: 00111011001010100101110110011111
8	A: 011100110010001011111110110000	W: 110001101001110101110001010011
9	A: 101011001000000111110011001111	W: 11111001111010011001000110011000
10	A: 100010010000101010000010101110	W: 11010111100111110000000101110
11	A: 0000110101101010010010101111111	W: 1010011000110111011001011101000
12	A: 0010110001010111111110001111011	W: 010001011011100100011000110001
13	A: 100010100110001010000011100101	W: 100101111110001100011111100101
14	A: 0000101001010001100010010101010	W: 001001111001010010111110001000
15	A: 0001110101010101011010101010100	W: 10111001111010001100001111101001
16	A: 000000010100010100101010101111	
17	A: 000111100111010000100100001110	
18	A: 0101011100001101000000010010100	
19	A: 01000000101111001101011000100	
20	A: 010010011111011101000001010100	
21	A: 1111001110110000010111111010100	
22	A: 01011101100110001001100110111010	
23	A: 010100001110110110001111000111	
24	A: 00000010001001000110111000110	
25	A: 101100001010100001010011101010	
26	A: 0000101010001001011011101000001	
27	A: 0000011011101101010101010110011	
28	A: 1011011001011101011010001000101	
29	A: 101000100000110100100001101001	
30	A: 001010011010111100011101100011	
31	A: 1111100101001010101011110101010	
32	A: 010011111010010011010000101111	
33	A: 001110000111101011011101100100	
34	A: 00100000111011110100000010101	
35	A: 010000001100110000010001110010	
36	A: 000001111101011110111001011001	
37	A: 11001000001101001000001100001100	
38	A: 1011000001100111110100110101100	
39	A: 0001001000001010000110110001100	
40	A: 110000001001000011000110000101	
41	A: 00000110100001101110101010010	
42	A: 010011101101110111111101000010	
43	A: 0101000011000111101000001101101	
44	A: 11110000010110111011100001100	
45	A: 100010101101010010110000000100	
46	A: 10000010100110010101100011011100	
47	A: 10000001111001011011010010111101	

input message

output words

Example hash collision

initial vectors

intermediate values

output words

i	A	W
-4	A: 01100111010001010010001100000001	
-3	A: 00010000001100100101010001110110	
-2	A: 10011000101110101101110011111110	
-1	A: 1101111110011011010101110001001	
0	A: 011010111101010011100100000010010	W: 010011010111010100110010000011
1	A: 0111011001001111111011100110001	W: 11010110110010111001001000111010
2	A: 010101101000000001110001110010	W: 1011001110101010101001010111000
3	A: 010111001111010101001001010001	W: 01010111010011110100101110110110
4	A: 010110001100011010101011110010	W: 1101111001110100100010100011100
5	A: 000110100110001010110100000001	W: 1101110011000011011001101010011
6	A: 00011011001000110001000001111010	W: 1011011010000011101000000100000
7	A: 010101111000001000011001010000	W: 00111011001010100101110110011111
8	A: 011100110010001111111110110000	W: 1100011010011101011000110110011
9	A: 010111010000001111100110011111	W: 11111001111010011001000110011000
10	A: 0001001000101010000010101110	W: 11010111100111110000000101110
11	A: 100011010110100100101011111111	W: 1010011000110111011001011101000
12	A: 01010110011010111111110001111011	W: 010001011101110100011000110001
13	A: 100110100110001010000011100101	W: 100101111110001100011111100101
14	A: 000010100101000110001001010101	W: 001001111001010010111110001000
15	A: 00011101010101011010101010100	W: 10111001111010001100001111101001
16	A: 000010110100010100101010101111	
17	A: 000111100111010000100100001110	
18	A: 0101011100001101000000010010100	
19	A: 11000000101111001101011000100	
20	A: 11001011111011101000001010100	
21	A: 1111001110110000010111111010100	
22	A: 01011011100110001100110111010	
23	A: 0101000011101101100011110001111	
24	A: 0000010001001000110111000110	
25	A: 01100001010100001000110101010	
26	A: 00001010100010010110111000001	
27	A: 0000011011101101010101010110011	
28	A: 0110110010111010110110001000101	
29	A: 0100010000011010100001101001	
30	A: 0010100111010111100011101100011	
31	A: 111110010100101010111110101010	
32	A: 010011111010010011010000101111	
33	A: 0011100001111010110111011100100	
34	A: 01000000111010111101000000101	
35	A: 010000001100110000010001110010	
36	A: 00001111101011110111001011001	
37	A: 1100100000110100100000110000100	
38	A: 011000001100111110100110101100	
39	A: 0001001000010100001101100011100	
40	A: 110000001001000011000110000101	
41	A: 000001101000011011101010100110	
42	A: 01001110110111111111010000110	
43	A: 0101000011000111101000001101101	
44	A: 1111100000101101111011100001100	
45	A: 0000101101101010010110000000100	
46	A: 10000010100110010101100011011100	
47	A: 1000001111001011011010010111101	

