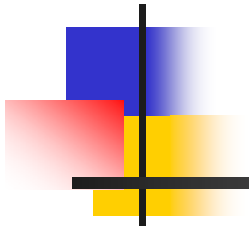
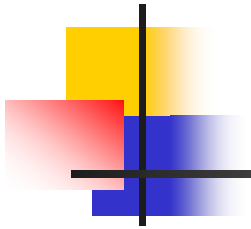


Cryptanalysis on Hash Functions

Xiaoyun Wang
Tsinghua University & Shandong University
05-10-2006



Outline

- Introduction to hash function
- Hash function and signature
- Dedicated hash function
- Modular differential attack
- How to fast find the collision path and efficient attack
- New collision attack on SHA-1
- Some potential dangers for hash functions

Introduction to Hash Function

- **Hash Function: a compress function Y**=H(M) which hash any message with arbitrary length into a fixed length output:

  $H(M)$: $M \in \{0,1\}^* \rightarrow \{0,1\}^l$

- One-way property： Given any Y=H(M), it is infeasible to get any substantial information of M
- Second-Preimage Resistance: Given any message M_1, it is difficult to find another message M_2 such that :

  $H(M_1)=H(M_2)$ 2^l

- Free-Collision: It is difficult to find two different messages (M_1, M_2) with the same hash value:

  $H(M_1)=H(M_2)$ $2^{\frac{l}{2}}$

Hash Function and Signature

H(x): hash function

S(M): signature algorithm

Signing process:

- Compute the fingerprint (or digest) of message:

  M —H→ H(M)

- Signing the fingerprint H(M): s=S(H(M))
- If the fingerprint of M_1 is the same as another different M_2

  $H(M_1)=H(M_2)$

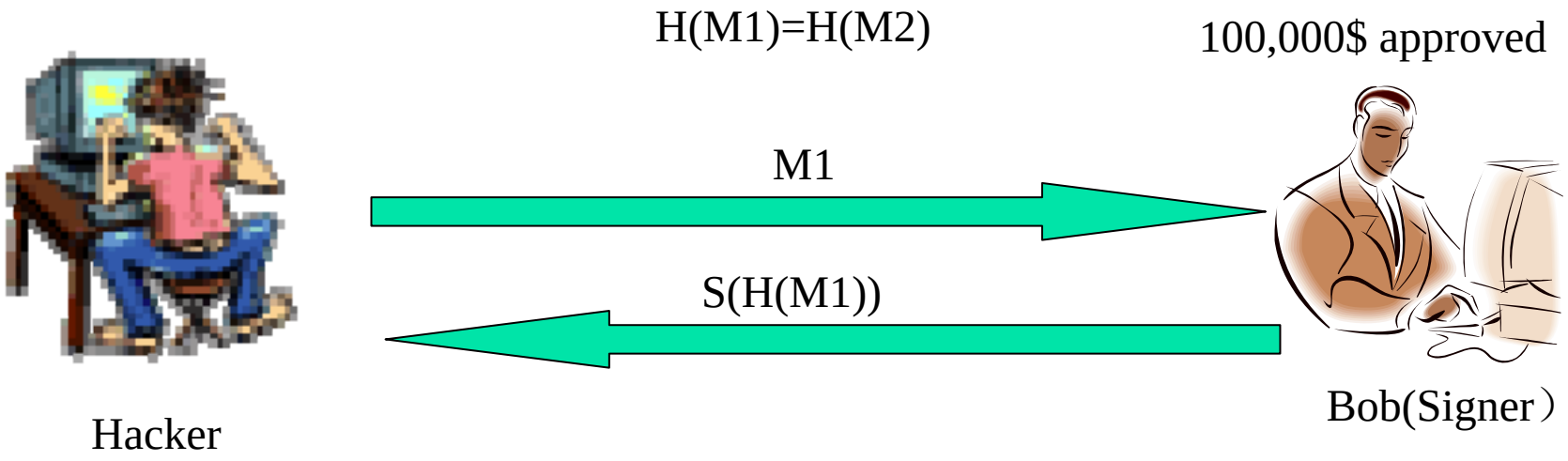
  Then M_1 and M_2 have the same signatures

  $S(H(M_1))=S(H(M_2))$

Fundamental tool to constructing many cryptosystems, especially provable cryptosystems

Hash Function and Signature

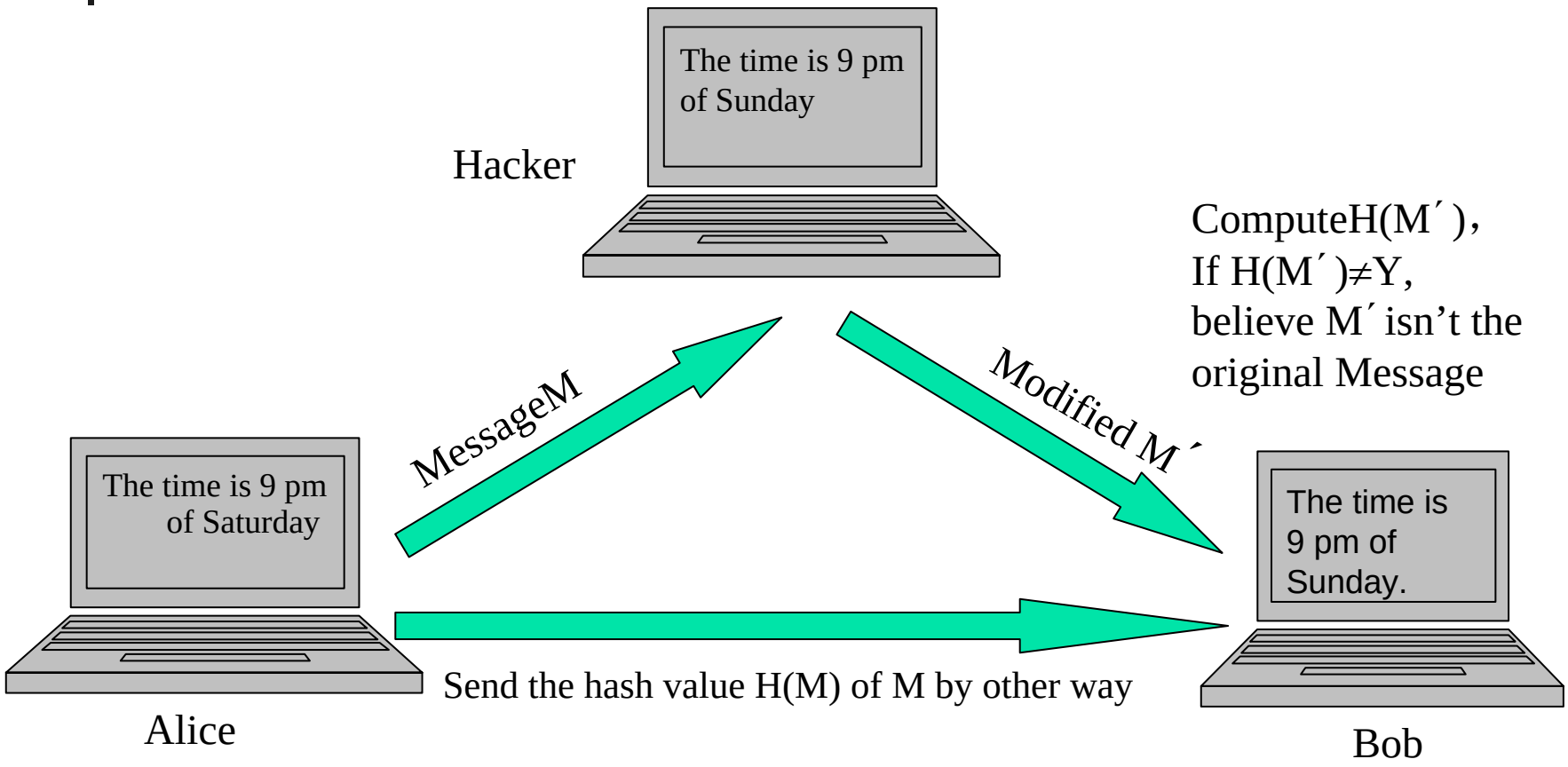
M_1=(project application 1+apllication fund100,000$)
M_2=(project application 2+application fund1000,000$)

