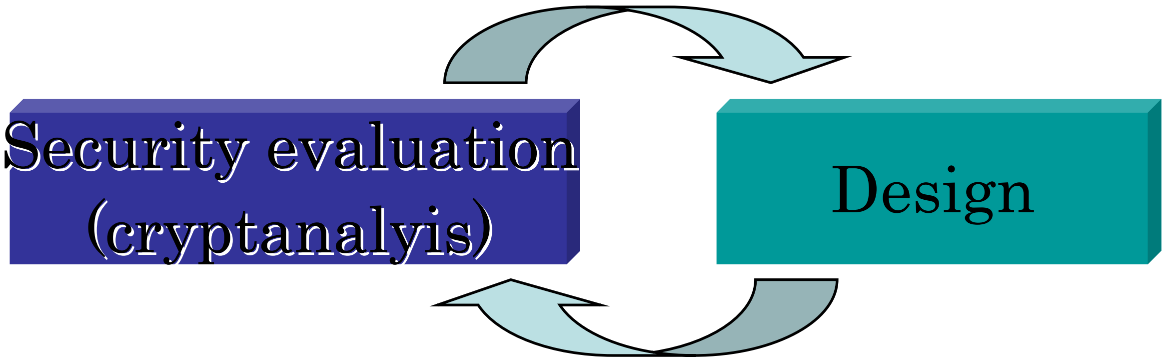
Security Evaluation of Hash Functions: Gröbner Basis Based Cryptanalysis of SHA-1

Makoto Sugita
IPA Security Center

Part I
Japanese Standardization Effort (CRYPTREC)

Security evaluation methods and the design of cryptographic algorithms

- Generally - No definitive security proof of a cryptographic algorithm against the attacks
- A cryptographic algorithm is considered as a secure one only if it is secure against all known attacks



CRYPTREC

Cryptographic Technology Evaluation Committee hosted by the Cryptographic Technology Investigative Committee organized by MIC/METI on conjoint basis and Cryptographic Technology Monitoring Committee and Cryptographic Module Committee organized by IPA and NICT on conjoint basis.

MIC: Ministry of International Affairs and Communications
METI: Ministry of Economy, Trade and Industry
IPA: Information-Technology, Promotion Agency
NICT: National Institute of Information and Communications Technology

The Mission Assigned to The Cryptography Research Group of IPA Security Center

Ensuring the Security of Cryptographic Algorithms

What is "Ensuring the Security of Cryptographic Algorithms"?

- **2000 ～ 2002:** Cryptographic Technology Evaluation
- **Feb. 2003:** Publication of the e-Government Recommended Cipher List
- **Feb. 2003:** Monitoring the Current Tendency of Cryptographic Study
  - Not only study the current tendency in cryptanalyzing research for cryptosystems but also **researches the ways to cryptanalyze by ourselves**

What Does The e-Government Recommended Cipher List Look Like?

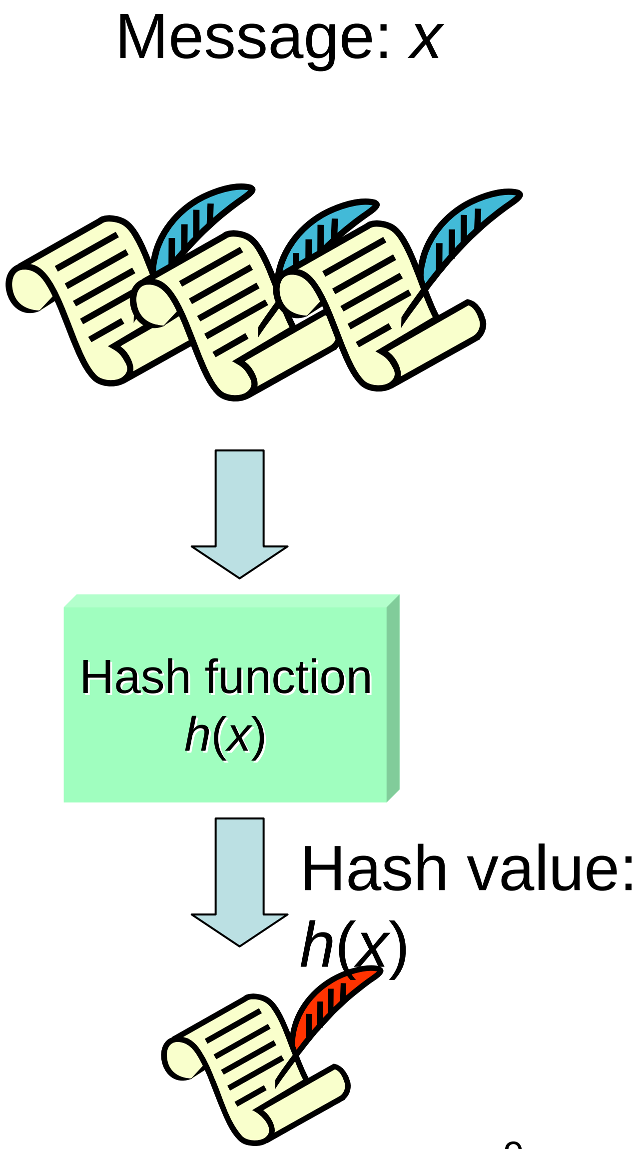
Classification in Technical		Appellation
Public Key	Signature	DSA
		ECDSA
		RSASSA PKCS1 v1_5
		RSA-PSS
	Confidentiality	RSA-OAEP
		RSAES-PKCS1 v1_5
	Key Agreement	DH
		ECDH
		PSEC-KEM
Symmetric Key	64 Bit Block Cipher	CIPHERUNIORN-E
		Hierocrypt-L1
		MISTY1
		3-key Triple DES
	128 Bit Block Cipher	AES
		Camellia
		CIPHERUNICORN-A
		Hierocrypt-3
		SC2000
	Stream Cipher	MUGI
		MULTI-S01
		128-bit RC4
Others	Hash Function	RIPEMD-160
		SHA-1
		SHA-256
		SHA-384
		SHA-512
	Pseude-Random Number Generator	PRNG based on SHA-1 in ANSI X9.42-2001 Annex C.1
		PRNG based on SHA-1 for general purpose in FIPS 186-2 (+change notice 1) Appendix 3.1
		PRNG based on SHA-1 for general purpose in FIPS 186-2

Part II
Gröbner Basis Based Cryptanalysis of SHA-1

Joint work with Mitsuru Kawazoe (Osaka Prefecture university) and Hideki Imai (Chuo University and RCIS, AIST)

