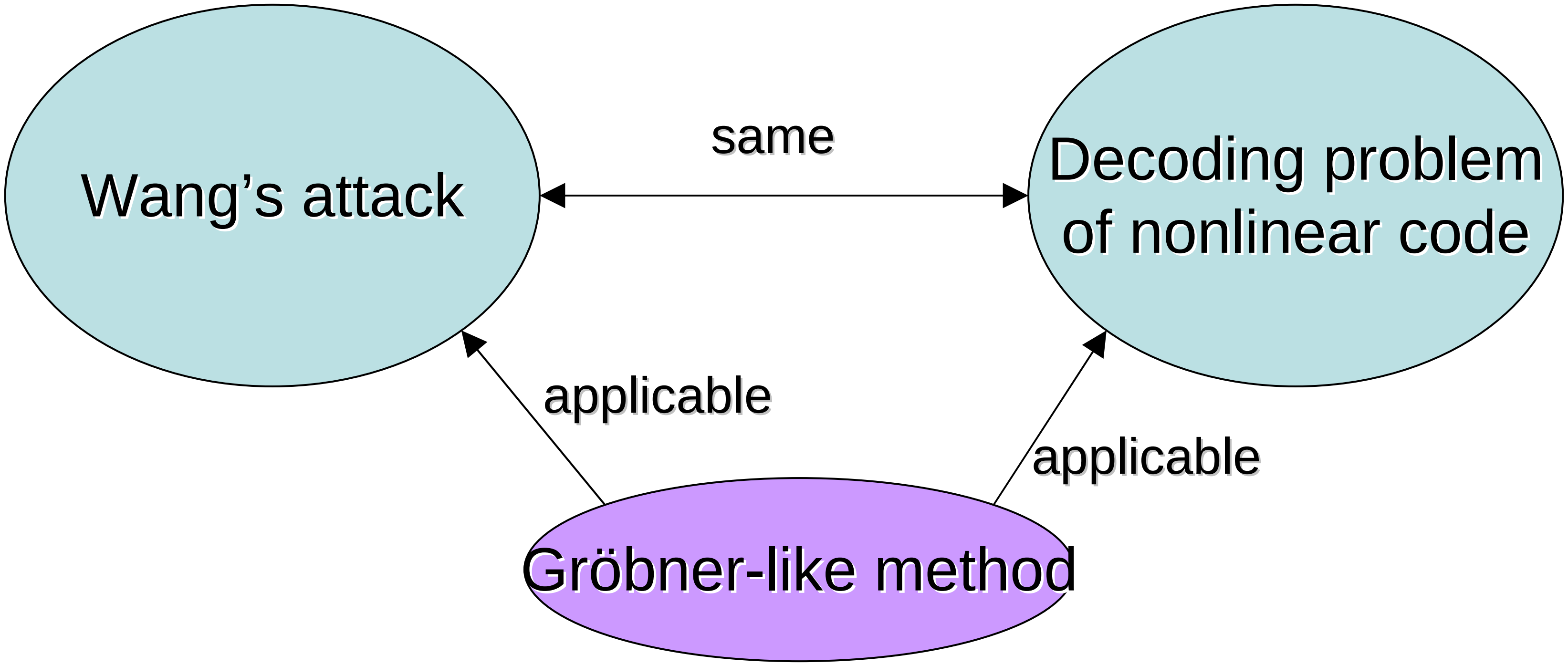
Gröbner Base Based Cryptanalysis of SHA-1

Makoto Sugita
IPA Security Center
Joint work with Mitsuru Kawazoe (Osaka Prefecture university) and Hideki Imai (Chuo University and RCIS, AIST)

Wang's attack, nonlinear code and Gröbner basis



- Wang's attack can be considered as decoding problem of nonlinear code.

Wang's attack

Outline of the attack.

- Find differential paths – characteristics (difference for subtractions modular 2^{32})
- Determine certain sufficient conditions
- For randomly chosen M, apply the message modification techniques
- However, not all information is published
  - How to find such differential path (disturbance vector)?
    - Candidates are too many
  - How to determine sufficient conditions?
  - What is multi-message modification?
    - Details are unpublished

Many details are not public!!