input message

Example hash collision

underspecified

i	A	W
-4	A: 01100111010001010010001100000001	
-3	A: 00010000001100100101010001110110	
-2	A: 10011000101110101101110011111110	
-1	A: 1110111111001101101010110001001	
0	A: 0110101111010101110010000010010	W: 010011010111010100111001000011
1	A: 0111011001001111111011100u110001	W: u101011011001011100100100100111010
2	A: 101010110100000001110u01n1110010	W: n01n1001110101010100010101111000
3	A: 101011u1001111010101001001010001	W: 0101011101001111010001011110110
4	A: 0010110001100011010101011110010	W: 1101111001110100100010100011100
5	A: 000110100110001010u110100000001	W: 11011100110000110110011010110011
6	A: 0001101100uuu110001000001111010	W: ??? ????? ????? ????? ????? ?????
7	A: 0010101110000010uum011001010000	W: 001110110010101001011101100111111
8	A: ????? ????? ????? ????? ????? ?????	W: 11000110100111010111000110110011
9	A: ????? ????? ????? ????? ????? ?????	W: 1111100111101001100100010011000
10	A: 10u001001000101 ????? ????? ????? 110	W: 11010111100111111000000001011110
11	A: u100011010110110 ????? ????? ????? 111	W: 1010011000110111011001011101000
12	A: 001011u00u10101 ????? ????? ????? 011	W: 010001011101110u1000111000110001
13	A: 10um1n010011000 ????? ????? ????? 101	W: 100101111100011000111111100101
14	A: 0000101001010001 ????? ????? ????? 110	W: 0010011110010100101111100001000
15	A: 0001111010101u01 ????? ????? ????? 100	W: 101110011110100011000011111010001
16	A: n00u0u01101001 ????? ????? ????? 111	
17	A: 000111100111010 ????? ????? ????? 110	
18	A: 010101110000110 ????? ????? ????? 100	
19	A: u1n10000001011 ????? ????? ????? 100	
20	A: n1um100111111011101000000110100	
21	A: 11110011101100000010111111010100	
22	A: 01011101110011010011001100111010	
23	A: 010100001101110110001111000111	
24	A: 0000001000010001000110111000110	
25	A: 101100001010101000001010011101010	
26	A: 00001010100010010111011101000001	
27	A: 000001101110111010101010101011	
28	A: 101101100101110101101100001000101	
29	A: 10100010000011010100100001101001	
30	A: 00101001110101111000111011000111	
31	A: 111111001001001011010111101011010	
32	A: 010011111101000100110100000101111	
33	A: 00111000001111010110111011100100	
34	A: 00100000011101011110100000010101	
35	A: n01000000011001100000100001110010	
36	A: n0000111111010111101111001011001	
37	A: 11001000000110100100001100001100	
38	A: 1011000001100111110100110101100	
39	A: 00010010000010100001101100011100	
40	A: 11000000010010000111000110000101	
41	A: 00000110100001101111010101000110	
42	A: 01001110110111011111111010000110	
43	A: 0101000001000111101000001101101	
44	A: 11111000000101101111011100001100	
45	A: 1000101011010100101100000000100	
46	A: 10000010100110010101100011011100	
47	A: 10000001111001011011010010111101	

Attack!



Attacking MD4 & SHA-256

MD4 testcases & results

i	A	W
-4	A: 0110011101000101001000110000001	
-3	A: 00010000001100100101010001110110	
-2	A: 10011000101110101101110011111110	
-1	A: 11101111110011011010101110001001	
0	A: x-----	W: x-----
1	A: -----	W: -----
2	A: -----	W: -----
3	A: xxx-----	W: x-----
4	A: -----	W: -----
5	A: -----	W: -----
6	A: x-----	W: x-----
7	A: -----	W: -----
8	A: -----	W: -----
9	A: -----	W: -----
10	A: -----	W: -----
11	A: x-----	W: -----
12	A: -----	W: -----
13	A: -----	W: -----
14	A: -x-----	W: -----
15	A: x-x-----	W: -----
16	A: -xxx-----	W: -----
17	A: -----	W: -----
18	A: -----	W: -----
19	A: x-----	W: -----
20	A: x-----	W: -----
21	A: -----	W: -----
22	A: -----	W: -----
23	A: -----	W: -----
24	A: -----	W: -----
25	A: -----	W: -----
26	A: -----	W: -----
27	A: -----	W: -----
28	A: -----	W: -----
29	A: -----	W: -----
30	A: -----	W: -----
31	A: -----	W: -----
32	A: x-----	W: -----
33	A: -----	W: -----
34	A: -----	W: -----
35	A: -----	W: -----
36	A: -----	W: -----
37	A: -----	W: -----
38	A: -----	W: -----
39	A: -----	W: -----
40	A: -----	W: -----
41	A: -----	W: -----
42	A: -----	W: -----
43	A: -----	W: -----
44	A: -----	W: -----
45	A: -----	W: -----
46	A: -----	W: -----
47	A: -----	W: -----

Testcase A [03]

All differences set. Just determine actual bits in both instances.