Hash function

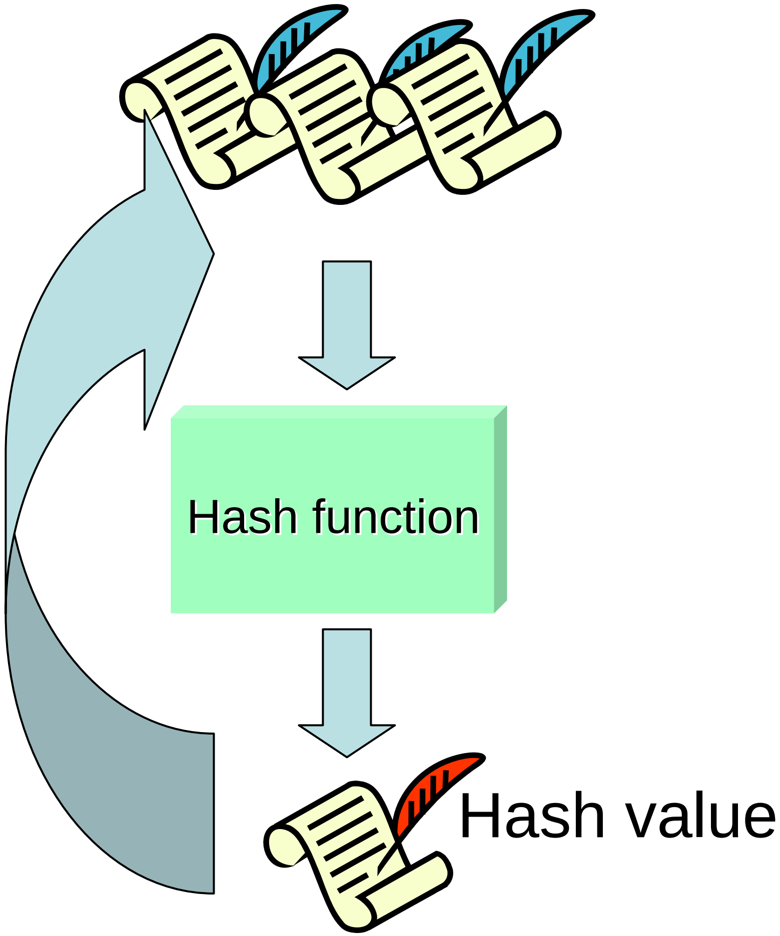
- Cryptographic hash function $y = h(x)$
  - Take a message x of arbitrary length
  - Output a short value y of a fixed length
- Basic security property
  - One-way: given y, hard to find x s.t. $x = h^{-1}(x)$
  - Collision resistant: hard to find $x \neq y$ s.t. $h(x) = h(y)$
- Applications
  - Digital signature, password verification, key generation…
  - Employed in almost all security systems