1. How to find the differentials?
2. How to determine sufficient conditions on a_i ?
3. What are the details of message modification technique?

=>

We have clarified 2 and 3, and partially 1

Our Contribution:

- Developing the searching method for ‘good’ message differentials
- Developing the method to determine sufficient conditions
- Developing new multi-message modification technique
  - Proposal of a novel message modification technique employing the Gröbner base based method

Wang's attack and nonlinear code

- Wang's attack is decoding a nonlinear code $\{a_i, m_i\}$ in $GF(2)^{32x80x2}$.
  - Satisfying sufficient conditions
  - Satisfying nonlinear relations between *a* and *m*

$$m_i = (m_{i-3} \oplus m_{i-8} \oplus m_{i-14} \oplus m_{i-16}) \lll 1$$

for $i = 16, \cdots, 79$, where $x \lll n$ denotes n-bit left rotation of x. Using expanded messages, for $i = 1, 2, \cdots, 80$,

$$\begin{aligned}
a_i &= (a_{i-1} \lll 5) + f_i(b_{i-1}, c_{i-1}, d_{i-1}) + e_{i-1} + m_{i-1} + k_i \\
b_i &= a_{i-1} \\
c_i &= b_{i-1} \lll 30 \\
d_i &= c_{i-1} \\
e_i &= d_{i-1}
\end{aligned}$$

where initial chaining value $IV = (a_0, b_0, c_0, d_0, e_0)$ is $(0x67452301, 0xefcdab89, 0x98badcfe, 0x10325476, 0xc3d2e1f0)$.

How to decode nonlinear code?

- A general method
  - Gröbner bases based algorithm
- Difficult to calculate Gröbner basis directly:
  - System of equations is very complex
- How to decode?
  - Employ Gröbner base based method
  - Employ techniques of error correcting code
  - Note: Nonlinear relations between *a* and *m* can be linearly approximated

Control sequence

- Control sequence represents Gröbner base

Control sequence s_i	Control bit b_i	Controlled relation r_i
s_{120}	$a_{16,31}$	$m_{15,31} = 1$
s_{119}	$a_{16,29}$	$m_{15,29} = 0$
s_{118}	$a_{16,28}$	$m_{15,28} + m_{10,28} + m_{8,29} + m_{7,29} + m_{4,28} + m_{2,28} = 1$
s_{117}	$a_{16,27}$	$m_{15,27} + m_{14,25} + m_{12,28} + m_{12,26} + m_{10,28} + m_{9,27} + m_{9,25} + m_{8,29} + m_{8,28} + m_{7,28} + m_{7,27} + m_{6,26} + m_{5,28} + m_{4,26} + m_{3,25} + m_{2,28} + m_{1,25} + m_{0,28} = 1$
s_{116}	$a_{16,26}$	$m_{15,26} + m_{10,28} + m_{10,26} + m_{8,28} + m_{8,27} + m_{7,27} + m_{6,29} + m_{5,27} + m_{4,26} + m_{2,27} + m_{2,26} + m_{0,27} = 1$
s_{115}	$a_{16,25}$	$m_{15,25} + m_{11,28} + m_{10,27} + m_{10,25} + m_{9,28} + m_{8,27} + m_{8,26} + m_{7,26} + m_{6,29} + m_{6,28} + m_{5,26} + m_{4,25} + m_{3,28} + m_{2,28} + m_{2,26} + m_{2,25} + m_{1,28} + m_{0,28} + m_{0,26} = 0$
s_{114}	$a_{16,24}$	$m_{15,24} + m_{12,28} + m_{11,27} + m_{10,26} + m_{10,24} + m_{9,28} + m_{9,27} + m_{8,29} + m_{8,26} + m_{8,25} + m_{7,25} + m_{6,29} + m_{6,28} + m_{6,27} + m_{5,25} + m_{4,28} + m_{4,24} + m_{3,28} + m_{3,27} + m_{2,27} + m_{2,25} + m_{2,24} + m_{1,28} + m_{1,27} + m_{0,27} + m_{0,25} = 1$
s_{113}	$a_{16,23}$	$m_{15,23} + m_{12,28} + m_{12,27} + m_{11,26} + m_{10,25} + m_{10,23} + m_{9,27} + m_{9,26} + m_{8,28} + m_{8,25} + m_{8,24} + m_{7,29} + m_{7,24} + m_{6,28} + m_{6,27} + m_{6,26} + m_{5,24} + m_{4,27} + m_{4,23} + m_{3,27} + m_{3,26} + m_{2,26} + m_{2,24} + m_{2,23} + m_{1,27} + m_{1,26} + m_{0,26} + m_{0,24} = 1$
s_{112}	$a_{16,22}$	$m_{15,22} + m_{14,25} + m_{12,28} + m_{12,27} + m_{11,25}$