MD4 testcases & results

i	A	W
-4	A: 0110011101000101001000110000001	
-3	A: 00010000001100100101010001110110	
-2	A: 10011000101110101101110011111110	
-1	A: 11101111110011011010101110001001	
0	A: x-----	W: x-----
1	A: -----	W: -----
2	A: -----	W: -----
3	A: -----	W: -----
4	A: xxx-----	W: -----
5	A: -----	W: -----
6	A: x-----	W: -----
7	A: -----	W: -----
8	A: -----	W: -----
9	A: -----	W: -----
10	A: -----	W: -----
11	A: -----	W: -----
12	A: -----	W: -----
13	A: -----	W: -----
14	A: -----	W: -----
15	A: -----	W: -----
16	A: -----	W: -----
17	A: -----	W: -----
18	A: -----	W: -----
19	A: -----	W: -----
20	A: -----	W: -----
21	A: -----	W: -----
22	A: -----	W: -----
23	A: -----	W: -----
24	A: -----	W: -----
25	A: -----	W: -----
26	A: -----	W: -----
27	A: -----	W: -----
28	A: -----	W: -----
29	A: -----	W: -----
30	A: -----	W: -----
31	A: -----	W: -----
32	A: -----	W: -----
33	A: -----	W: -----
34	A: -----	W: -----
35	A: -----	W: -----
36	A: -----	W: -----
37	A: -----	W: -----
38	A: -----	W: -----
39	A: -----	W: -----
40	A: -----	W: -----
41	A: -----	W: -----
42	A: -----	W: -----
43	A: -----	W: -----
44	A: -----	W: -----
45	A: -----	W: -----
46	A: -----	W: -----
47	A: -----	W: -----

Testcase A [03]

All differences set. Just determine actual bits in both instances.

always seconds

solver	version	runtime
MiniSat	2.2.0	3
CryptoMiniSat	4.5.3	26
	5	29
Lingeling	ats1	23
Plingeling	ats1	88
Treengeling	ats1	64
Glucose	4.0	8
Glucose Syrup	4.0	14

MD4 testcases & results

i	A	W
-4	A: 0110011101000101001000110000001	
-3	A: 00010000001100100101010001110110	
-2	A: 10011000101110101101110011111110	
-1	A: 11101111110011011010101110001001	
0	A: x-----	W: x-----
1	A: -----	W: -----
2	A: -----	W: -----
3	A: -----	W: -----
4	A: xxx-----	W: -----
5	A: -----	W: -----
6	A: x-----	W: -----
7	A: -----	W: -----
8	A: -----	W: -----
9	A: -----	W: -----
10	A: -----	W: -----
11	A: -----	W: -----
12	A: -----	W: -----
13	A: -----	W: -----
14	A: -----	W: -----
15	A: -----	W: -----
16	A: -----	W: -----
17	A: -----	W: -----
18	A: -----	W: -----
19	A: -----	W: -----
20	A: -----	W: -----
21	A: -----	W: -----
22	A: -----	W: -----
23	A: -----	W: -----
24	A: -----	W: -----
25	A: -----	W: -----
26	A: -----	W: -----
27	A: -----	W: -----
28	A: -----	W: -----
29	A: -----	W: -----
30	A: -----	W: -----
31	A: -----	W: -----
32	A: -----	W: -----
33	A: -----	W: -----
34	A: -----	W: -----
35	A: -----	W: -----
36	A: -----	W: -----
37	A: -----	W: -----
38	A: -----	W: -----
39	A: -----	W: -----
40	A: -----	W: -----
41	A: -----	W: -----
42	A: -----	W: -----
43	A: -----	W: -----
44	A: -----	W: -----
45	A: -----	W: -----
46	A: -----	W: -----
47	A: -----	W: -----

Interesting fact:
In 2006, Mironov and Zhang [04] evaluated such testcases *within 10 minutes*.