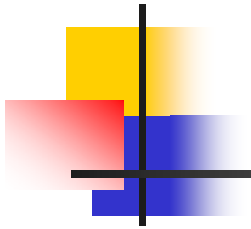


Bob has signed both messages M1 and M2 because of S(H(M1)) ＝S(H(M2))

Hacker prepares two application versions for a project in advance

Hash Function and Data Integrity

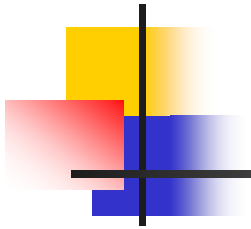




Dedicated (MDx) Hash Functions

- Before 1990: Hash functions based on block ciphers
  Since 1990: Dedicated hash functions (constructed directly)
- Two kinds of dedicated hash functions
- MDx (Rivest)： MD4, MD5, HAVAL, RIPEMD, RIPEMD-160.
- SHAx (NIST)： SHA-0, SHA-1, SHA-2(256, 384, 512)

Earlier Cryptanalysis on Hash Functions



1 1993: Boer and Bosselor found one message with two different sets of initial values.

MSB has weak avalanche

2 1996: Dobbertin found a collision attack on MD4 with probability 2^{-22} (FSE'96)

3 1996: Dobbertin gave a psuodrandom collision example of MD5 which is two messages with another set of initial values (Eucrypt'96: Rump session)

It is possible to control the bit avalanche

4 2003: Rompay etc: collision attack with probability 2^{-29} (Asiacrypt'03)

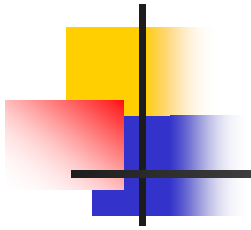
Earlier Cryptanalysis on Hash Functions

- 1997: Wang gave an algebraic method attack to find collision with probability 2^{-58} . The collision path is found by linear algebraic equations, not by computer search

  Circulated in China, wrote in Chinese

- 1998: Chabaud and Joux found a collision attack with probability: 2^{-61}

- 1998: Improved to about 2^{-45} by message modification



The Modular Differential Attack

- XOR differential cryptanalysis:

  Used to analyze block ciphers and hash functions:

  Difference for two variables X and X’ is defined as

  $$\Delta X = X \oplus X'$$

  Uncertainty: For example: $\Delta X = X \oplus X' = 00100000$, for any bit of ΔX, there are two cases. 1st bit is 0: $X_1 = 1$, $X_1' = 1$, or $X_1 = 0$, $X_1' = 0$.

The Modular Differential on Hash functions (2)

- Modular difference and some notations: precise difference with modular subtraction and bit carry

(some suggestions given by Eli Biham and Hans Dobbertin)

1 $\Delta X=X'-X=2^{i-1}$ with bit carries $X'=X[-i,-(i+1),\ldots,(i+t)]$

Sufficient condition: $X_i=1$, $X_{i+1}=1,\ldots X_{(i+t-1}=1$, $X_{i+t}=0$

$$2^{i-1}+ 2^{i-1}+ 2^{i} \ldots..+ 2^{i+t-2}= 2^{i+t-1}$$

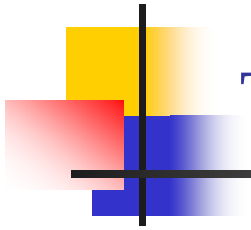
(100..0+111…10=00…01)

2 $\Delta X=X'-X=-2^{i-1}$ with bit carries $X'=X[i,(i+1),\ldots,-(i+t)]$

Sufficient condition: $X_i=0$, $X_{i+1}=0,\ldots X_{(i+t-1)}=0, X_{i+t}=1$

3 For example: $\Delta X=X'-X=-2^5$, $X'=X[-6,-7,-8,9]$

Sufficient conditions: $X_6=0$, $X_7=0$, $X_8=0$, $X_9=1$



The Modular Differential on Hash functions (3)

Iterating Process for Hash Functions:

Merkle-Damgard Meta method

Inputs:

Initial value IV_0

Message $M=(M_0, M_1, \ldots., M_{k-1})$ (padded)

$H_i=F(H_{i-1}, M_{i-1})$, $H_0=IV_0$, $0 \le i \le k-1$

$Y=H_k=H(IV_0, M)$

where

F is a compress function : $\{0, 1\}^l \rightarrow \{0, 1\}^s$, $l>s$

The Modular Differential on Hash functions (4)

- **A collision differential path**

  $\Delta M_0 \rightarrow (\Delta H_1, \Delta M_1) \rightarrow (\Delta H_2, \Delta M_2) \rightarrow \ldots\ldots\ldots \rightarrow \Delta H = \Delta H_k$

  where $\Delta H = 0$

  $M = (M_0, M_1, \ldots M_{k-1})$, $M' = (M_0', M_1', \ldots M_{k-1}')$

  $\Delta M = (\Delta M_0, \ldots\ldots \Delta M_{k-1})$

- One-time Iteration Differential

  $(\Delta H_i, \Delta M_i) \rightarrow \Delta H_{i+1}$

  $(\Delta H_i, \Delta M_i) \rightarrow \Delta R_1 \rightarrow \Delta R_2 \rightarrow \Delta R_3 \rightarrow \Delta R_4$

- Round Differential

$$\Delta R_{j-1} \xrightarrow{P_{j1}} \Delta X_1 \xrightarrow{P_{j2}} \cdots\cdots \xrightarrow{P_{j16}} \Delta X_{16} = \Delta R_j \qquad P_j \geq \prod_{k=1}^{16} P_{jk}$$

- Collision differential with one iteration

$$\Delta M \longrightarrow \Delta R_1 \longrightarrow \Delta R_2 \longrightarrow \Delta R_3 = \Delta H = 0$$

How to Fast Find the Collision Path and Efficient Attack

- **Bit tracing method**
  Bit–tracing method is the better method to fast find the collision path for most existing hash functions
- **Exact conditions for the collision path**
- **Message modification** to correct the wrong conditions in the first round and some conditions in the second round to improve the collision probability.
- **Convert impossible differential to possible differential**

  SHA-Family The good choice for near collision path (by Eli Biham, the same technique used in MD5 collision attack at the same time) has a impossible differential in the first round

  How to convert the impossible differential into another possible path.