Neutral bit

- Introduced by Biham and Chen
- Some bits do not affect relations
  - Increase the probability of collision

Semi-neutral bit

- We introduce new notion 'Semi-neutral bit'
- Change of some bits can easily be adjusted in a few steps of control sequence
  - Which means that noise on semi-neutral bits can be easily decoded

Sufficient conditions and new message modification techniques

chaining variable	31 - 24	23 - 16	15 - 8	8 - 0
a_0	01100111	01000101	00100011	00000001
a_1	101V--vV	Y-------	--------	-1-a10aa
a_2	01100vVv	------0-	----a---	1-w00010
a_3	0010--Vv	-10---1a	------0-	0aX1a0W0
a_4	11010vv-	-01-----	01aaa---	0W10-100
a_5	10w01aV-	-1-01-aa	--00100-	0w--01W1
a_6	11W-0110	-a-1001-	01100010	1-a111W1
a_7	w1x-1110	a1a1111-	-101-001	1---0-10
a_8	h0Xvvv10	0000000a	a001a1--	100X0-1h
a_9	00XVrr-V	11000100	00000000	101-1-1y
a_{10}	0w1-rv-v	11111011	11100000	00hW0-1h
a_{11}	1w0--V-V	-------1	01111110	11x---0Y
a_{12}	0w1-rV-V	--------	--------	-1XWa-Wh
a_{13}	1w0--vv-	-rr-----	--------	-1-qq01y
a_{14}	1rhhvvVh	hh------	qNNNNNqN	N1hhh1hh
a_{15}	0rwhhhVh	hhhh---N	qNNqqNqN	NNhh0hh0
a_{16}	W1whhhhh	hhqNqNqN	NNqNNqqq	qWWhahhh
a_{17}	-0------	--------	--------	----100-
a_{18}	1-1-----	--------	--------	-----00-
a_{19}	--------	--------	--------	-------0

1, 0, a: Wang's sufficient conditions

w: adjust $a_{i+1,j}$ so that $m_{i,j} = 0$

W: adjust $a_{i+1,j}$ so that $m_{i,j} = 1$

v: adjust $a_{i,j-5}$ so that $m_{i,j} = 0$

V: adjust $a_{i,j-5}$ so that $m_{i,j} = 1$

N: semi-neutral bit

…

Proposal of the method to determine sufficient conditions and new message modification technique using Gröbner basis

New collision example of 58-step SHA-1

M = 0x

1ead6636 319fe59e 4ea7ddcb c7961642 0ad9523a f98f28db 0ad135d0 e4d62aec 6c2da52c 3c7160b6 06ec74b2 b02d545e bdd9e466 3f156319 4f497592 dd1506f93

M' = 0x

ead6636 519fe5ac 2ea7dd88 e7961602 ead95278 998f28d9 8ad135d1 e4d62acc 6c2da52f 7c7160e4 46ec74f2 502d540c 1dd9e466 bf156359 6f497593 fd150699

- Note that the proposed method is the first fully-published method that can cryptanalyze 58-round SHA-1

Cryptanalysis of 58-round SHA-1

- We can achieve all message conditions and 8 chaining value conditions in 17 – 23 round (success probability is 0.5)
- 29 conditions remained
  - > exhaustive search (2^{29} message modification)
- Constant is practical?
  - Utilization of Groebner base based method
  - 2^{29} message modification -> 2^{8} message modification (symbolic computation)
  - However, complexity is exactly same
    - 2^{29} SHA-1 -> 2^{29} SHA-1
  - Complexity can be reduced employing a suitable technique of error correcting code and Groebner basis?