MD4 testcases & results

i	A	W
-4	A: 0110011101000101001000110000001	
-3	A: 00010000001100100101010001110110	
-2	A: 10011000101110101101110011111110	
-1	A: 11101111110011011010101110001001	
0	A: ?????? ?????? ?????? ?????? ?????? ??	W: -x-----
1	A: ?????? ?????? ?????? ?????? ?????? ??	W: -x-----
2	A: ?????? ?????? ?????? ?????? ?????? ??	W: -x-----
3	A: ?????? ?????? ?????? ?????? ?????? ??	W: -x-----
4	A: ?????? ?????? ?????? ?????? ?????? ??	W: -x-----
5	A: ?????? ?????? ?????? ?????? ?????? ??	W: -x-----
6	A: ?????? ?????? ?????? ?????? ?????? ??	W: -x-----
7	A: ?????? ?????? ?????? ?????? ?????? ??	W: -x-----
8	A: ?????? ?????? ?????? ?????? ?????? ??	W: -x-----
9	A: ?????? ?????? ?????? ?????? ?????? ??	W: -x-----
10	A: ?????? ?????? ?????? ?????? ?????? ??	W: -x-----
11	A: ?????? ?????? ?????? ?????? ?????? ??	W: -x-----
12	A: ?????? ?????? ??????-----	W: -x-----
13	A: ?????? ?????? ??????-----	W: -x-----
14	A: ?????? ?????? ??????-----	W: -x-----
15	A: ?????? ?????? ??????-----	W: -x-----
16	A: ???x-----	W: -x-----
17	A: ?-----	W: -x-----
18	A: ?-----	W: -x-----
19	A: ?-----	W: -x-----
20	A: x-----	W: -x-----
21	A: -----	W: -x-----
22	A: -----	W: -x-----
23	A: -----	W: -x-----
24	A: -----	W: -x-----
25	A: -----	W: -x-----
26	A: -----	W: -x-----
27	A: -----	W: -x-----
28	A: -----	W: -x-----
29	A: -----	W: -x-----
30	A: -----	W: -x-----
31	A: -----	W: -x-----
32	A: x-----	W: -x-----
33	A: -----	W: -x-----
34	A: -----	W: -x-----
35	A: -----	W: -x-----
36	A: -----	W: -x-----
37	A: -----	W: -x-----
38	A: -----	W: -x-----
39	A: -----	W: -x-----
40	A: -----	W: -x-----
41	A: -----	W: -x-----
42	A: -----	W: -x-----
43	A: -----	W: -x-----
44	A: -----	W: -x-----
45	A: -----	W: -x-----
46	A: -----	W: -x-----
47	A: -----	W: -x-----

Testcase B [03]

Words in rounds 0–11 completely undetermined.

MD4 testcases & results

i	A	W
-4	A: 0110011101000101001000110000001	
-3	A: 00010000001100100101010001110110	
-2	A: 10011000101110101101110011111110	
-1	A: 11101111110011011010101110001001	
0	A: ?????? ?????? ?????? ?????? ?????? ??	W: -x-----
1	A: ?????? ?????? ?????? ?????? ?????? ??	W: -x-----
2	A: ?????? ?????? ?????? ?????? ?????? ??	W: -x-----
3	A: ?????? ?????? ?????? ?????? ?????? ??	W: -x-----
4	A: ?????? ?????? ?????? ?????? ?????? ??	W: -x-----
5	A: ?????? ?????? ?????? ?????? ?????? ??	W: -x-----
6	A: ?????? ?????? ?????? ?????? ?????? ??	W: -x-----
7	A: ?????? ?????? ?????? ?????? ?????? ??	W: -x-----
8	A: ?????? ?????? ?????? ?????? ?????? ??	W: -x-----
9	A: ?????? ?????? ?????? ?????? ?????? ??	W: -x-----
10	A: ?????? ?????? ?????? ?????? ?????? ??	W: -x-----
11	A: ?????? ?????? ?????? ?????? ?????? ??	W: -x-----
12	A: ?????? ?????? ??????-----	W: -x-----
13	A: ?????? ?????? ??????-----	W: -x-----
14	A: ?????? ?????? ??????-----	W: -x-----
15	A: ?????? ?????? ??????-----	W: -x-----
16	A: ???x-----	W: -x-----
17	A: ?-----	W: -x-----
18	A: ?-----	W: -x-----
19	A: ?-----	W: -x-----
20	A: x-----	W: -x-----
21	A: -----	W: -x-----
22	A: -----	W: -x-----
23	A: -----	W: -x-----
24	A: -----	W: -x-----
25	A: -----	W: -x-----
26	A: -----	W: -x-----
27	A: -----	W: -x-----
28	A: -----	W: -x-----
29	A: -----	W: -x-----
30	A: -----	W: -x-----
31	A: -----	W: -x-----
32	A: x-----	W: -x-----
33	A: -----	W: -x-----
34	A: -----	W: -x-----
35	A: -----	W: -x-----
36	A: -----	W: -x-----
37	A: -----	W: -x-----
38	A: -----	W: -x-----
39	A: -----	W: -x-----
40	A: -----	W: -x-----
41	A: -----	W: -x-----
42	A: -----	W: -x-----
43	A: -----	W: -x-----
44	A: -----	W: -x-----
45	A: -----	W: -x-----
46	A: -----	W: -x-----
47	A: -----	W: -x-----

Testcase B [03]

Words in rounds 0–11 completely undetermined.