Two major attacks on hash functions

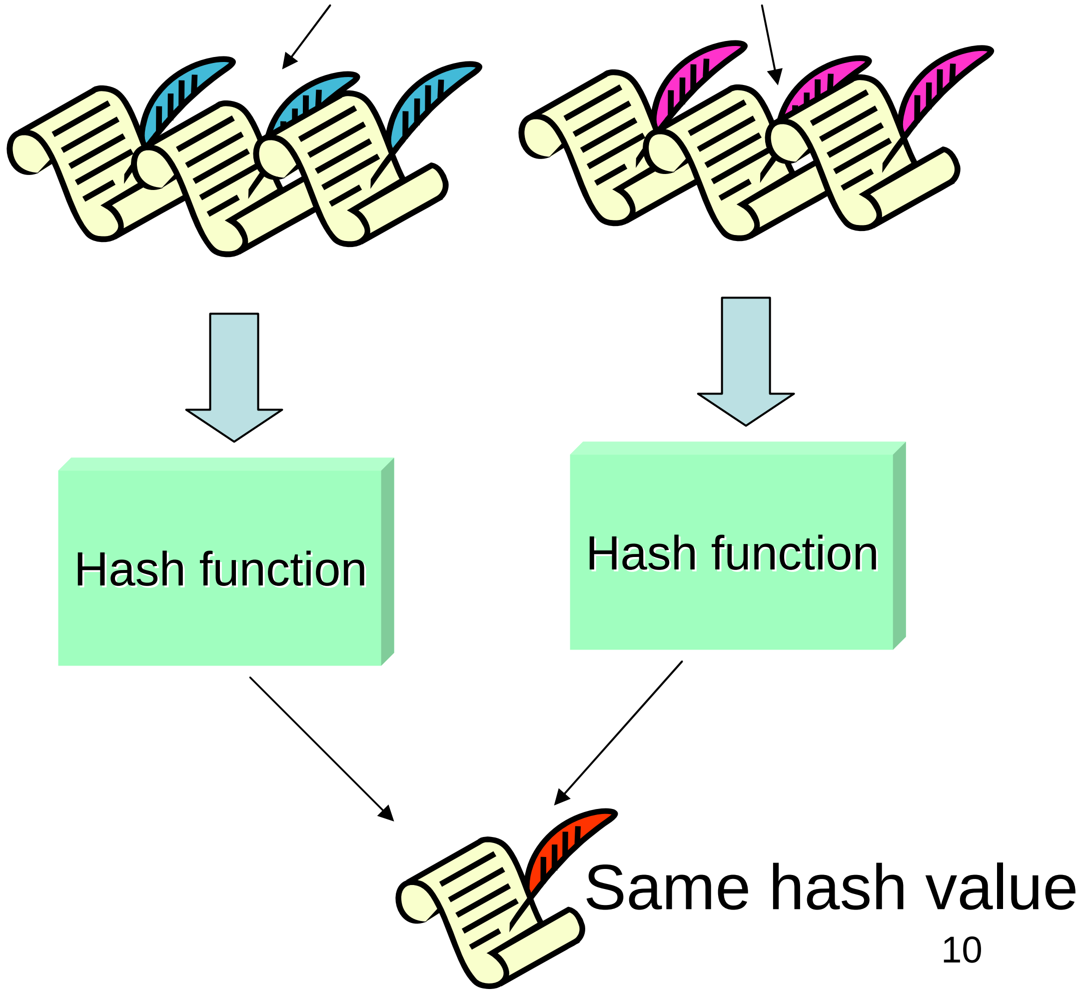
(2nd) preimage attack

Guess Message
from hash value



collision attack

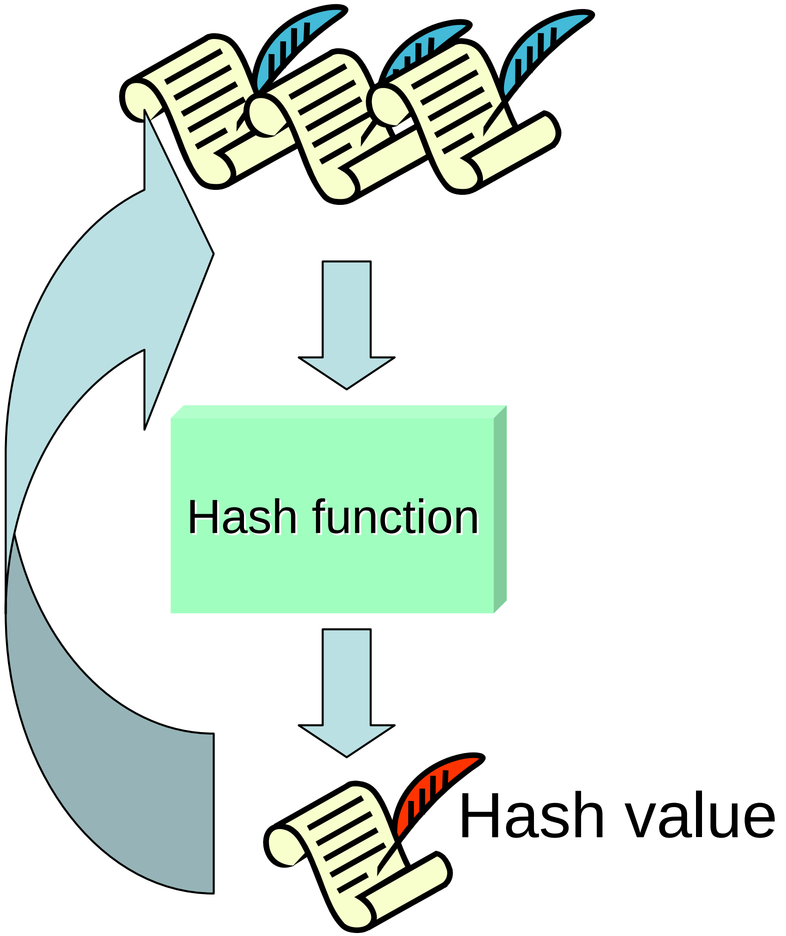
Find two messages s.t.
hash values are same