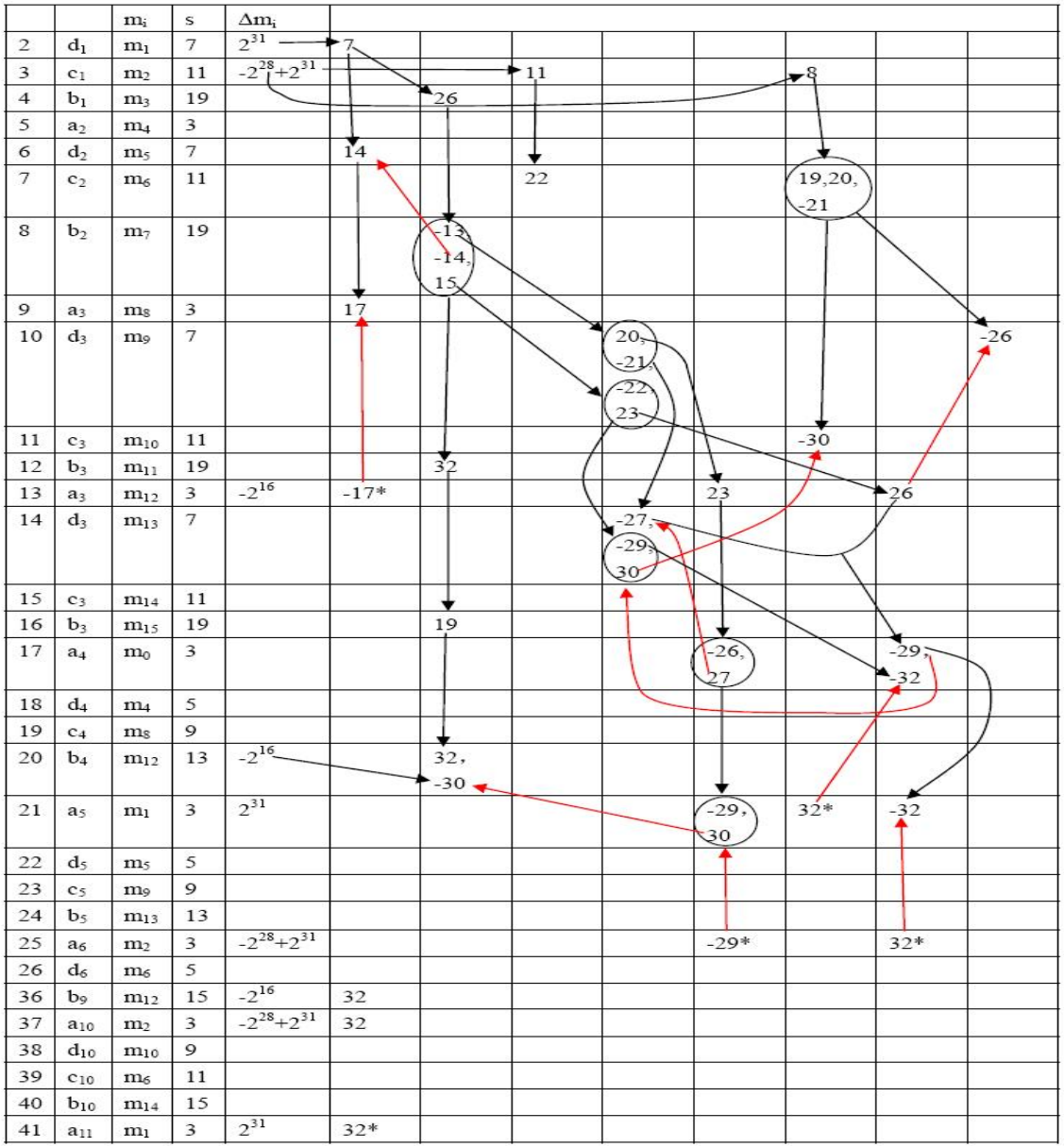
Fast Find the Collision Path---Bit Tracing Method

Bit tracing method

Find an optimized collision path by combining with the properties of nonlinear functions, tracing the possible changed bit locations by the shift operation, producing some useful bits by bit carries to cancel the avalanche.

Usually, it is easy to find short modular differentials, different short modular differentials with no carry bit not only are difficult to cancel with each other, but also increase the new avalanche.

Bit Tracing to Find the Collision Path for MD4



The Collision Path is Composed of Local Collisions ----Local Collision Example

Step	Chaining value for M	$w_{j,i}$	Shift	Δm_i	The i-th step difference	The i-th output for M'	
36	b_9	m_{12}	15	-2^{16}	2^{31}	$b_9[-32]$	$b_{9,32} = 1$
37	a_{10}	m_2	3	$-2^{28} + 2^{31}$	2^{31}	$a_{10}[-32]$	$a_{10,32} = 1$
38	d_{10}	m_{10}	9			d_{10}	
39	c_{10}	m_6	11			c_{10}	
40	b_{10}	m_{14}	15			b_{10}	
41	a_{11}	m_1	3	2^{31}		a_{11}	

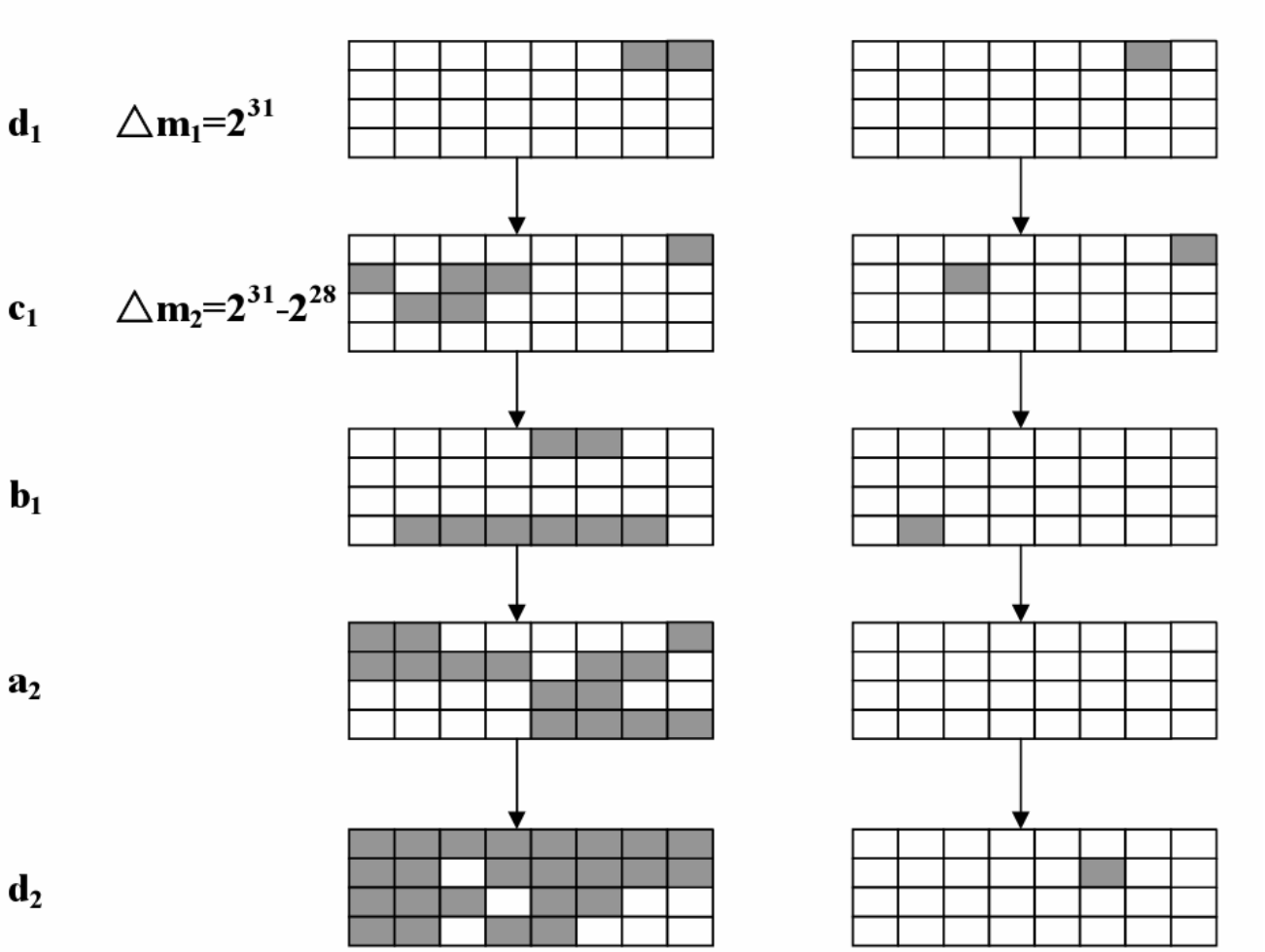
Sufficient Conditions for Collision Path