~ 20 minutes

MD4 testcases & results

i	A	W
-4	A: 0 11001 110100010 1001000110 0000001	
-3	A: 000100 0000 11001 00101 01000 11101 10	
-2	A: 1001100010 11101 01101 11001 11111 10	
-1	A: 1 11011 111100110 11010 10111 0001001	
0	A: ? ????? ???? ???? ???? ???? ???? ???? ??	W: x-----
1	A: ? ????? ???? ???? ???? ???? ???? ???? ??	W: -----
2	A: ? ????? ???? ???? ???? ???? ???? ???? ??	W: x-----
3	A: ? ????? ???? ???? ???? ???? ???? ???? ??	W: -----
4	A: ? ????? ???? ???? ???? ???? ???? ???? ??	W: x-----
5	A: ? ????? ???? ???? ???? ???? ???? ???? ??	W: -----
6	A: ? ????? ???? ???? ???? ???? ???? ???? ??	W: -----
7	A: ? ????? ???? ???? ???? ???? ???? ???? ??	W: -----
8	A: ? ????? ???? ???? ???? ???? ???? ???? ??	W: x-----
9	A: ? ????? ???? ???? ???? ???? ???? ???? ??	W: -----
10	A: ? ????? ???? ???? ???? ???? ???? ???? ??	W: -----
11	A: ? ????? ???? ???? ???? ???? ???? ???? ??	W: -----
12	A: ? ????? ???? ???? ???? ???? ???? ???? ??	W: x-----
13	A: ? ????? ???? ???? ???? ???? ???? ???? ??	W: -----
14	A: ? ????? ???? ???? ???? ???? ???? ???? ??	W: -----
15	A: ? ????? ???? ???? ???? ???? ???? ???? ??	W: -----
16	A: ? ????? ???? ???? ???? ???? ???? ???? ??	W: -----
17	A: ? ????? ???? ???? ???? ???? ???? ???? ??	W: -----
18	A: ? ????? ???? ???? ???? ???? ???? ???? ??	W: -----
19	A: ? ????? ???? ???? ???? ???? ???? ???? ??	W: -----
20	A: ? ????? ???? ???? ???? ???? ???? ???? ??	W: -----
21	A: -----	
22	A: -----	
23	A: -----	
24	A: -----	
25	A: -----	
26	A: -----	
27	A: -----	
28	A: -----	
29	A: -----	
30	A: -----	
31	A: -----	
32	A: x-----	
33	A: -----	
34	A: -----	
35	A: -----	
36	A: -----	
37	A: -----	
38	A: -----	
39	A: -----	
40	A: -----	
41	A: -----	
42	A: -----	
43	A: -----	
44	A: -----	
45	A: -----	
46	A: -----	
47	A: -----	

Testcase C [03]

Words in rounds 0–20 completely undetermined. Collision still given in round 32.

MD4 testcases & results

i	A	W
-4	A: 0 11001 1101 00010 10010 00110 0000001	
-3	A: 000100 0000 11001 00101 01000 11101 10	
-2	A: 100110 0010 11101 01101 11001 11111 10	
-1	A: 1 11011 1111 00110 11010 10111 0001001	
0	A: ?	W: x
1	A: ?	W: x
2	A: ?	W: x
3	A: ?	W: x
4	A: ?	W: x
5	A: ?	W: x
6	A: ?	W: x
7	A: ?	W: x
8	A: ?	W: x
9	A: ?	W: x
10	A: ?	W: x
11	A: ?	W: x
12	A: ?	W: x
13	A: ?	W: x
14	A: ?	W: x
15	A: ?	W: x
16	A: ?	W: x
17	A: ?	W: x
18	A: ?	W: x
19	A: ?	W: x
20	A: ?	W: x
21	A: ?	W: x
22	A: ?	W: x
23	A: ?	W: x
24	A: ?	W: x
25	A: ?	W: x
26	A: ?	W: x
27	A: ?	W: x
28	A: ?	W: x
29	A: ?	W: x
30	A: ?	W: x
31	A: ?	W: x
32	A: x	W: x
33	A: ?	W: x
34	A: ?	W: x
35	A: ?	W: x
36	A: ?	W: x
37	A: ?	W: x
38	A: ?	W: x
39	A: ?	W: x
40	A: ?	W: x
41	A: ?	W: x
42	A: ?	W: x
43	A: ?	W: x
44	A: ?	W: x
45	A: ?	W: x
46	A: ?	W: x
47	A: ?	W: x

Testcase C [03]

Words in rounds 0–20 completely undetermined. Collision still given in round 32.