Using Groebner base based method (Algorithm 3)

chaining variable	31 - 24	23 - 16	15 - 8	8 - 0
a_0	01100111	01000101	00100011	00000001
a_1	101V--vV	Y-------	--------	-1-a10aa
a_2	01100vVv	------0-	----a---	1-w00010
a_3	0010--Vv	-10---1a	------0-	0aX1a0W0
a_4	11010vv-	-01-----	01aaa---	0W10-100
a_5	10w01aV-	-1-01-aa	--00100-	0w--01W1
a_6	11W-0110	-a-1001-	01100010	1-a111W1
a_7	w1x-1110	a1a1111-	-101-001	1---0-10
a_8	h0Xvvv10	0000000a	a001a1--	100X0-1h
a_9	00XVrr-V	11000100	00000000	101-1-1y
a_{10}	0w1-rv-v	11111011	11100000	00hW0-1h
a_{11}	1w0--V-V	-------1	01111110	11x---0Y
a_{12}	0w1-rV-V	--------	--------	-1XWa-Wh
a_{13}	1w0--vv-	-rr-----	--------	-1-qq01y
a_{14}	1rhhvvVh	hh------	qNNNNNqN	N1hhh1hh
a_{15}	0rwhhhVh	hhhh---N	qNNqqNqN	NNhh0hh0
a_{16}	W1whhhhh	hhqNqNqN	NNqNNqqq	qWWhahhh
a_{17}	-0------	--------	--------	----100-
a_{18}	1-1-----	--------	--------	-----00-
a_{19}	--------	--------	--------	-------0

Problem to determine semi-neutral bits denoted as ‘N’ is equivalent to calculating Groebner basis from algebraic equations on variable denoted as ‘q’ or ‘N’

Calculation of Groebner basis

A message differential of full SHA-1 slightly different from Wang's (first iteration)

	$\Delta^{\pm} m$	$\Delta^{+} m$	$\Delta^{-} m$
$i = 0$	a0000003	00000001	a0000002
$i = 1$	20000030	20000020	00000010
$i = 2$	60000000	60000000	00000000
$i = 3$	e000002a	40000000	a000002a
$i = 4$	20000043	20000042	00000001
$i = 5$	b0000040	a0000000	10000040
$i = 6$	d0000053	d0000042	00000011
$i = 7$	d0000022	d0000000	00000022
$i = 8$	20000000	00000000	20000000
$i = 9$	60000032	20000030	40000002
$i = 10$	60000043	60000041	00000002
$i = 11$	20000040	00000000	20000040
$i = 12$	e0000042	c0000000	20000042
$i = 13$	60000002	00000002	60000000
$i = 14$	80000001	00000001	80000000
$i = 15$	00000020	00000020	00000000
$i = 16$	00000003	00000002	00000001
$i = 17$	40000052	00000002	40000050
$i = 18$	40000040	00000000	40000040
$i = 19$	e0000052	00000002	e0000050
$i = 20$	a0000000	00000000	a0000000
$i = 21$	80000040	80000000	00000040
$i = 22$	20000001	00000001	20000000
$i = 23$	20000060	00000000	20000060

	$\Delta^{\pm} a$	$\Delta^{+} a$	$\Delta^{-} a$
$i = 0$	00000000	00000000	00000000
$i = 1$	e0000001	a0000000	40000001
$i = 2$	20000004	20000000	00000004
$i = 3$	c07fff84	803fff84	40400000
$i = 4$	800030e2	800010a0	00002042
$i = 5$	084080b0	08008020	00400090
$i = 6$	80003a00	00001a00	80002000
$i = 7$	0fff8001	08000001	07fff8000
$i = 8$	00000008	00000008	00000000
$i = 9$	80000101	80000100	00000001
$i = 10$	00000002	00000002	00000000
$i = 11$	00000100	00000000	00000100
$i = 12$	00000002	00000002	00000000
$i = 13$	00000000	00000000	00000000
$i = 14$	00000000	00000000	00000000
$i = 15$	00000001	00000001	00000000
$i = 16$	00000000	00000000	00000000
$i = 17$	80000002	80000002	00000000
$i = 18$	00000002	00000002	00000000
$i = 19$	80000002	80000002	00000000
$i = 20$	00000000	00000000	00000000
$i = 21$	00000002	00000002	00000000
$i = 22$	00000000	00000000	00000000
$i = 23$	[illegible]	[illegible]	[illegible]