a_1-b_1	$b_{1,26} = c_{1,26}$
a_2-b_2	$a_{2,26} = 0,\ d_{2,26} = 0,\ c_{2,26} = 1,\ b_{2,29} = c_{2,29},\ b_{2,30} = c_{2,30}$
a_3-d_3	$a_{3,29} = 1,\ a_{3,30} = 0,\ d_{3,8} = a_{3,8},\ d_{3,29} = 1,\ d_{3,30} = 0$
c_3-b_3	$c_{3,8} = 1,\ c_{3,29} = 1,\ c_{3,30} = 1,\ b_{3,8} = 0,\ b_{3,32} = c_{3,32}$
a_4-d_4	$a_{4,8} = 1,\ a_{4,32} = 0,\ d_{4,19} = a_{4,19},\ d_{4,32} = 0$
c_4-b_4	$c_{4,19} = 1,\ c_{4,32} = 1,\ b_{4,3} = c_{4,3} + 1,\ b_{4,19} = d_{4,19}$
a_5	$a_{5,3} = 0,\ a_{5,8} = b_{4,8},\ a_{5,19} = b_{4,19},\ a_{5,28} = b_{4,28}$
d_5	$d_{5,3} = b_{4,3},\ d_{5,8} = 0,\ d_{5,28} = 0$
c_5	$c_{5,3} = d_{5,3},\ c_{5,8} = a_{5,8},\ c_{5,28} = 1$
b_5	$b_{5,6} = c_{5,6},\ b_{5,8} = c_{5,8},$
a_6	$a_{6,6} = 0,\ a_{6,13} = b_{5,13},\ a_{6,28} = b_{5,28} + 1$
d_6	$d_{6,5} = a_{6,5},\ d_{6,6} = b_{5,6},\ d_{6,13} = 0$
c_6	$c_{6,5} = 0,\ c_{6,6} = 1,\ c_{6,13} = a_{6,13}$
b_6	$b_{6,5} = d_{6,5},\ b_{6,6} = d_{6,6} + 1,\ b_{6,13} = c_{6,13}$
a_7-d_7	$a_{7,5} = b_{6,5},\ a_{7,6} = b_{6,6},\ a_{7,18} = b_{6,18},\ d_{7,14} = a_{7,14},\ d_{7,18} = 0$
c_7-b_7	$c_{7,14} = 1, c_{7,18} = a_{7,18},\ b_{7,14} = d_{7,14}, b_{7,18} = c_{7,18}$
a_8-a_9	$a_{8,14} = b_{7,14},\ a_{8,23} = b_{7,23},\ d_{8,23} = 0,\ c_{8,23} = 1,\ a_{9,23} = b_{8,23}$

How to control the bit avalanche

---the left is a kind of bit avalanche

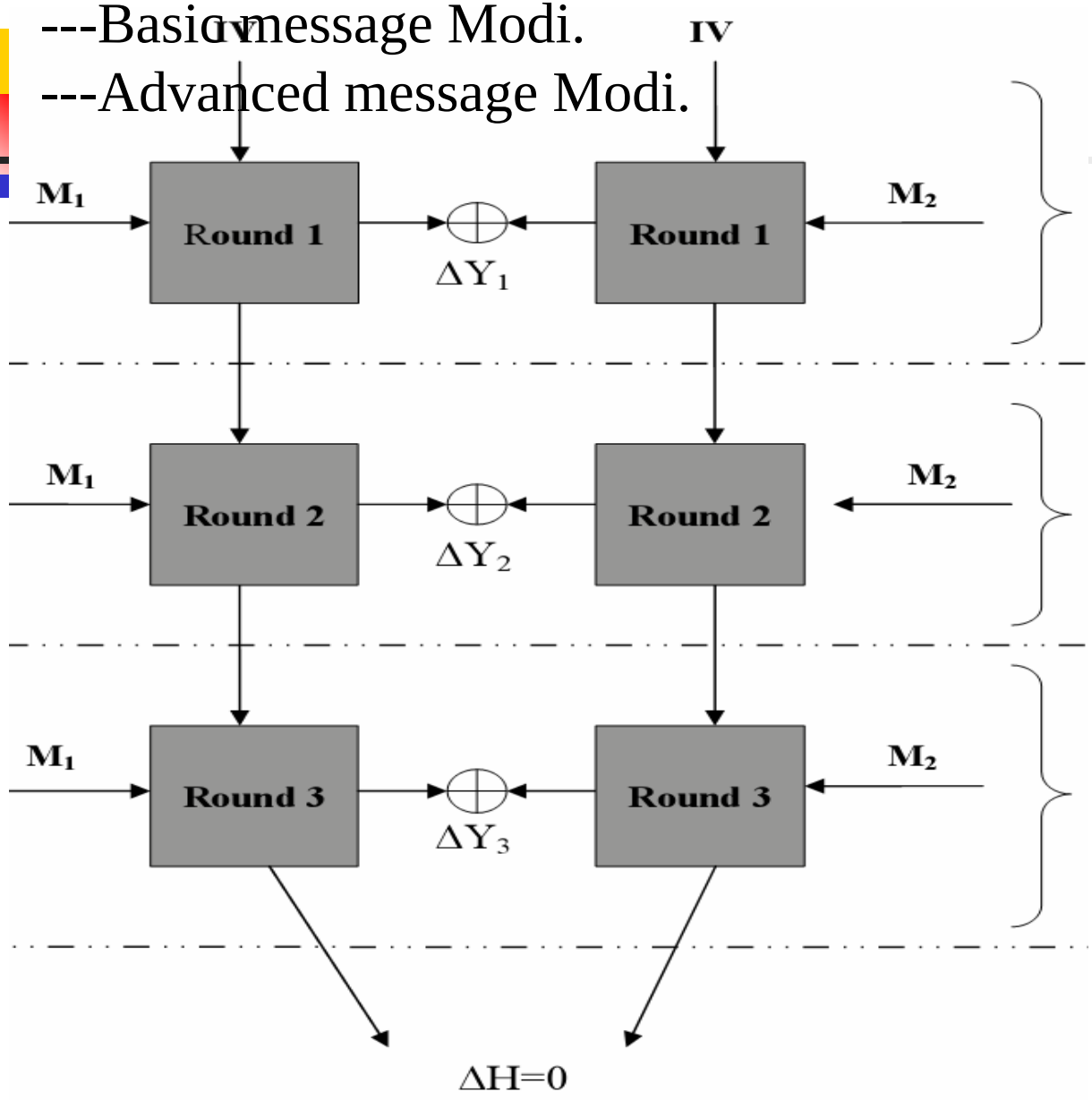
---the right is controlled under the sufficient conditions



Message Modification

---Basic message Modi.

---Advanced message Modi.



The differential holds with probability 1 after basic message modification

The differential holds with high probability after advanced message modification

The differential is selected in advance with high probability

An Example for One Condition Correction for SHA-1

step	Δw_i	Additional Cons	Control bits	Closest Cons	Pr_1	Pr_2
11	2^{11}	$a_{11,12} = m_{10,12}$	$a_{11,12}$	$a_{11,30}$	$\frac{1}{2^{18}}$	
12	2^{16}	$m_{11,17} = 1 + m_{10,12}$				
13		$c_{12,12} = d_{12,12}$		$a_{13,32}$		
14		$b_{13,10} = 0$		$a_{14,32}$		
15		$b_{14,10} = 1$		$a_{15,1}$		
16	$\mathbf{2^9}$	$m_{15,10} = 1 + m_{10,12}$		$a_{16,31}$		
...	...	...	...	...	...	...
19	$\mathbf{2^{10}}$, 2^{12}		$\mathbf{a_{19,11}}$, $a_{19,13}$	$a_{19,32}$	$\frac{1}{2^{19}}$	
20	2^{17}		$\mathbf{a_{20,16}}$, $a_{20,18}$	$a_{20,4}$	$\frac{1}{2^{20}}$	
21			$a_{21,11}$, $a_{21,13}$, $\mathbf{a_{21,21}}$, $a_{21,23}$	$a_{21,30}$	$\frac{1}{2^9}$	
22	2^{12}, 2^{13}		$a_{22,9}$, ..., $a_{22,18}$, $\mathbf{a_{22,26}}$, $a_{22,28}$	$a_{22,3}$	$\frac{1}{2^9}$	
23	2^{18}		$a_{23,1}$, ..., $a_{23,23}$, $\mathbf{a_{23,31}}$	$a_{23,1}$		
24			$\mathbf{a_{24,4}}$ $a_{24,6}$, $a_{24,10}$,....., $a_{24,28}$	$a_{24,31}$		$\frac{1}{2^8}$