~ 18 minutes

MD4 testcases & results

i	A	W
-4	A: 0 11001 1101 00010 10010 00110 0000001	
-3	A: 000100 0000 11001 00101 01000 11101 10	
-2	A: 100110 0010 11101 01101 11001 11111 10	
-1	A: 1 11011 1111 00110 11010 10111 0001001	
0	A: ?	W: x
1	A: ?	W: x
2	A: ?	W: x
3	A: ?	W: x
4	A: ?	W: x
5	A: ?	W: x
6	A: ?	W: x
7	A: ?	W: x
8	A: ?	W: x
9	A: ?	W: x
10	A: ?	W: x
11	A: ?	W: x
12	A: ?	W: x
13	A: ?	W: x
14	A: ?	W: x
15	A: ?	W: x
16	A: ?	W: x
17	A: ?	W: x
18	A: ?	W: x
19	A: ?	W: x
20	A: ?	W: x
21	A: ?	W: x
22	A: ?	W: x
23	A: ?	W: x
24	A: ?	W: x
25	A: ?	W: x
26	A: ?	W: x
27	A: ?	W: x
28	A: ?	W: x
29	A: ?	W: x
30	A: ?	W: x
31	A: ?	W: x
32	A: x	W: x
33	A: ?	W: x
34	A: ?	W: x
35	A: ?	W: x
36	A: ?	W: x
37	A: ?	W: x
38	A: ?	W: x
39	A: ?	W: x
40	A: ?	W: x
41	A: ?	W: x
42	A: ?	W: x
43	A: ?	W: x
44	A: ?	W: x
45	A: ?	W: x
46	A: ?	W: x
47	A: ?	W: x

Testcase C [03]

Words in rounds 0–20 completely undetermined. Collision still given in round 32.

~ 18 minutes

Nice, but we want to attack SHA-256, which is much more difficult.

Tweaking SAT encoding

Definition. Given two CNFs A and B , they are called *equisatisfiable* iff A is satisfiable iff B .

CNF simplification is done to transform the CNF into a representation which improves the performance of the SAT solver.

cmsat

lingeling

minisat

satelite

Tweaking SAT encoding

Simplification reduces the problem size.

simplification	variables	percent of none	clauses	percent of none
none	48,704	100 %	253,984	100 %
cmsat	24,503	50 %	111,931	44 %
lingeling	48,704	100 %	106,626	42 %
minisat	20,895	43 %	118,236	47 %
satelite	27,495	56 %	153,262	60 %

for Testcase C

Tweaking SAT encoding

Simplification as preprocessing step does not significantly improve the runtime of SAT solvers.

solver	version	none	cmsat	lingeling	minisat	satelite
MiniSat	2.2.0	4,519	7,649	1,337	1,476	1,293
CryptoMiniSat	5	1,064	973	1,201	4,470	3,920
Lingeling	ats1	1,492	906	356	860	1,297
Treengeling	ats1	1,281	13,401	20,903	13,790	10,840
Plingeling	ats1	2,310	1,232	955	1,384	2,030

for Testcase C

Tweaking SAT encoding

Definition. *Differential description*

Boolean function IF returns the second argument, if the first is true, otherwise the third argument.

Differential behavior:

$$(0, 1, 1) \implies 1 \quad (0, 0, 0) \implies 0$$

$$(\neg a \wedge b \wedge c) \implies r \iff a \vee \neg b \vee \neg c \vee r$$

$$(\neg a \wedge \neg b \wedge \neg c) \implies \neg r \iff a \vee b \vee c \vee \neg r$$

Tweaking SAT encoding

Idea. *Assigning false first*

Few differences are more likely to cancel out.
⇒ Guess all difference variables *false* first.

Basic idea of differential cryptanalysis:
Assign all difference variables first.

Tweaking SAT encoding

Definition. *Preference variables*

Let Δx be the difference variable of pair (x, x') . We introduce a new Boolean variable x^* called *preference variable*. We add clause

$$x^* = (\Delta x \wedge x)$$

Assume x^* is assigned (Boolean false) first.

Assume Δx is assigned first. Worked well in a previous, non-SAT tool.