Two major attacks on hash functions

(2nd) preimage attack

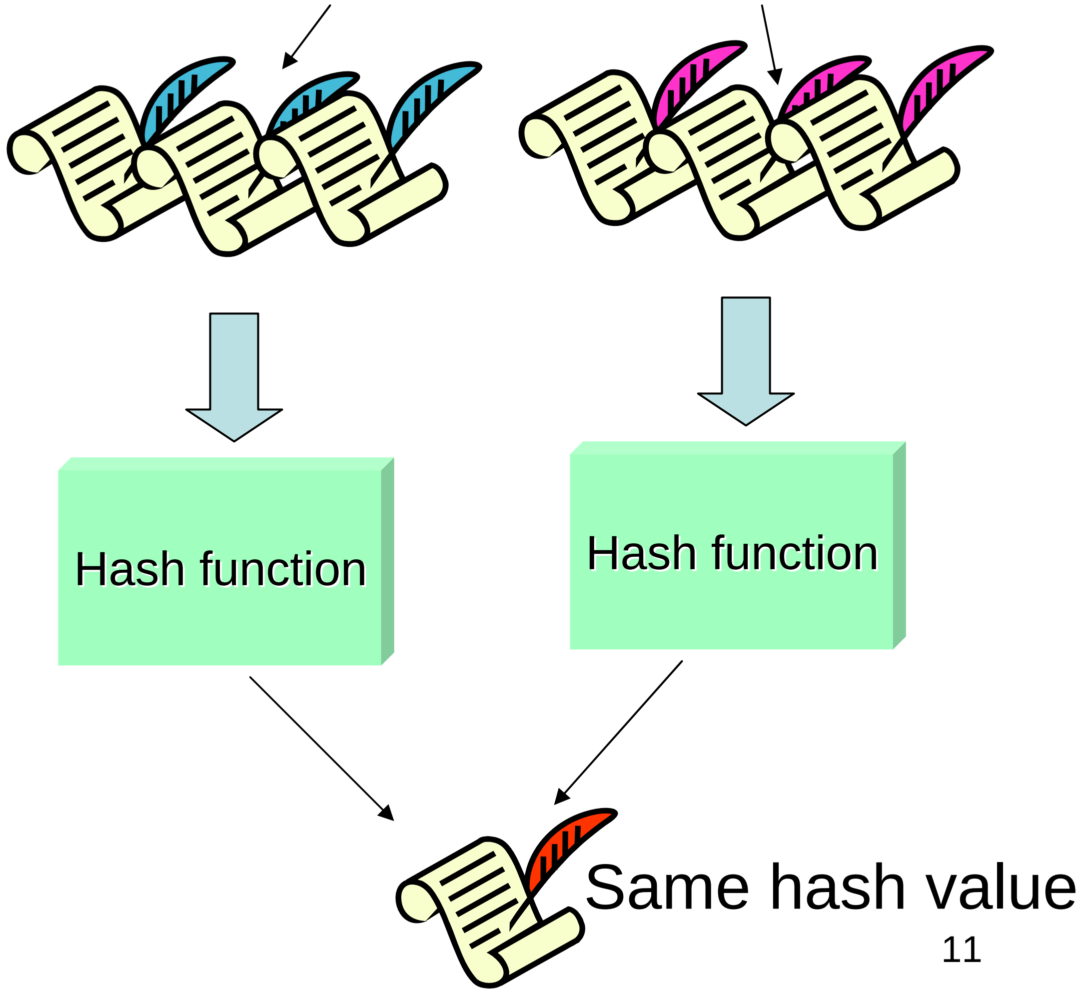
Guess Message
from hash value



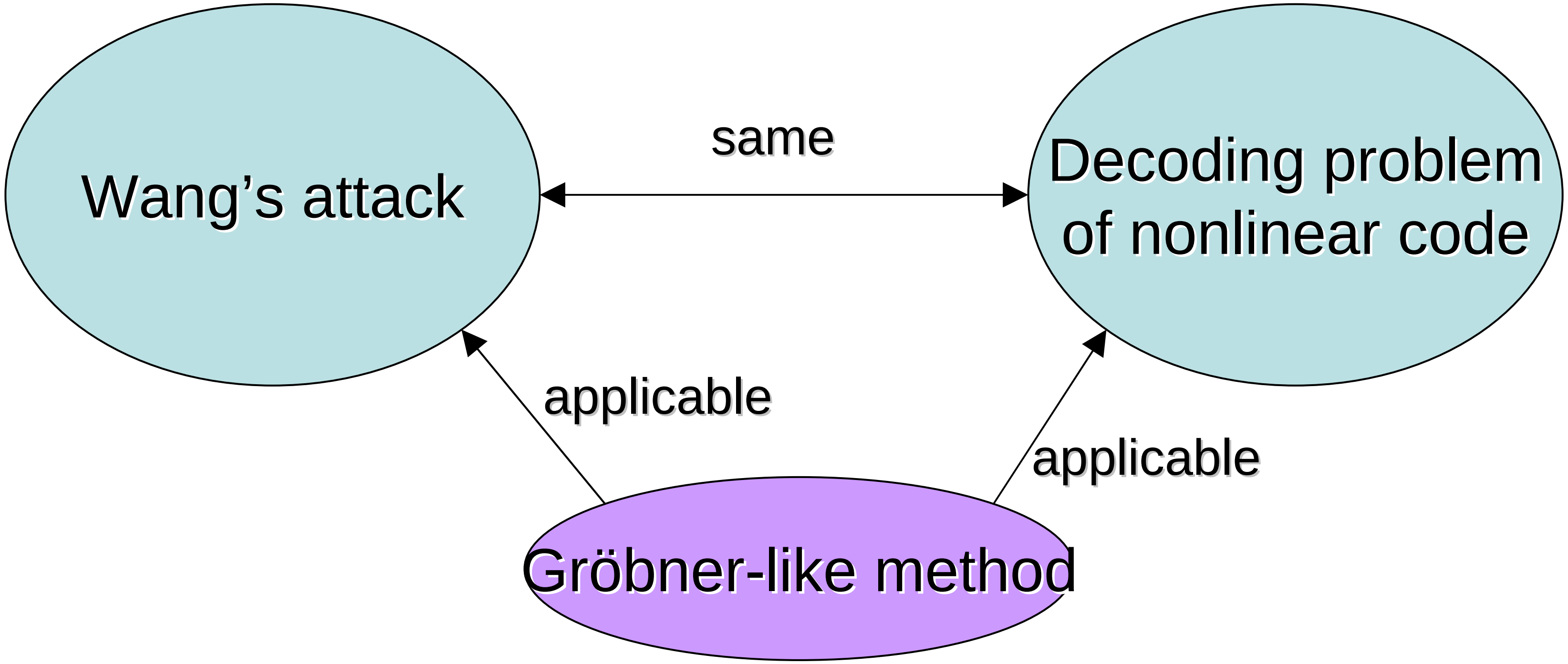
We treat
this attack

collision attack

Find two messages s.t.
hash values are same

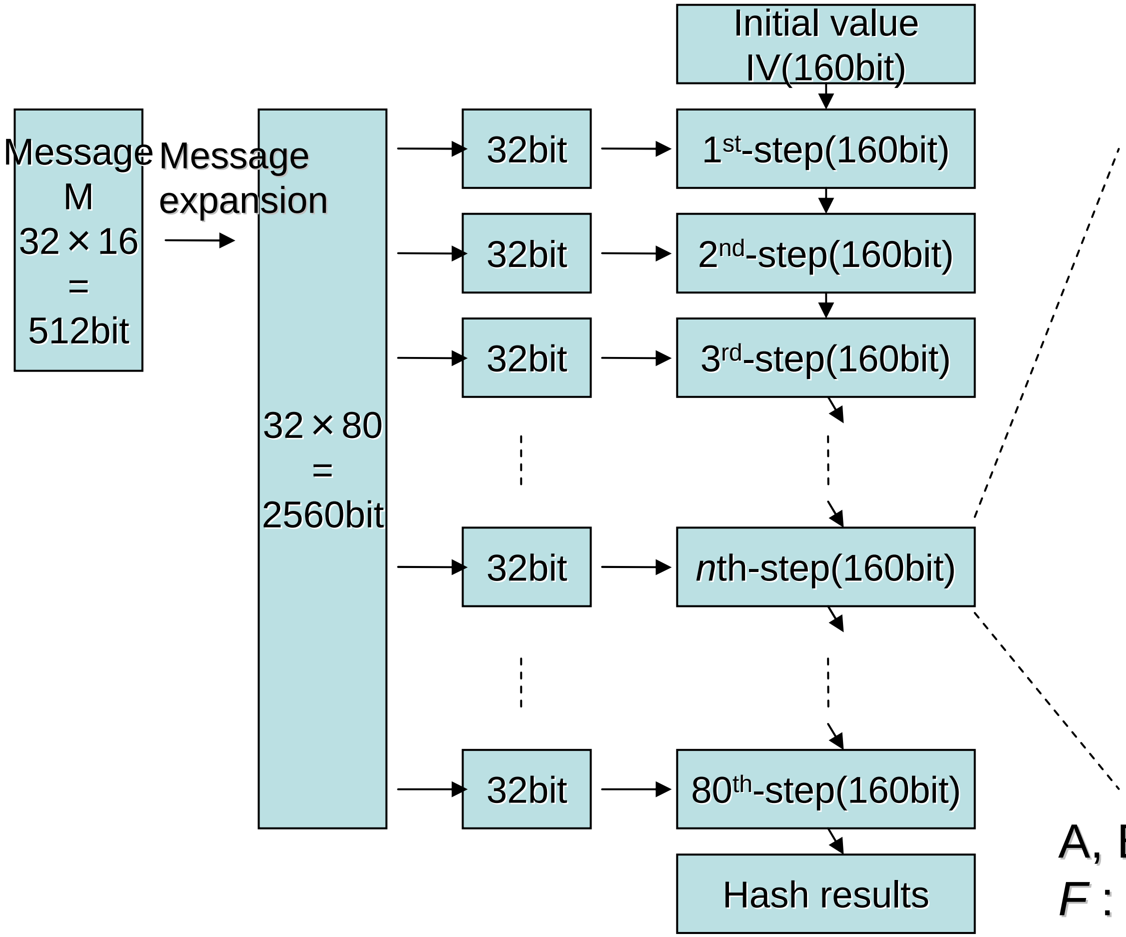


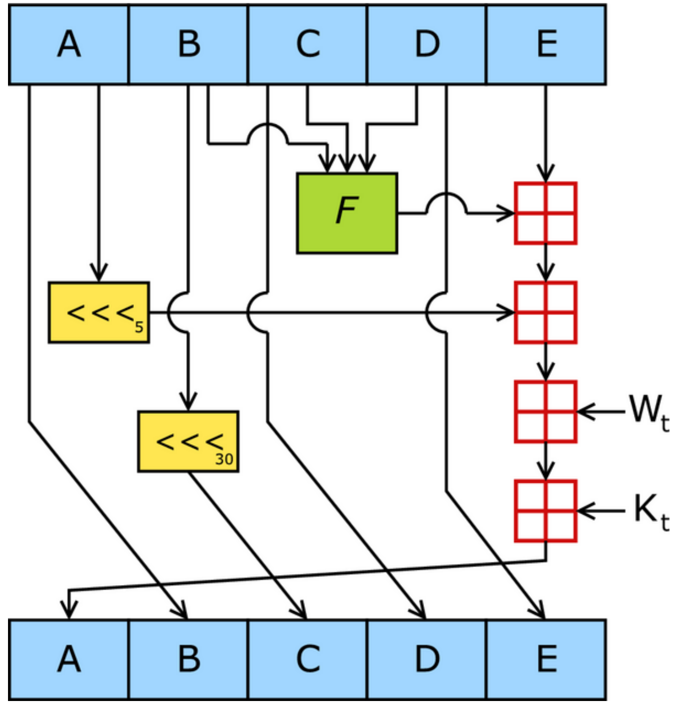
Wang's attack, nonlinear code and Gröbner basis



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

Structure of hash function SHA-1





A, B, C, D,E :32-bit words of the state

F : nonlinear function

$<<<_s$: left bit rotation by *s* places;

$\boxplus$: addition modulo 2^{32}.

Kt :constant.