Collision Attack on SHA-1

---Obstacles for Further Improvement on SHA-1 Attack

- 2005. 2 Collision attack by Wang, Yin, Yu: 2^{69}

2005. 7-8: Collision path by Xiaoyun Wang, Andrew C. Yao and Frances Yao

- Unlike SHA-0 and MD5, too many message conditions and chaining variable conditions must co-exist in each step of differential path

	$m_{6,1}=1,\ m_{6,2}=0,\ m_{6,5}=1,\ m_{6,7}=0,\ m_{6,29}=0,\ m_{6,31}=0,\ m_{6,32}=0$
	$a_{7,1}=0,\ a_{7,3}=1,\ a_{7,4}=0,\ a_{7,6}=0,\ a_{7,7}=0,\ a_{7,9}=0,\ a_{7,10}=1$ $a_{7,12}=0,\ a_{7,16}=1,\ a_{7,17}=1,\ a_{7,18}=1,\ a_{7,19}=1,\ a_{7,20}=1,\ a_{7,21}=1,\ a_{7,22}=1$ $a_{7,23}=1,\ a_{7,24}=1,\ a_{7,25}=1,\ a_{7,26}=1,\ a_{7,27}=1,\ a_{7,28}=0,\ a_{7,30}=0$
m_6	$m_{23,7}=m_{22,1},\ m_{23,6}=m_{23,7}+1,\ m_{23,30}=m_{19,5},\ m_{25,7}=m_{24,1}+1,\ m_{27,6}=m_{26,1}+1,$ $m_{27,31}=1+m_{22,1},\ m_{29,7}=m_{28,2}+1,\ m_{30,7}=m_{29,2}+1,\ m_{31,6}=m_{30,1}+1,\ m_{31,31}=m_{26,1}+1$ $m_{34,7}=m_{33,2}+1,\ m_{34,2}=m_{34,1}+1,\ m_{35,6}=m_{35,7}+1,\ m_{35,7}=m_{34,2}+1,\ m_{35,31}=m_{30,1}+1$ $m_{37,7}=m_{36,1}+1,\ m_{38,7}=m_{37,2}+1,\ m_{39,31}=m_{34,2}+1,\ m_{41,7}=m_{40,2}+1,\ m_{42,2}=m_{40,2}+1$ $m_{45,7}=m_{44,2}+1,\ m_{47,7}=m_{44,2}+1,\ m_{49,7}=m_{44,2}+1,\ m_{51,7}=m_{44,2}+1,\ m_{52,2}=m_{44,2}+1$ $m_{67,8}=m_{66,3}+1,\ m_{70,9}=m_{69,4}+1,\ m_{71,1}=m_{66,3}+1,\ m_{73,10}=m_{72,5}+1,\ m_{74,2}=m_{69,4}+1$ $m_{75,9}=m_{74,4}+1,\ m_{76,11}=m_{75,6}+1,\ m_{77,3}=m_{72,5}+1,\ m_{79,12}=m_{78,7}+1,\ m_{79,2}=m_{74,4}+1$

Obstacles for Further Improvement on SHA-1 Attack (continued)

- Difficult, because message space available is tight:

-- 50 message conditions in steps 17-80

-- hence 50 message conditions in steps 12-16

-- resulting in 50 message bit equations

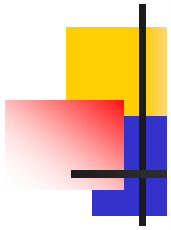
-- most message bits are involved

$$
\begin{aligned}
m_{13,29} &= m_{0,2} + m_{0,24} + m_{0,25} + m_{0,28} + m_{0,29} + m_{0,30} + m_{1,0} + m_{1,3} + m_{1,26} + m_{1,27} + m_{1,28} + m_{1,29} \\
&+ m_{1,30} + m_{2,0} + m_{2,2} + m_{2,3} + m_{2,24} + m_{2,25} + m_{2,29} + m_{2,30} + m_{2,31} + m_{3,2} + m_{3,3} + m_{3,4} + m_{3,25} + m_{3,27} \\
&+ m_{3,28} + m_{3,31} + m_{4,2} + m_{4,3} + m_{4,4} + m_{4,28} + m_{4,30} + m_{4,31} + m_{5,0} + m_{5,3} + m_{5,25} + m_{5,26} + m_{5,29} + m_{5,31} + m_{6,0} \\
&+ m_{6,3} + m_{6,26} + m_{6,27} + m_{7,1} + m_{7,4} + m_{7,28} + m_{7,29} + m_{8,2} + m_{8,3} + m_{8,24} + m_{8,25} + m_{8,26} + m_{8,27} + m_{8,28} + m_{8,29} \\
&+ m_{8,31} + m_{9,0} + m_{9,1} + m_{9,2} + m_{9,3} + m_{9,4} + m_{9,26} + m_{9,28} + m_{9,31} + m_{10,1} + m_{10,2} + m_{10,3} + m_{10,5} + m_{10,28} + m_{10,29} \\
&+ m_{11,0} + m_{11,2} + m_{11,3} + m_{11,25} + m_{11,26} + m_{11,27} \\
&+ m_{11,28} + m_{11,29} + m_{11,30} + m_{11,31} + m_{12,1} + m_{12,2} + m_{12,5} + m_{12,28} + m_{12,30} + m_{13,0} + m_{13,1} + m_{13,3} + m_{13,24} + m_{13,25} + m_{13,}
\end{aligned}
$$

-- in addition, 51 chaining variable conditions in steps 10-16

-- extra chaining variable conditions and message conditions coming from the message modification

The Collision Path for SHA-1
(Presented in Crypt05 and NIST)



i	x_{i-1}	Δm_{i-1}	Δa_i	Δb_i	Δc_i	Δd_i	Δe_i
1	80000001	1,-2 -30,-32	32,-1 30,-31				
2		-5,6 30	-3,30	32,-1 30,-31			
3	40000001	30,31	-31,32 3, 8,9,..,-23	-3,30	30,-31 28,-29		
4	2	-2,-4,-6 -30,31,-32	-2, 6,-7 8,13,-14, 32	-31,32 3,8,9,.-23	-1, 28	30,-31 28,-29	
5	2	-1,2,7,30	5,-6 8,-9, -23, 28	-2,6,-7 8,13,-14, 32	-29,30 1,6,7,..-21	-1,28	30,-31 28,-29
6	80000002	-7 29,-30,-32	-32 -11,12	5,-6 8,-9,-23,28	-32, 4,-5 6,11,-12, 30	-29,30 1,6,7,..-21	-1,28
7	1	-1,2,-5,7 29,31,32	1 -16,-27,28	-32 -11,12	3,-4 6,-7, -21,26	-32, 4,-5 6,11,-12, 30	-29,30 1,6,7,..-21
8		-2,6 29, 31,32	4	1 -16,-27,28	-30 -9,10	3,-4 6,-7, -21,26	-32,4,-5 6,11,-12, 30
9	80000001	-30	32,1 9,-10	4	31 -14,-25,26	-30 -9,10	3,-4 6,-7, -21,26
10	2	-2,5,6 30,-31	2	32,1 9,-10	2	31 -14,-25,26	-30 -9,10
11	2	1,-2,-7 30,31	9,-10	2	30,31 7,-8	2	31 -7 -14,-25,26
12	2	7,-30	2	9,-10	32	30,31 7,-8	2
13		-2,-7 -30,31,32		2	7,-8	32	30,31 7,-8
14		2,-30,-31			32	7,-8	32
15	1	1,32	1			32	7,-8
16		6		1			32

Table 2 An Sample Solution to 1-10 Steps Differential

M	b67fd432	15fdd1d6	8627ed48	a5fcd96b	83dad005
	172193ca	2132f639	58de2ce	7c7e019a	ccceb003
M'	167fd431	35fdd1e6	e627ed48	45fcd941	a3dad046
	a721938a	f132f66a	d58de2ec	5c7e019a	acceb031
$\triangle M$	a0000003	20000030	60000000	e000002a	20000043
	b0000040	d0000053	d0000022	20000000	60000032

Comparison between New Collision Path and Previous Collision Path

- Comparison:

	Old	New
1. Message conditions	50	42
2. Chaining variable conditions in steps 10-16	51	30
3. Message space in steps 10-16 available for direct modification	2^{47}	2^{55}
4. Message space in steps 10-16 available for searching collision before advanced message modification	2^{123}	2^{151}

Strategies for Message Modification

- Determine which message bits are *possible candidates (control bits)* for modification (Table 3).