SHA-256 testcases

i	A	E	W
-4	A: -----	E: -----	W: -----
-3	A: -----	E: -----	W: -----
-2	A: -----	E: -----	W: -----
-1	A: -----	E: -----	W: -----
0	A: -----	E: -----	W: -----
1	A: -----	E: -----	W: -----
2	A: -----	E: -----	W: -----
3	A: x-----	E: ?????? ????? ????? ????? ????? ??	W: ??? ????? ????? ????? ????? ?????
4	A: -----	E: ?????? ????? ????? ????? ????? ??	W: ??? ????? ????? ????? ????? ?????
5	A: -----	E: ?????? ????? ????? ????? ????? ??	W: ??? ????? ????? ????? ????? ?????
6	A: -----	E: ?????? ????? ????? ????? ????? ??	W: ??? ????? ????? ????? ????? ?????
7	A: -----	E: ?????? ????? ????? ????? ????? ??	W: ??? ????? ????? ????? ????? ?????
8	A: -----	E: -----	W: ??? ????? ????? ????? ????? ?????
9	A: -----	E: -----	W: -----
10	A: -----	E: -----	W: -----
11	A: -----	E: -----	W: ?????? ????? ????? ????? ?????
12	A: -----	E: -----	W: -----
13	A: -----	E: -----	W: -----
14	A: -----	E: -----	W: -----
15	A: -----	E: -----	W: -----
16	A: -----	E: -----	W: -----
17	A: -----	E: -----	W: -----

Testcase 18 [05]

SHA-256 testcases

i	A	E	W
-4	A: -----	E: -----	W: -----
-3	A: -----	E: -----	W: -----
-2	A: -----	E: -----	W: -----
-1	A: -----	E: -----	W: -----
0	A: -----	E: -----	W: -----
1	A: -----	E: -----	W: -----
2	A: -----	E: -----	W: -----
3	A: -----	E: -----	W: -----
4	A: -----	E: -----	W: -----
5	A: x????????????????????????????????	E: ?????? ?????? ?????? ?????? ?????? ??	W: ??? ?????? ?????? ?????? ?????? ??????
6	A: -----	E: ?????? ?????? ?????? ?????? ?????? ??	W: ??? ?????? ?????? ?????? ?????? ??????
7	A: -----	E: ?????? ?????? ?????? ?????? ?????? ??	W: ??? ?????? ?????? ?????? ?????? ??????
8	A: -----	E: ?????? ?????? ?????? ?????? ?????? ??	W: ??? ?????? ?????? ?????? ?????? ??????
9	A: -----	E: ?????? ?????? ?????? ?????? ?????? ??	W: -----
10	A: -----	E: -----	W: -----
11	A: -----	E: -----	W: -----
12	A: -----	E: -----	W: -----
13	A: -----	E: -----	W: ??? ?????? ?????? ?????? ?????? ??????
14	A: -----	E: -----	W: -----
15	A: -----	E: -----	W: -----
16	A: -----	E: -----	W: -----
17	A: -----	E: -----	W: -----
18	A: -----	E: -----	W: -----
19	A: -----	E: -----	W: -----
20	A: -----	E: -----	W: -----

Testcase 21 [05]

SHA-256 testcases

i	A	E	W
-4	A: -----	E: -----	W: -----
-3	A: -----	E: -----	W: -----
-2	A: -----	E: -----	W: -----
-1	A: -----	E: -----	W: -----
0	A: -----	E: -----	W: -----
1	A: -----	E: -----	W: -----
2	A: -----	E: -----	W: -----
3	A: -----	E: -----	W: -----
4	A: -----	E: -----	W: -----
5	A: -----	E: -----	W: -----
6	A: -----	E: -----	W: -----
7	A: x????????????????????????????????	E: ?????? ?????? ?????? ?????? ?????? ??	W: ??? ?????? ?????? ?????? ?????? ??????
8	A: -----	E: ?????? ?????? ?????? ?????? ?????? ??	W: ??? ?????? ?????? ?????? ?????? ??????
9	A: -----	E: ?????? ?????? ?????? ?????? ?????? ??	W: ??? ?????? ?????? ?????? ?????? ??????
10	A: -----	E: ?????? ?????? ?????? ?????? ?????? ??	W: ??? ?????? ?????? ?????? ?????? ??????
11	A: -----	E: ?????? ?????? ?????? ?????? ?????? ??	W: -----
12	A: -----	E: -----	W: -----
13	A: -----	E: -----	W: -----
14	A: -----	E: -----	W: -----
15	A: -----	E: -----	W: ?????? ?????? ?????? ?????? ?????? ??????
16	A: -----	E: -----	W: -----
17	A: -----	E: -----	W: -----
18	A: -----	E: -----	W: -----
19	A: -----	E: -----	W: -----
20	A: -----	E: -----	W: -----
21	A: -----	E: -----	W: -----
22	A: -----	E: -----	W: -----

Testcase 23 [05]