Sufficient conditions for the full SHA-1 (first iteration)

message variable	31 - 24	23 - 16	15 - 8	8 - 0
m_0	1-1-----	--------	--------	------10
m_1	--0-----	--------	--------	--01----
m_2	-00-----	--------	--------	--------
m_3	101-----	--------	--------	--1-1-1-
m_4	--0-----	--------	--------	-0----01
m_5	0-01----	--------	--------	-1------
m_6	00-0----	--------	--------	-0-1--01
m_7	00-0----	--------	--------	--1---1-
m_8	--1-----	--------	--------	--------
m_9	-10-----	--------	--------	--00--1-
m_{10}	-00-----	--------	--------	-0----10
m_{11}	--1-----	--------	--------	-1------
m_{12}	001-----	--------	--------	-1----1-
m_{13}	-11-----	--------	--------	------0-
m_{14}	1-------	--------	--------	-------0
m_{15}	--------	--------	--------	--0-----
m_{16}	--------	--------	--------	------01
m_{17}	-1------	--------	--------	-1-1--0-
m_{18}	-1------	--------	--------	-1------
m_{19}	111-----	--------	--------	-1-1--0-
m_{20}	1-1-----	--------	--------	--------
m_{21}	0-------	--------	--------	-1------
m_{22}	--1-----	--------	--------	-------0
m_{23}	--1-----	--------	--------	-11-----

chaining variable	31 - 24	23 - 16	15 - 8	8 - 0
a_0	01100111	01000101	00100011	00000001
a_1	010----0	-0-01-0-	10-0-10-	---a0101
a_2	-100---1	0aa10a1a	01a1a011	1--a11a1
a_3	01011---	-1000000	00000000	01--a0a1
a_4	0-101--a	---10000	00101000	010---10
a_5	0-0101-1	-1-11110	00111-00	10010100
a_6	1-0a1a0a	a0a1aaa-	--10010-	--01-0--
a_7	--0-0111	11111111	111-010-	0-0-0110
a_8	-10---01	11110000	010-111-	1---000-
a_9	00----11	11111111	111----0	----1-01
a_{10}	-11-----	--------	-----a--	-1--1-0-
a_{11}	100-----	--------	-------1	-1--0---
a_{12}	--------	--------	--------	-1----0-
a_{13}	0-------	--------	--------	-1---0--
a_{14}	1-------	--------	--------	-----1--
a_{15}	--------	--------	--------	----0--0
a_{16}	-1------	--------	--------	----1-A-
a_{17}	00------	--------	--------	----0-0-
a_{18}	1-1-----	--------	--------	----a-0-
a_{19}	0-b-----	--------	--------	------0-
a_{20}	--0-----	--------	--------	----a---
a_{21}	--b-----	--------	--------	------0-
a_{22}	--------	--------	--------	----aa--
a_{23}	--------	--------	--------	------00

Control sequence of full SHA-1 (first iteration)