- The message modification process *must respect* all chaining variable conditions and message conditions.

  --require adding *extra chaining variable* conditions in steps 1-16 and message conditions.
  Especially Consider the carry effect.

  -- message modification follow certain *topological order* coming from correlations among chaining variable conditions.

42 Message Conditions in Steps 17-80 for SHA-1 First Iteration

0	$m_{17,7} = m_{16,2} + 1,\ m_{17,31} = 1$
1	$m_{18,7} = m_{17,2} + 1,\ m_{18,31} = 0$
2	$m_{19,30} = m_{17,5},\ m_{19,31} = 1$
3	$m_{23,7} = m_{22,1},\ m_{23,6} = m_{23,7} + 1,\ m_{23,30} = m_{19,5}$
6-7	$m_{25,7} = m_{24,1} + 1,\ m_{26,7} = m_{25,2} + 1$
8	$m_{27,6} = m_{26,1} + 1,\ m_{27,31} = 1 + m_{22,1}$
10-12	$m_{29,7} = m_{28,2} + 1,\ m_{30,7} = m_{29,2} + 1,\ m_{31,6} = m_{30,1} + 1$
13-15	$m_{31,31} = m_{26,1} + 1,\ m_{34,7} = m_{33,2} + 1,\ m_{34,2} = m_{34,1} + 1$
16-18	$m_{35,6} = m_{35,7} + 1,\ m_{35,7} = m_{34,2} + 1,\ m_{35,31} = m_{30,1} + 1$
19-21	$m_{37,7} = m_{36,1} + 1,\ m_{38,7} = m_{37,2} + 1,\ m_{39,31} = m_{34,2} + 1$
22-24	$m_{41,7} = m_{40,2} + 1,\ m_{42,2} = m_{40,2} + 1,\ m_{45,7} = m_{44,2} + 1$
25-27	$m_{47,7} = m_{44,2} + 1,\ m_{49,7} = m_{44,2} + 1,\ m_{51,7} = m_{44,2} + 1$
28-30	$m_{52,2} = m_{44,2} + 1,\ m_{67,8} = m_{66,3} + 1,\ m_{70,9} = m_{69,4} + 1$
31-33	$m_{71,1} = m_{66,3} + 1,\ m_{73,10} = m_{72,5} + 1,\ m_{74,2} = m_{69,4} + 1$
34-36	$m_{75,9} = m_{74,4} + 1,\ m_{76,11} = m_{75,6} + 1,\ m_{77,3} = m_{72,5} + 1$
37-38	$m_{79,12} = m_{78,7} + 1,\ m_{79,2} = m_{74,4} + 1$

Details for Message Modification —

Available Message Bits to Correct Sufficient Conditions

0	7 1	5 4	15 4	3 2	10 5	0* 3	0* 0*	0* 4	0* 11	0 15	0 14	1 9	0 15	0* 14	0 15	0 7
1	6 0	12 0	12 2	13 0	4 7	3 5	0* 1	0 7	0 5	0 5	0 9	0 10	0 12	3 12	0 16	1 13
2	11 1	8 5	19 4	10 0	12 7	3 3	0 7	0 6	0 12	0 15	0 14	1 13	0 14	0 12	0 19	4 14
3	14 0	13 4	14 2	18 3	10 7	3 6	0 1	0 5	0 8	0 14	0 14	0 13	1 15	3 14	0 15	1 9
4	11 2	11 3	14 0	7 1	4 2	6 4	0 5	0 4	0 9	0 5	0* 8	0 15	0 16	0 11	4 13	4 13
5	14 2	9 5	11 5	10 3	6 0	0* 7	0 1	0* 5	0* 7	0* 10	0 11	0 12	1 12	0* 17	0* 15	0* 14
6	10 2	6 0	10 3	10 3	14 4	3 4	0 5	0* 7	0 8	0 4	0* 15	0 18	0* 12	1 14	4 21	0 15
7	8 2	13 2	14 3	16 0	10 2	3 3	1 8	0* 4	0 1	0 8	0 5	0 7	0* 14	0 12	0 13	3 10
8	11 1	9 1	16 5	12 5	1 5	0* 4	0* 8	0* 5	0 12	0 16	0* 16	1 13	0* 22	1 15	0 7	0 12
9	13 0*	19 1	14 2	8 1	13 7	7 0*	1 8	0 8	0* 7	0* 8	0 13	0 13	0* 14	3 10	1 15	2 14
10	8 4	17 6	10 2	14 3	6 2	4 1	0 6	0 6	0 4	0 8	0* 10	0* 7	0* 13	0* 19	0* 17	4 17
11	12 0*	7 4	16 1	19 3	9 5	1 7	0 3	0 5	0* 7	0* 15	0* 14	0* 15	1 11	0* 19	1 16	0* 6
12	8 2	8 0*	10 2	11 1	4 1	4 4	2 0*	0 4	0* 9	0* 5	0* 9	0* 12	0* 14	0* 12	4 15	1 17
13	15 1	10 2	11 4	9 0*	9 6	1 1	0* 1	0* 6	0* 13	0* 11	0* 11	1 7	0* 10	0* 12	0* 11	0* 14
14	5 0*	9 0*	5 3	8 0*	8 6	9 4	0* 6	0* 8	0* 4	0* 2	0* 9	0* 10	0* 5	4 9	0* 9	2 9
15	6 1	4 3	1 0*	6 1	6 2	2 1	1 1	1 0*	0* 1	0* 1	0* 1	0* 1	0* 1	0* 1	0* 1	3 1

Details for Message Modification — Control bit and Control path

- Choices for control bit: a message bit m_{i',j'} (i'<16) which does not appear explicitly in 42 message conditions or chaining variable conditions. (marked by 0* and 0 in Table 3)

0*: No appearance in 42 message bit equations and no chaining variable condition in the same bit position

0 : No appearance in 42 message bit equation, but a chaining variable condition in the same bit position

- Control Path: A chain of intermediate variable bits which can transmit a bit change from control bit m_{i',j'} to the target bit a_{i,j}.
- An example for Control Path:

$$m_{14,10} \longrightarrow a_{18,11} \longrightarrow a_{20,11} \longrightarrow a_{21,16} \longrightarrow a_{22,21} \longrightarrow a_{23,26} \longrightarrow a_{24,31}$$

Details for Message Modification — Topological Order

- A preferred order for processing a set of conditions a_{i,j} so as to minimize the chance that a previously enforced condition may later get undone.
- An example of topological order

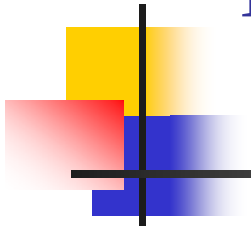
$$a_{18,2} \rightarrow a_{17,31} \rightarrow a_{17,32} \rightarrow a_{17,2} \rightarrow a_{16,31} \rightarrow a_{17,4} \rightarrow a_{20,4}$$

$$\rightarrow a_{19,32} \rightarrow a_{19,2} \rightarrow a_{18,30} \rightarrow a_{18,32} \rightarrow a_{20,30} \rightarrow a_{21,30} \rightarrow a_{21,2} \rightarrow a_{22,3}$$

$$\rightarrow a_{24,4} \rightarrow a_{23,1} \rightarrow a_{24,31} \rightarrow a_{25,31}$$

$$a_{18,29} \rightarrow (a_{19,2}, a_{18,30})$$

Details for Message Modification -----Error Probability



- Error probability In spite of topological order, there is some probability that at the end of the message modification process, not all conditions are satisfied . We refer to this probability as error probability.