Definition of SHA-1

The hash function SHA-1 generates 160-bit hash result from message of length less than 2^{64} bits. It has Merkle/Damgard structure like other hash functions, and has 160-bit chaining value and 512-bit message block, and initial chaining values (IV) are fixed. From 512-bit block of the padded message, SHA-1 divides it into 16×32-bit words $(m_0, m_1, \cdots, m_{15})$ and expands the message by

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

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

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

where initial chaining value $IV = (a_0, b_0, c_0, d_0, e_0)$ is

$$(0x67452301, 0xefcdab89, 0x98badcfe, 0x10325476, 0xc3d2e1f0)$$

and function f_i is defined as in Table 1. In the following, we express 32-bit words as hexadecimal numbers.

Description of SHA-1 algorithm

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

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

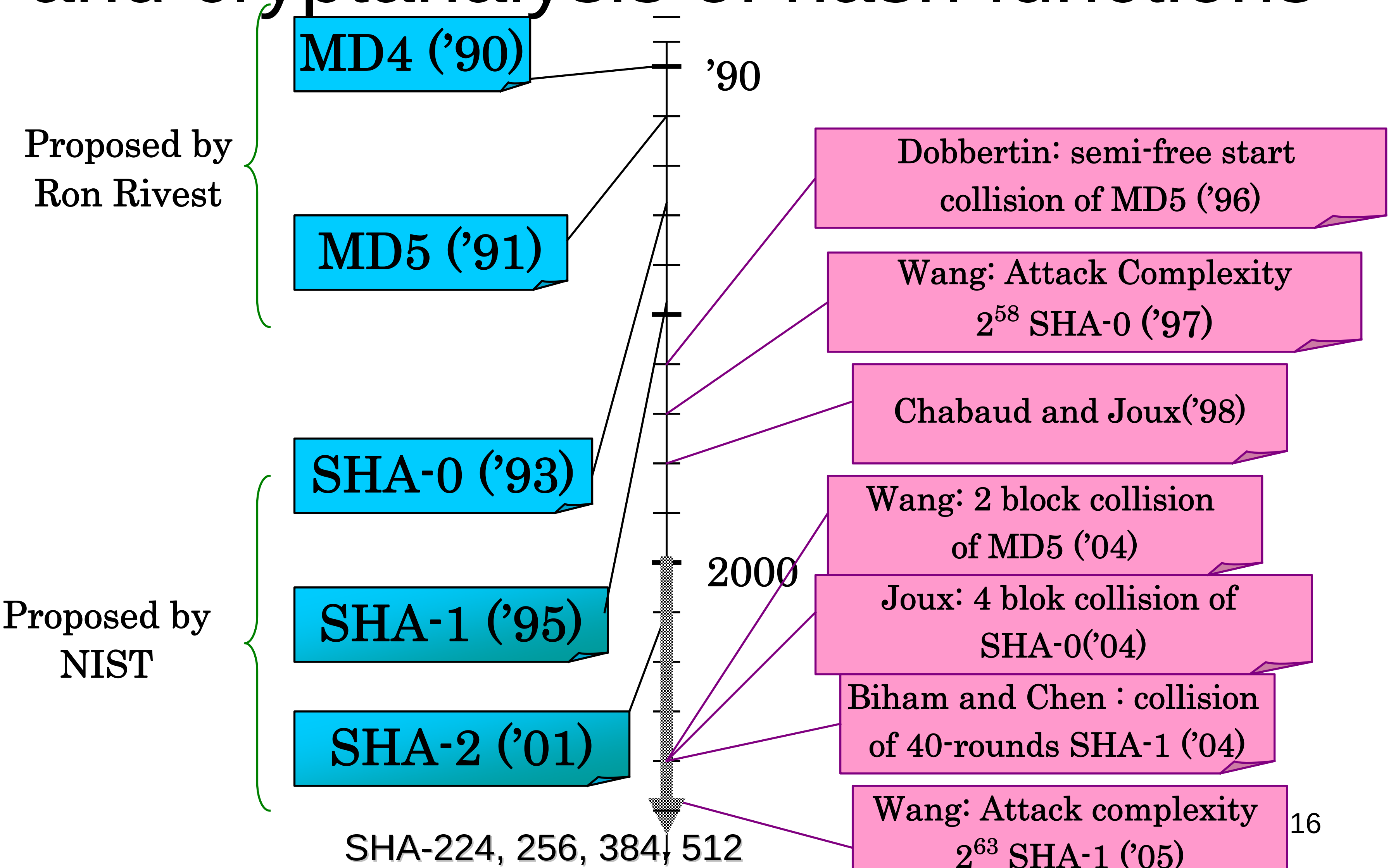
$$
\begin{aligned}
a_i &= (a_{i-1} \lll 5) + f_i(b_{i-1}, c_{i-1}, d_{i-1}) + e_{i-1} + m_{i-1} + k_i \\
b_i &= a_{i-1} \\
c_i &= b_{i-1} \lll 30 \\
d_i &= c_{i-1} \\
e_i &= d_{i-1}
\end{aligned}
$$