SHA-256 testcases

i	A	E	W
-4	A: -----	E: -----	W: -----
-3	A: -----	E: -----	W: -----
-2	A: -----	E: -----	W: -----
-1	A: -----	E: -----	W: -----
0	A: -----	E: -----	W: -----
1	A: -----	E: -----	W: -----
2	A: -----	E: -----	W: -----
3	A: -----	E: -----	W: -----
4	A: -----	E: -----	W: -----
5	A: -----	E: -----	W: -----
6	A: -----	E: -----	W: -----
7	A: x???????? ?????????? ?????????? ??????????	E: ?????????? ?????????? ?????????? ??????????	W: ??? ?????????? ?????????? ?????????? ??????????
8	A: -----	E: ?????????? ?????????? ?????????? ??????????	W: ??? ?????????? ?????????? ?????????? ??????????
9	A: -----	E: ?????????? ?????????? ?????????? ??????????	W: ?????????? ?????????? ?????????? ??????????
10	A: -----	E: ?????????? ?????????? ?????????? ??????????	W: ?????????? ?????????? ?????????? ??????????
11	A: -----	E: ?????????? ?????????? ?????????? ??????????	W: -----
12	A: -----	E: -----	W: -----
13	A: -----	E: -----	W: -----
14	A: -----	E: -----	W: -----
15	A: -----	E: -----	W: ?????????? ?????????? ?????????? ??????????
16	A: -----	E: -----	W: -----
17	A: -----	E: -----	W: -----
18	A: -----	E: -----	W: -----
19	A: -----	E: -----	W: -----
20	A: -----	E: -----	W: -----
21	A: -----	E: -----	W: -----
22	A: -----	E: -----	W: -----
23	A: -----	E: -----	W: -----

Testcase 24 [05]

SHA-256 results

A differential description encoding improves the runtime compared to a missing differential description.

testcase	CryptoMiniSat 5		lingeling-ats1	
	w/o dd	w/ dd	w/o dd	w/ dd
MD4, C	1,064	231	798	53
SHA-256, 18	37	37	31	160
SHA-256, 21	T	7,855	28,621	5,513
SHA-256, 23	T	26,212	76,196	1,450
SHA-256, 24	T	37,194	78,017	1,235

SHA-256 results

Lingeling option `--phase=-1` improves its runtime for our testcases.

Option `--phase=-1` of lingeling is described as “default phase” set to `-1` (negative), `0` (default, Jeroslow-Wang strategy [06]) or `1` (positive).

testcase	18		21		23		24	
phase	0	-1	0	-1	0	-1	0	-1
runtime	31	22	28,621	19,717	76,196	71,677	85,774	70,259

SHA-256 results

Evaluating difference variables false first improves the runtime.

testcase	C	18	21	23	24
basic approach (ats1)	798	31	28,621	76,196	85,774
diff-first-false (ats1o1)	652	29	27,599	59,312	66,052

SHA-256 results

Adding preference variables dramatically worsens performance.

testcase	A	B	C
CNF with diff-desc	11	133	155
pref variables added	8	50	62

SHA-256 results

Adding preference variables dramatically worsens performance.

testcase	A	B	C	18	21	23	24
CNF with diff-desc	11	133	155	49	2,282	1,314	2,632
pref variables added	8	50	62	T	T	T	T

Conclusion

- 1 very successful tweak (diff-desc)
- 2 promising tweaks
- 1 insignificant tweak (CNF simplification)
- 1 worsening tweak (preference variables)

We found full-round hash collisions in MD4.

We found a hash collision
in 24 rounds-reduced SHA-256.

References

- 01 Christophe De Cannière and Christian Rechberger.
“Finding SHA-1 Characteristics: General Results and Applications”
- 02 Florian Mendel, Tomislav Nad, Martin Schläffer.
“Improving Local Collisions: New Attacks on Reduced SHA-256”
- 03 Noboru Kunihiro, Kazuo Ohta, Yu Sasaki and Lei Wang.
“New message difference for MD4” (MD4 testcases)
- 04 Ilya Mironov and Lintao Zhang.
“Applications of SAT solvers to cryptanalysis of hash functions”
- 05 Alex Biryukov and Ivica Nikolić.
“Collisions for step-reduced SHA-256” (SHA-256 testcases)
- 06 Robert G. Jeroslow and Jinchang Wang.
“Solving Propositional Satisfiability Problems”

Silly walk image Leo Antunes **Knight image** Springfield Punx

どうもありがとうございました

Acknowledgements: Florian Mendel,
Maria Eichlseder, Armin Biere, Roderick
Bloem, Mate Soos, Martina, my parents
and many others . . .

All resources available online at:

<http://lukas-prokop.at/proj/megosat>