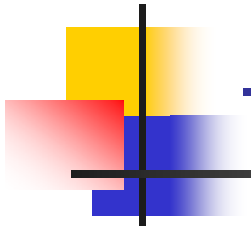
- Calculation of error probability (See Table 4)

Conditions can be Corrected by Advanced Message Modification (with Star)

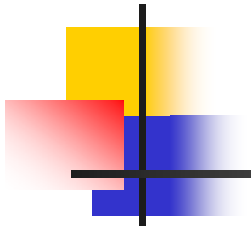
10	$a_{10,2} = 0$, $a_{10,4} = 1$, $a_{10,7} = 0$, $a_{10,8} = 0$, $a_{10,11} = a_{9,11}$, $a_{10,12} = a_{9,12}$, $a_{10,30} = 1$, $a_{10,31} = 1$,
11	$a_{11,4} = 0$, $a_{11,7} = 1$, $a_{11,8} = 1$, $a_{11,9} = 0$, $a_{11,10} = 1$, $a_{11,30} = 0$, $a_{11,31} = 1$, $a_{11,32} = 1$,
12	$a_{12,2} = 0$, $a_{12,7} = 1$, $a_{12,8} = 0$, $a_{12,32} = 1$
13	$a_{13,7} = 1$, $a_{13,8} = 1$, $a_{13,32} = 1$
14	$a_{14,3} = a_{13,4} + 1 = m_{16,1}$, $a_{14,32} = 1$,
15	$a_{15,1} = 0$,
16	$a_{16,1} = 0$,$a_{16,2} = a_{15,2}$, $a_{16,31} = 1$
17	$a_{17,2} = m_{17,2} + m_{19,7} + 1^*$, $a_{17,32} = m_{20,30}{}^*$, $a_{17,4} = m_{19,2} + m_{17,2}{}^*$, $a_{17,31} = 0^*$
18	$a_{18,2} = m_{17,2}{}^*$, $a_{18,32} = 1^*$, $a_{18,30} = 1^*$
19	$a_{19,32} = 1 + m_{19,5}{}^*$, $a_{19,2} = a_{18,2} + a_{17,2}{}^*$,
20	$a_{20,30} = 1 + a_{17,32} + a_{18,32}{}^*$, $a_{20,4} = m_{22,1} + 1 + a_{19,4}{}^*$
21	$a_{21,2} = a_{18,4} + a_{17,4}{}^*$, $a_{21,2} = m_{21,7} + 1^*$, $a_{21,30} = 1 + m_{22,30} + a_{20,32}{}^*$
22	$a_{22,3} = m_{24,1} + a_{21,3}{}^*$
23	$a_{23,1} = 1 + m_{22,1}{}^*$
24	$a_{24,4} = w_{26,2} + 1 + a_{23,4}{}^*$, $a_{24,31} = w_{25,31} + a_{22,1}{}^*$
25	$a_{25,2} = m_{24,1}$, $a_{25,31} = w_{26,31} + a_{23,1}{}^*$
26	$a_{26,2} = w_{25,2}$, $a_{26,3} = w_{28,1} + 1 + a_{25,3}$

Complexity Estimation

----Complexity for Second Iteration



- There are 83 conditions in steps 17-80
- After advanced message modification, there are 65 conditions left in 17-80 steps
- Searching for two conditions in steps 25-26 by one computation
- Relax one condition in the final step
- 62 conditions left
- Error probability for correcting 17-25 conditions amounts to one failed condition.
- The complexity is about 2^{63} computations.



Complexity Estimation ---Total Complexity

- Complexity for first iteration: further relax 3 conditions in the final 2 steps.

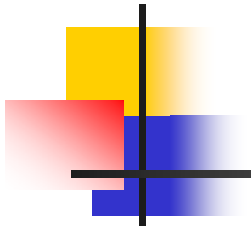
  The complexity is about 2^{60} computations

- Complexity for the second iteration

  2^{63} computations

- Total complexity

  $2^{63} + 2^{60} = 1.125 \times 2^{63} \sim 2^{63}$



Another New Collision Path for SHA-1

---Xiaoyun Wang, Andrew C. Yao and Frances Yao

2006. 2

- A new path with probability 2^{63} to search for SHA-1 collisions presented by Adi Shamir on behalf of Xiaoyun Wang, Andrew C Yao and Frances Yao in Crypt 05.
- The more details about the new path and collision attack on SHA-1 was presented in 2005 NIST hash function workshop.
- The latest new path for SHA-1 collision search was found recently.

  The purpose of the new path from that of Crypt 05 is to relax more message space from steps 11-16.

New Collision Path for SHA-1 (First Iteration)

i	x_{i-1}	Δm_{i-1}	Δa_i	Δb_i	Δc_i	Δd_i	Δe_i
1	80000001	1,-2 -30,-32	32,-1 30,-31				
2		-5,6 30	-3,30	32,-1 30,-31			
3	40000001	30,31	-31,32 3, 8,9,..,-23	-3,30	30,-31 28,-29		
4	2	-2,-4,-6 -30,31,-32	-2, 6,-7 8,13,-14, 32	-31,32 3,8,9,.-23	-1, 28	30,-31 28,-29	
5	2	-1,2,7,30	-5,6,-8,16 -23, 28	-2,6,-7 8,13,-14, 32	-29,30 1,6,7,..-21	-1,28	30,-31 28,-29
6	80000002	-7 29,-30,-32	-32 10,12,13,-14	-5,6,-8, 16 -23,28	-32, 4,-5 6,11,-12, 30	-29,30 1,6,7,..-21	-1,28
7	1	-1,2,-5,7 29,31,32	1 -16,-27,28	-32 10,12,13,-14	-3,4,-6, 14 -21,26	-32, 4,-5 6,11,-12, 30	-29,30 1,6,7,..-21
8		-2,-6 29, 31,32	4	1, -16,-27,28	-30 8,10,11,-12	-3,4,-6, 14 -21,26	-32,4,-5 6,11,-12, 30
9	80000001	-30	32,-1 9	4	31 -14,-25,26	-30 8,10,11,-12	-3,4,-6, 14 -21,26
10	2	-2,5,6 30,-31	2	32,-1 9	2	31 -14,-25,26	-30 8,10,11,-12
11	2	1,-2,7 30,31	-9	2	30,-31 7	2	31 -14,-25,26
12	2	-7,-30	2	-9	32	30,-31 7	2
13		-2,-7 -30,31,32		2	-7	32	30,-31 7
14		2,-30,-31			32	-7	32
15	1	1,32	1			32	-7
16		6		1			32