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

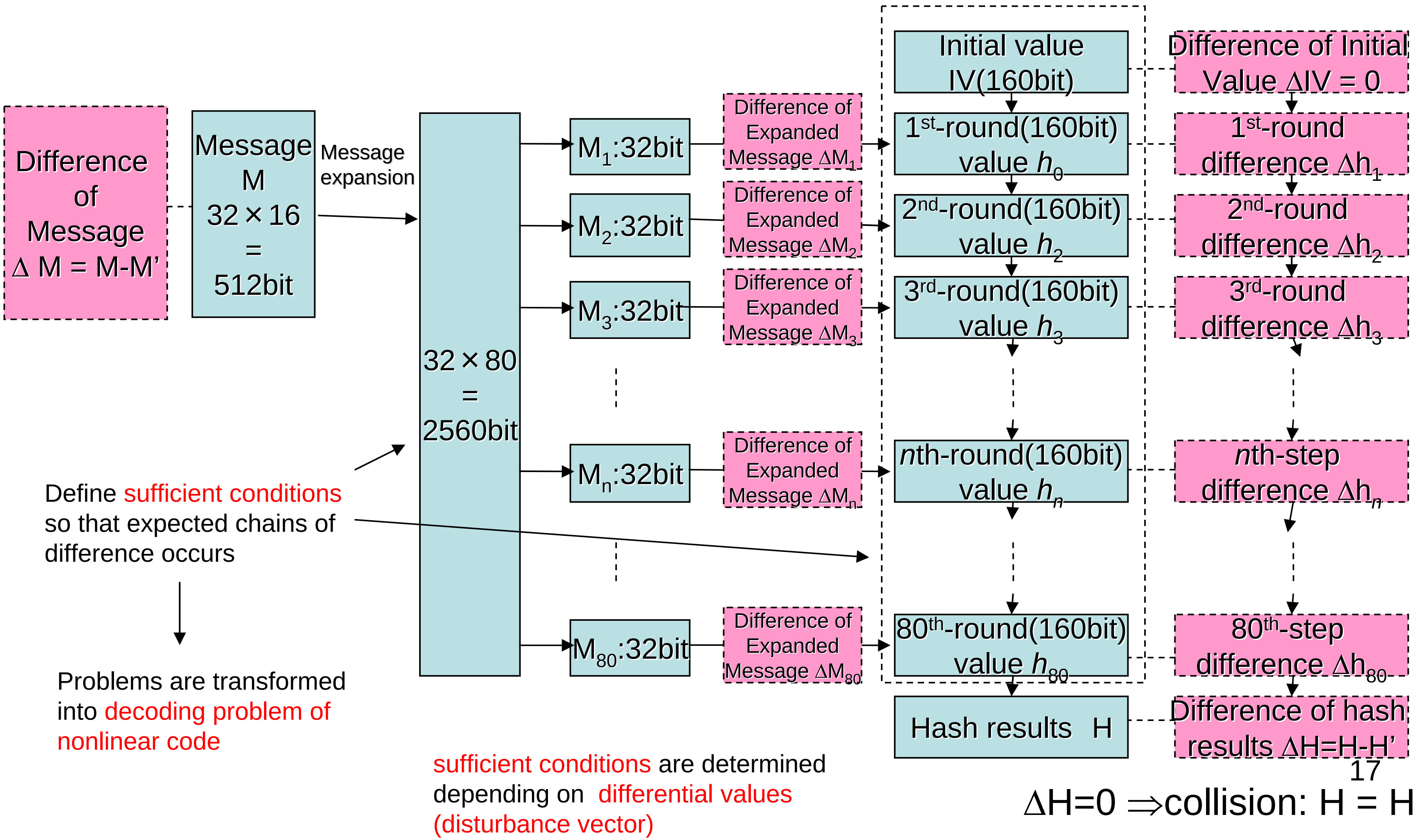
round	step	Boolean function f_i	constant k_i
1	$1-20$	IF: $(x \wedge y) \vee (\neg x \wedge z)$	$0x5a827999$
2	$21-40$	XOR: $x \oplus y \oplus z$	$0x6ed6eba1$
3	$41-60$	MAJ: $(x \wedge y) \wedge (x \vee z) \wedge (y \vee z)$	$0x8fabbcdc$
4	$61-80$	XOR: $x \oplus y \oplus z$	$0xca62c1d6$

Table 1 Definition of function f_i

A history of hash function proposals and cryptanalysis of hash functions



Differential cryptanalysis against Hash functions



ΔH=0 ⇒collision: H = H'

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

Disturbance vector and sufficient conditions

Disturbance vector

- ΔM = Disturbance vector
  - There exist messages m, m' s.t. $\Delta M = m - m'$
- SD: Sufficient conditions (w.r.t. ΔM)
  - If a message m satisfies SD, then $h(m)=h(m+\Delta M)$

Message modification

- M: a randomly chosen message
- $M \rightarrow M'$ such that M' satisfies SD

Sufficient condition and message modification techniques by Wang

chaining variable	conditions on bits			
	32 – 25	24 – 17	16 – 9	8 – 1
a_1	a00-----	--------	1-----aa	1-0a11aa
a_2	01110---	------1-	0aaa-0--	011-001-
a_3	0-100---	-0-aaa0-	--0111--	01110-01
a_4	10010---	a1---011	10011010	10011-10
a_5	001a0---	--01-000	10001111	-010-11-
a_6	1-0-0011	1-1001-0	111011-1	a10-00a-
a_7	0---1011	1a0111--	101--010	-10-11-0
a_8	-01---10	000000aa	001aa111	---01-1-
a_9	-00-----	10001000	0000000-	---11-1-
a_{10}	0-------	1111111-	11100000	0-----0-
a_{11}	--------	------10	11111101	1-a--0--
a_{12}	0-------	--------	--------	10--11--
a_{13}	--------	--------	--------	11----10
a_{14}	-0------	--------	--------	----0-1-
a_{15}	10------	--------	--------	----1-0-
a_{16}	--1-----	--------	--------	----0-0-
a_{17}	0-0-----	--------	--------	------1-
a_{18}	--1-----	--------	--------	----a---
a_{19}	--b-----	--------	--------	------0-
a_{20}	--------	--------	--------	-----a--
a_{21}	--------	--------	--------	-------1

Table 10 A set of sufficient conditions on a_i for the 80-step differential path given in Table 9. b denote the condition $a_{19,30} = a_{18,32}$

Method for determining sufficient conditions is 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

Definitions of differential and disturbance vector

Definition 1. *Let m_i and a_i be as in the definition of SHA-1. When we consider a_i as a vector of $\mathbb{F}_2^{32}$, let $a_{i,j}$ be the jth bit of variable a_i. Let m'_i and a'_i be another pair and consider the difference $\Delta a_i := a'_i - a_i$. Then for Δa_i, we define the following notation.*

$$\Delta^+ a_{i,j} = \begin{cases} 1 & \text{if } a'_{i,j} = 1 \text{ and } a_{i,j} = 0 \\ 0 & \text{otherwise,} \end{cases} \qquad \Delta^- a_{i,j} = \begin{cases} 1 & \text{if } a'_{i,j} = 0 \text{ and } a_{i,j} = 1 \\ 0 & \text{otherwise,} \end{cases}$$

We define $\Delta^\pm a_{i,j}$ by $\Delta^\pm a_{i,j} = \Delta^+ a_{i,j} \oplus \Delta^- a_{i,j}$. Moreover, we define $\Delta^+ a_i = (\Delta^+ a_{i,0}, \Delta^+ a_{i,1}, \cdots, \Delta^+ a_{i,31})$, $\Delta^- a_i = (\Delta^- a_{i,0}, \Delta^- a_{i,1}, \cdots, \Delta^- a_{i,31})$ and $\Delta^\pm a_i = \Delta^+ a_i \oplus \Delta^- a_i$. Similarly, for m, b, c, d, e, we define $\Delta^+ m_{i,j}$, $\Delta^- m_{i,j}$, $\Delta^\pm m_{i,j}$, $\Delta^+ m_i$, $\Delta^- m_i$, $\Delta^\pm m_i$, and so on. Following Wang's notation, we call a vector in the form $(\Delta^\pm m_i, \Delta^\pm a_i, \Delta^\pm b_i, \Delta^\pm c_i, \Delta^\pm d_i, \Delta^\pm e_i)$ a "disturbance vector", and $(\Delta^+ m_i, \Delta^- m_i, \Delta^+ a_i, \Delta^- a_i, \Delta^+ b_i, \Delta^- b_i, \ldots, \Delta^+ e_i, \Delta^- e_i)$ a "differential without carry".

How to find disturbance vector?

See our preprint, but after that, some better methods have already been published by other teams.

How to calculate sufficient conditions?