ctrl. seq.	control bits	controlled relation
s_{168}	$a_{15,8}$	$a_{30,2} + a_{29,2} = 1$
s_{167}	$a_{16,6}$	$a_{26,2} + a_{25,2} = 1$
s_{166}	$a_{15,7}$	$a_{25,3} + a_{24,3} = 0$
s_{165}	$a_{13,7}$	$a_{24,3} + a_{23,3} = 0$
s_{164}	$a_{13,9}$	$a_{23,0} = 0$
s_{163}	$a_{16,10}$	$a_{22,3} + a_{21,3} = 0$
s_{162}	$a_{16,11}$	$a_{21,29} + a_{20,31} = 0$
s_{161}	$a_{16,8}$	$a_{21,1} = 0$
s_{160}	$a_{16,9}$	$a_{20,29} = 0$
s_{159}	$a_{15,10}$	$a_{20,3} + a_{19,3} = 0$
s_{158}	$a_{15,11}$	$a_{19,31} = 0$
s_{157}	$a_{15,9}$	$a_{19,29} + a_{18,31} = 0$
s_{156}	$a_{14,8}$	$a_{19,1} = 0$
s_{155}	$a_{14,11}$	$a_{18,31} = 1$
s_{154}	$a_{15,14}$	$a_{18,29} = 1$
s_{153}	$a_{13,8}$	$a_{18,1} = 0$
s_{152}	$a_{13,11}$	$a_{17,31} = 0$
s_{151}	$a_{13,10}$	$a_{17,30} = 0$
s_{150}	$a_{13,13}$	$a_{17,1} = 0$
s_{149}	$a_{16,31}$	$m_{15,31} = 0$
s_{148}	$a_{16,29}$	$m_{15,29} = 1$
s_{147}	$a_{16,28}$	$m_{15,28} + m_{10,28} + m_{4,28} + m_{2,28} = 0$
s_{146}	$a_{16,27}$	$m_{15,27} + m_{10,27} + m_{8,28} + m_{4,27} + m_{2,28} + m_{2,27} + m_{0,28} = 1$
s_{145}	$a_{16,26}$	$m_{15,26} + m_{10,28} + m_{10,26} + m_{8,28} + m_{8,27} + m_{7,27} + m_{5,27} + m_{4,26} + m_{2,27} + m_{2,26} + m_{0,27} = 0$
s_{144}	$a_{16,25}$	$m_{15,25} + m_{11,28} + m_{10,27} + m_{10,25} + m_{9,28} + m_{8,27} + m_{8,26} + m_{7,26} + m_{5,26} + m_{4,25} + m_{3,28} + m_{2,28} + m_{2,26} + m_{2,25} + m_{1,28} + m_{0,28} + m_{0,26} = 0$
s_{143}	$a_{16,24}$	$m_{15,24} + m_{12,28} + m_{11,27} + m_{10,26} + m_{10,24} + m_{9,28} + m_{9,27} + m_{8,26} + m_{8,25} + m_{7,25} + m_{6,27} + m_{5,25} + m_{4,28} + m_{4,24} + m_{3,28} + m_{3,27} + m_{2,27} + m_{2,25} + m_{2,24} + m_{1,28} + m_{1,27} + m_{0,27} + m_{0,25} = 1$
s_{142}	$a_{16,23}$	$m_{15,23} + m_{12,28} + m_{12,27} + m_{11,26} + m_{10,25} + m_{10,23} + m_{9,27} + m_{9,26} + m_{8,28} + m_{8,25} + m_{8,24} + m_{7,24} + m_{7,0} + m_{6,27} + m_{6,26} + m_{5,24} + m_{4,27} + m_{4,23} + m_{3,27} + m_{3,26} + m_{2,26} + m_{2,24} + m_{2,23} + m_{1,30} + m_{1,27} + m_{1,26} + m_{1,0} + m_{0,26} + m_{0,24} = 0$

Advanced sufficient conditions and semi-neutral bits of full-round SHA-1

message variable	31 - 24	23 - 16	15 - 8	8 - 0
m_0	1-1-----	--------	--------	------10
m_1	L-0-----	--------	--------	--01----
m_2	L00-----	--------	--------	-------L
m_3	101-----	--------	--------	--1-1-1L
m_4	LL0-----	--------	--------	-0----01
m_5	0L01----	--------	--------	-1-----L
m_6	00L0----	--------	--------	-0-1--01
m_7	00-0----	--------	--------	--1L--1-
m_8	L-1-----	--------	--------	----L--L
m_9	L10-----	--------	--------	--00-L1L
m_{10}	L00-----	--------	--------	-0LLLL10
m_{11}	LL1-----	--------	--------	-1LLLLLL
m_{12}	001-----	--------	--------	-1LLL-1L
m_{13}	L11LLLLL	LLLLLLLL	L-L-----	--LLLL0L
m_{14}	1LLLLLLL	LLLLLLLL	L-LL----	--LLLLL0
m_{15}	LLLLLLLL	LLLLLLLL	LL-L----	L-0LLLLL
m_{16}	--------	--------	--------	------01
m_{17}	-1------	--------	--------	-1-1--0-
m_{18}	-1------	--------	--------	-1------
m_{19}	111-----	--------	--------	-1-1--0-
m_{20}	1-1-----	--------	--------	--------
m_{21}	0-------	--------	--------	-1------
m_{22}	--1-----	--------	--------	-------0
m_{23}	--1-----	--------	--------	-11-----
m_{24}	1-------	--------	--------	-------1