Table 6 New Correction for $a_{23,1}$

step	Δw_i	Extra Cond	Change bits	Closest Cond	Pr_1	Pr_2
5	2^{10}	$a_{5,11} = m_{4,11}$	$\mathbf{a_{5,11}}$,		1	
6	2^{15}	$m_{4,11} = 1 + m_{5,16}$			1	
7		$a_{4,13} = a_{3,13}$			1	
8		$a_{6,9} = 0$			1	
9		$a_{7,9}$=1				
10	2^8	$m_{9,9} = 1 + m_{4,11}$				
...	...	...	...	...	...	
18	2^9		$a_{18,10}$	$a_{18,30}$	$\frac{1}{2^{20}}$	
19	$\mathbf{2^{11}}$		$a_{19,12}$, $a_{19,15}$	$a_{19,32}$	$\frac{1}{2^{17}}$	
20	$\mathbf{2^{16}}$	$m_{19,17} = m_{18,12}$	$a_{20,10}$, $\mathbf{a_{20,18}}$,$a_{20,20}$	$a_{20,4}$	$\frac{1}{2^{16}}$	
21			$a_{21,8}$,...,$a_{21,15}$,$\mathbf{a_{21,23}}$, $a_{21,25}$	$a_{21,30}$	$\frac{1}{2^5}$	
22			$a_{22,10}$,...,$a_{22,20}$,$\mathbf{a_{22,28}}$, $a_{22,30}$	$a_{22,3}$	$\frac{1}{2^5}$	
23			$a_{23,8}$,$a_{23,9}$,..$a_{23,25}$, $\mathbf{a_{23,1}}$, $a_{23,3}$			$\frac{1}{2^8}$

Table 7 Old Correction for $a_{23,1}$

step	Δw_i	Extra Cons	Starting change bits	closest conditions	Pr^1	Pr^2
14	2^9	$\mathbf{a_{14,10} = m_{13,10}}$	$a_{14,10}$	$a_{14,32}$	$\frac{1}{2^{22}}$	
15	2^{14}	$m_{13,10} = m_{14,15} + 1$		$a_{15,1}$		
16		$c_{15,10} = d_{15,10}$		$a_{16,31}$		
17	2^{10}	$b_{16,8} = 0$ $a_{17,11} = m_{16,11}$	$a_{17,11}$	$a_{17,31}$	$\frac{1}{2^{20}}$	
18	2^{15}	$b_{17,8} = 0$ $m_{17,16} = 1 + m_{16,11}$	$\mathbf{a_{18,8}}$	$a_{18,30}$	$\frac{1}{2^{22}}$	
19		$c_{18,11} = d_{18,11}$	$a_{19,8}$, $\mathbf{a_{19,13}}$	$a_{19,32}$	$\frac{1}{2^{19}}$	
20	2^{11}		$a_{20,8}$,...., $a_{20,13}$, $\mathbf{a_{20,18}}$	$a_{20,4}$	$\frac{1}{2^{18}}$	
21	2^{16}		$a_{21,6}$,...., $a_{21,18}$, $\mathbf{a_{21,23}}$	$a_{21,30}$	$\frac{1}{2^7}$	
22			$a_{22,6}$, ...,$a_{22,23}$,$\mathbf{a_{22,28}}$	$a_{22,3}$	$\frac{1}{2^7}$	
23			$a_{23,6}$, ..., $a_{23,28}$, $\mathbf{a_{23,1}}$	$a_{23,1}$		$\frac{1}{2^5}$

Advantages for the SHA-1 New Collision Path

- Remove two 6-step collisions in steps 9-15

  decrease 6 conditions in steps 11-16.

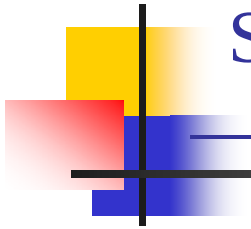
  decrease another 6 conditions in steps 11-16 for the correction of $a_{23,1}$.

  relax 2^{12} times message space from word 10 to word 15 to search for the collision.
- For the old path (Crypto rump session and NIST workshop), the message space from step 11 to step 16 is tight
- For the new, the message space from step 11 to step 16 is available to search for collisions which is about 2^{91}

Table 8 An 1-10 Steps Differential Sample for the New Path

M	b6801432	15f811d6	85dfed48	a60d0f6b	81d83f85	174997c2	211d1939	57fc72e	746f8260	cfceb742
M'	16801431	35f811e6	e5dfed48	460d0f41	a1d83fc6	a7499782	f11d196a	d57fc70c	546f8260	afceb770
ΔM	a0000003	20000030	60000000	e000002a	20000043	b0000040	d0000053	d0000022	20000000	60000032

Table 16: A example for two messages which satisfying the conditions in first 10 steps

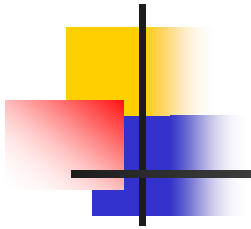


Some Potential Dangers for Hash Function

—The Estimated Time to Find SHA-1 Collision is Not Long

- The message modification becomes more easier because of message space enlarging.
- With the new path, it is possible to find the first iteration differential path by four conditions relax in the final two steps. The computations is about 2^{59}.
- After finding the first iteration path, it is possible to find the second iteration path with 1-2 more conditions corrected, the computation is about 2^{61}-2^{62} computations.

This is only an estimation by the existing techniques.



Some Potential Dangers for Hash Functions

---Second Pre-image Attack for MD4(1)

Hans Dobbertin described the definitions of weak message and close message by low hamming weight for the second pre-image attack during my visit of Ruhr University in Nov. of 2004.

Weak message Given a message M, there exists an efficient attack to find its second-preimage. M is a weak message.

The modular differential attack was suggested by Hans Dobbertin and Magnus.

Some Potential Dangers for Hash Functions

---Second Pre-image Attack for MD4(2)

- In CANS 05, Hongbo Yu and Xiaoyun Wang gave a new collision path for second pre-image attack on MD4.
- Any message is a weak message with probability 2^{-56} by the new collision differential path .
- For any message M, after the basic message modification, only 38 conditions in 2-3 rounds left, so the resulting message M is a weak message with probability 2^{-38} and their hamming weight is only about 12. To find the meaningful message and it's meaningful second pre-image is available.
- Using the advanced message modification, M is a weak message with probability 2^{-27}.
- So any message can be converted into a weak message with 2^{27} MD4 computations and the Hamming weight is about 44.

An Example for MD4 Second Pre-image

M_0	ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff ffffffff
M	ffffffff ffffffff ffffffff ffffffff ffbfffff fffbffff ffffbfff fffff9ff ffffffff ffdfffff fffbffff ffffefff ffffffff f73ff7ff adac30f9 2e2b883e
M’	ffffffff ffffffff ffffffff ffffffff ffffffff fffbffff ffffbfff fffff9ff ffffffff ffdfffff fffbffff ffffefff ffffffff f73ff7ff adac30f9 2e2b983e
H	36c6ff7 b4f8abf9 bcaaff6e faa6e73d

Some Potential Dangers for Hash Functions

---A Solution to Find Collisions of MD5 with Any Two IVs(1)

- This question was suggested by Arjen K. Lenstra.

  MD5(IV, M)=MD5(IV', M')

- Xioyun Wang and Jinfu Xu gave the following solution.

1 **Birthday attack** search for two messages M1and M2 with about 2^{48} computations such that

IV1=(a1, b1, c1,d1)=MD5(IV, M1).

IV1'= (a1', b1', c1', d1')=MD5(IV', M1').

(△a1, △b1,△c1, △d1)=(a1'-a1, b1'-b1, c1'-c1, d1'-d1).

△a1=0, △b1=△c1= △d1

Some Potential Dangers for Hash Functions

---A Solution to Find Collisions of MD5 with Any Two IVs(2)

2 Find multi-block collisions such that

MD5(IV1, M2)=MD5(IV1', M2')

By searching for several near-collisions to cancel all the non-zero bit differences in △b1(=△c1= △d1). Each message block pair will cancel a few non-zero bit differences in △b1.

For example, if one message block pair cancels 4 non-zero bit differences (any locations), then it is possible to find 5 message block pair have the same hash values. One message pair will produce the difference in item 1, and other four message block pairs will cancel all the non-zero bit differences in △b1, and there are about 16 non-zero bit differences in △b1.

It is remarked that the item 2 is a theoretical result.

Thanks!