Definition and proposition

Definition 2: For a message space $M = \mathbb{Z}/2^{32}\mathbb{Z}$, we define function $f : (M \times M) \to M : (x_1, x_2) \mapsto (x_1 - x_2)$ where we consider $'-'$ as subtraction of $\mathbb{Z}/2^{32}\mathbb{Z}$. We define differential δM by $\delta M = (M \times M)/\sim$ where for $\delta m_1, \delta m_2 \in \delta M$, $\delta m_1 \sim \delta m_2$ is satisfied if and only if $f(\delta m_1) = f(\delta m_2)$.

Proposition 1: $\delta M \cong M$

proof) This is obvious from the definition of δM.

Definitions

In calculation, we use the following steps.

- Calculate $\delta m_3 = (\Delta^+ m_3, \Delta^- m_3) = \delta m_1 + \delta m_2 = (\Delta^+ m_1 + \Delta^+ m_2, \Delta^- m_1 + \Delta^- m_2)$.
- Cancel the bit of $(\Delta^+ m_3, \Delta^- m_3)$: If $\Delta^+ m_{3,j} = \Delta^- m_{3,j} = 1$, change $\Delta^+ m_{3,j} = \Delta^- m_{3,j} = 0$.

We define operator $-$ in δM as follows. For $\delta m_1 = (\Delta^+ m_1, \Delta^- m_1)$, $\delta m_2 = (\Delta^+ m_2, \Delta^- m_2)$,

$$\delta m_1 - \delta m_2 = (\Delta^+ m_1 + \Delta^- m_2, \Delta^- m_1 + \Delta^+ m_2)$$

In calculation, we also use the steps given below.

- Calculate $\delta m_3 = (\Delta^- m_3, \Delta^- m_3) = \delta m_1 - \delta m_2 = (\Delta^+ m_1 + \Delta^- m_2, \Delta^- m_1 + \Delta^+ m_2)$
- Cancel the bit of $(\Delta^+ m_3, \Delta^- m_3)$: If $\Delta^+ m_{3,j} = \Delta^- m_{3,j} = 1$, change $\Delta^+ m_{3,j} = \Delta^- m_{3,j} = 0$.

In order to check whether $\delta m_1 = \delta m_2$ or not, calculate $\delta m_1 - \delta m_2$ and check $\delta m_1 - \delta m_2 = (0, 0)$.

How to find sufficient conditions on a_i?

- Ignore message expansion in this step

We will calculate sufficient conditions of chaining variables by adjusting b_i, c_i, d_i so that

$$\delta f(i, b_i, c_i, d_i) = \delta a_{i+1} - (\delta a_i \lll 5) - \delta e_i - \delta m_i.$$

In this calculation, we must adjust carry effect by hand, where we must take into account that when $\delta a_{i+1,j} = (\delta a_i \lll 5)_j = \delta e_{i,j} = \delta m_{i,j} = 0$, $\delta f(i, b_i, c_i, d_i)_j$ must be 0, not 1. Adjusting carry effect is difficult to calculate automatically.

Sufficient conditions of full-round SHA-1 by Wang

chaining variable	conditions	on bits		
	32 – 25	24 – 17	16 – 9	8 – 1
a_1	a00-----	--------	1-----aa	1-0a11aa
a_2	01110---	------1-	0aaa-0--	011-001-
a_3	0-100---	-0-aaa0-	--0111--	01110-01
a_4	10010---	a1---011	10011010	10011-10
a_5	001a0---	--01-000	10001111	-010-11-
a_6	1-0-0011	1-1001-0	111011-1	a10-00a-
a_7	0---1011	1a0111--	101--010	-10-11-0
a_8	-01---10	000000aa	001aa111	---01-1-
a_9	-00-----	10001000	0000000-	---11-1-
a_{10}	0-------	1111111-	11100000	0-----0-
a_{11}	--------	------10	11111101	1-a--0--
a_{12}	0-------	--------	--------	10--11--
a_{13}	--------	--------	--------	11----10
a_{14}	-0------	--------	--------	----0-1-
a_{15}	10------	--------	--------	----1-0-
a_{16}	--1-----	--------	--------	----0-0-
a_{17}	0-0-----	--------	--------	------1-
a_{18}	--1-----	--------	--------	----a---
a_{19}	--b-----	--------	--------	------0-
a_{20}	--------	--------	--------	-----a--
a_{21}	--------	--------	--------	-------1

Table 10 A set of sufficient conditions on a_i for the 80-step differential path given in Table 9. b denote the condition $a_{19,30} = a_{18,32}$

Sufficient conditions of message *m* in 58-round SHA-1

message variable	31 - 24	23 - 16	15 - 8	8 - 0
m_0	--0-----	--------	--------	--------
m_1	-01-----	--------	--------	--01--1-
m_2	-10-----	--------	--------	-1----11
m_3	--0-----	--------	--------	-1------
m_4	000-----	--------	--------	-0----1-
m_5	-11-----	--------	--------	------1-
m_6	0-------	--------	--------	-------0
m_7	--------	--------	--------	--1-----
m_8	--------	--------	--------	------00
m_9	-0------	--------	--------	-0-1--1-
m_{10}	-0------	--------	--------	-0------
m_{11}	101-----	--------	--------	-1-1--1-
m_{12}	1-1-----	--------	--------	--------
m_{13}	0-------	--------	--------	-0------
m_{14}	--0-----	--------	--------	-------0
m_{15}	--0-----	--------	--------	-11-----
m_{16}	0-------	--------	--------	-------0
m_{17}	-0------	--------	--------	-1----0-
m_{18}	00------	--------	--------	-1----01
m_{19}	-0------	--------	--------	--1---1-
m_{20}	--------	--------	--------	------11
m_{21}	-0------	--------	--------	-0----1-
m_{22}	01------	--------	--------	-0----10

Sufficient conditions of chaining variables *a* in 58-round SHA-1

chaining variable	31 - 24	23 - 16	15 - 8	8 - 0
a_0	01100111	01000101	00100011	00000001
a_1	101-----	--------	--------	-1-a10aa
a_2	01100---	------0-	----a---	1--00010
a_3	0010----	-10---1a	------0-	0a-1a0-0
a_4	11010---	-01-----	01aaa---	0-10-100
a_5	10-01a--	-1-01-aa	--00100-	0---01-1
a_6	11--0110	-a-1001-	01100010	1-a111-1
a_7	-1--1110	a1a1111-	-101-001	1---0-10
a_8	-0----10	0000000a	a001a1--	100-0-1-
a_9	00------	11000100	00000000	101-1-1-
a_{10}	0-1-----	11111011	11100000	00--0-1-
a_{11}	1-0-----	-------1	01111110	11----0-
a_{12}	0-1-----	--------	--------	-1--a---
a_{13}	1-0-----	--------	--------	-1---01-
a_{14}	1-------	--------	--------	-1---1--
a_{15}	0-------	--------	--------	----0--0
a_{16}	-1------	--------	--------	----a---
a_{17}	-0------	--------	--------	----1-0-
a_{18}	1-1-----	--------	--------	----a-0-
a_{19}	--------	--------	--------	-------0
a_{20}	-C------	--------	--------	----A---
a_{21}	--------	--------	--------	----a-1-
[illegible]	--------	--------	--------	-----A1-

Procedures for Message modification

- Our method

Two Elimination Orders

- Elimination order of *m*

Here we introduce elimination order of $\{m_{i,j}\}\{i = 0, 1, \cdots, 15, j = 0, 1, \cdots, 31\}$ by

$$m'_{i',j'} \leq m_{i,j} \text{ if } i' \leq i \text{ or } (i' = i \text{ and } j' \leq j).$$

- Elimination order of *a*

Similarly we can consider different elimination order of $a_{i,j}\{i = 0, 1, \cdots, 15, j = 0, 1, \cdots, 31\}$ by

$$a'_{i',j'} \leq a_{i,j} \text{ if } i' \leq i \text{ or } (i' = i \text{ and } j' \leq j).$$

These two orders are different but approximately similar because transformation between them is not so complicated.