chaining variable	31 - 24	23 - 16	15 - 8	8 - 0
a_0	01100111	01000101	00100011	00000001
a_1	010-FrF0	y0-01-0-	10-0-10-	F-Fa0101
a_2	F100-Vv1	0aa10a1a	01a1a011	1-wa11a1
a_3	01011VFV	-1000000	00000000	01FFa0a1
a_4	0w101v-a	y--10000	00101000	010XWF10
a_5	0w0101y1	V1-11110	00111-00	10010100
a_6	1w0a1a0a	a0a1aaa-	--10010F	-W01F0Fh
a_7	ww0w0111	11111111	111-010F	0w0W0110
a_8	w10wvv01	11110000	010-111F	1-Wh000F
a_9	00WV--11	11111111	111----0	---F1F01
a_{10}	W11x-Vvv	--------	-----a--	-1ww1h0w
a_{11}	100V----	--------	-------1	-1hh0hWw
a_{12}	wwWF-v--	--------	--------	-1hhhh0h
a_{13}	0wW--V--	-F-F-F--	FNqNqqqq	q1hhh0WW
a_{14}	1WWhhhhh	hhhhhhhh	hNhNqNNq	NNhhh1wh
a_{15}	WWwhhhhh	hhhhhhhh	hqhhqqqq	qNwh0hh0
a_{16}	w1Whhhhh	hhhhhhhh	hhNhqqqq	hqwh1hAh
a_{17}	00------	--------	--------	----0-0-
a_{18}	1-1-----	--------	--------	----a-0-
a_{19}	0-b-----	--------	--------	------0-
a_{20}	--0-----	--------	--------	----a---
a_{21}	--b-----	--------	--------	------0-
a_{22}	--------	--------	--------	----aa--
a_{23}	--------	--------	--------	------00
a_{24}	-c------	--------	--------	----a---

Cryptanalysis of full-round SHA-1 (first iteration)

- We can achieve all message conditions and all chaining variable conditions in 17 – 26 round
- 64 conditions remained
  - > exhaustive search (2^{64} message modification)
- Constant is practical?
  - Utilization of Groebner base based method
  - 2^{64} message modification -> 2^{51} message modification (symbolic computation)
  - However, total complexity is still same
  - Complexity can be reduced employing a suitable technique of error correcting code and Groebner basis?

Example which satisfies sufficient conditions until 28-th round

M = 0x

aa740c82 9f91e819 84c3e50f a898306b
1e5b4111 1867d96b 0616ea95 014a2f32
7ae92980 d5e4d6c6 9d49d0ba 3b8087d3
32717277 edcec899 dc537498 63bca615

- The above M satisfies all message conditions of 0-80 rounds and all chaining variable conditions of 0-28 rounds

Gröbner cryptanalysis of SHA-1

- Gröbner base based cryptanalysis (simplification of Wang's attack) of SHA-1 can be easily implemented by everyone
  - Everyone can evaluate the complexity accurately
  - Everyone can easily evaluate the immunity of SHA-2 against Gröbner base based attack (or Wang's attack)
  - Everyone can propose new algorithms immune to our attack (or Wang's attack)

(Near) Future Work

- Find the collision of full-round SHA-1
  - Use Gröbner base based cryptanalysis
    - As an improvement of Wang's attack
  - Community of symbolic computation has so many good techniques
  - Wang (probably) does not use such techniques e.g. iterative decoding, list decoding, Sudan algorithm, Groebner basis based method

Question:
Who and when will find the collision of full-round SHA-1?

- My (only personal, not public) conjecture
  - Someone in the crypto community or the community of symbolic computation
  - In a few years, not in 10 years as NIST considers

Future work: Application to SHA-2

- Finding good sufficient conditions
  - Difficult to find?
  - Hint: Sufficient conditions do not need to be linear relations on $\{m_{ij}\}$ or $\{a_{ij}\}$
- Once good sufficient conditions are determined, problems are degenerated into symbolic computation