Sufficient conditions of message

message variable	31 - 24	23 - 16	15 - 8	8 - 0
m_0	--0-----	--------	--------	--------
m_1	-01-----	--------	--------	--01--1-
m_2	-10-----	--------	--------	-1----11
m_3	--0-----	--------	--------	-1------
m_4	000-----	--------	--------	-0----1-
m_5	-11-----	--------	--------	------1-
m_6	0-------	--------	--------	-------0
m_7	--------	--------	--------	--1-----
m_8	--------	--------	--------	------00
m_9	-0------	--------	--------	-0-1--1-
m_{10}	-0------	--------	--------	-0------
m_{11}	101-----	--------	--------	-1-1--1-
m_{12}	1-1-----	--------	--------	--------
m_{13}	0-------	--------	--------	-0------
m_{14}	--0-----	--------	--------	-------0
m_{15}	--0-----	--------	--------	-11-----
m_{16}	0-------	--------	--------	-------0
m_{17}	-0------	--------	--------	-1----0-
m_{18}	00------	--------	--------	-1----01
m_{19}	-0------	--------	--------	--1---1-
m_{20}	--------	--------	--------	------11
m_{21}	-0------	--------	--------	-0----1-
m_{22}	01------	--------	--------	-0----10

Sufficient conditions of chaining variables *a*

chaining variable	31 - 24	23 - 16	15 - 8	8 - 0
a_0	01100111	01000101	00100011	00000001
a_1	101-----	--------	--------	-1-a10aa
a_2	01100---	------0-	----a---	1--00010
a_3	0010----	-10---1a	------0-	0a-1a0-0
a_4	11010---	-01-----	01aaa---	0-10-100
a_5	10-01a--	-1-01-aa	--00100-	0---01-1
a_6	11--0110	-a-1001-	01100010	1-a111-1
a_7	-1--1110	a1a1111-	-101-001	1---0-10
a_8	-0----10	0000000a	a001a1--	100-0-1-
a_9	00------	11000100	00000000	101-1-1-
a_{10}	0-1-----	11111011	11100000	00--0-1-
a_{11}	1-0-----	-------1	01111110	11----0-
a_{12}	0-1-----	--------	--------	-1--a---
a_{13}	1-0-----	--------	--------	-1---01-
a_{14}	1-------	--------	--------	-1---1--
a_{15}	0-------	--------	--------	----0--0
a_{16}	-1------	--------	--------	----a---
a_{17}	-0------	--------	--------	----1-0-
a_{18}	1-1-----	--------	--------	----a-0-
a_{19}	--------	--------	--------	-------0
a_{20}	-C------	--------	--------	----A---
a_{21}	--------	--------	--------	----a-1-
[illegible]	--------	--------	--------	[illegible]

message variable	31 - 24	23 - 16	15 - 8	8 - 0
m_0	--0-----	--------	--------	--------
m_1	-01-----	--------	--------	--01--1-
m_2	-10-----	--------	--------	-1----11
m_3	--0-----	--------	--------	-1------
m_4	000-----	--------	--------	-0----1-
m_5	-11-----	--------	--------	------1-
m_6	0-------	--------	--------	-------0
m_7	--------	--------	--------	--1-----
m_8	--------	--------	--------	------00
m_9	-0------	--------	--------	-0-1--1-
m_{10}	-0------	--------	--------	-0------
m_{11}	101-----	--------	--------	-1-1--1-
m_{12}	1-1-----	--------	--------	--------
m_{13}	0-------	--------	--------	-0------
m_{14}	--0-----	--------	--------	-------0
m_{15}	--0-----	--------	--------	-11-----
m_{16}	0-------	--------	--------	-------0
m_{17}	-0------	--------	--------	-1----0-
m_{18}	00------	--------	--------	-1----01
m_{19}	-0------	--------	--------	--1---1-
m_{20}	--------	--------	--------	------11
m_{21}	-0------	--------	--------	-0----1-
m_{22}	01------	--------	--------	-0----10
m_{23}	11------	--------	--------	--1---0-
m_{24}	--------	--------	--------	-------0
m_{25}	-1------	--------	--------	------1-
m_{26}	10------	--------	--------	-0----10
m_{27}	-1------	--------	--------	-01---0-
m_{28}	1-------	--------	--------	-------0
m_{29}	-1------	--------	--------	-1----0-
m_{30}	-0------	--------	--------	-1----0-
m_{31}	-1------	--------	--------	------0-
m_{32}	--------	--------	--------	------1-
m_{33}	--------	--------	--------	-0------
m_{34}	0-------	--------	--------	------1-
m_{35}	0-------	--------	--------	--------
m_{36}	1-------	--------	--------	------1-
m_{37}	1-------	--------	--------	-0------
m_{38}	--------	--------	--------	--------
m_{39}	0-------	--------	--------	-1------
m_{40}	1-------	--------	--------	--------
m_{41}	--------	--------	--------	-1------
m_{42}	1-------	--------	--------	--------
m_{43}	--------	--------	--------	-1------
m_{44}	1-------	--------	--------	------1-
m_{45}	--------	--------	--------	--------
m_{46}	1-------	--------	--------	--------
m_{47}	0-------	--------	--------	--------
m_i $(i \geq 48)$	--------	--------	--------	--------

chaining variable	31 - 24	23 - 16	15 - 8	8 - 0
a_0	01100111	01000101	00100011	00000001
a_1	101-----	--------	--------	-1-a10aa
a_2	01100---	------0-	----a---	1--00010
a_3	0010----	-10---1a	------0-	0a-1a0-0
a_4	11010---	-01-----	01aaa---	0-10-100
a_5	10-01a--	-1-01-aa	--00100-	0---01-1
a_6	11--0110	-a-1001-	01100010	1-a111-1
a_7	-1--1110	a1a1111-	-101-001	1---0-10
a_8	-0----10	0000000a	a001a1--	100-0-1-
a_9	00------	11000100	00000000	101-1-1-
a_{10}	0-1-----	11111011	11100000	00--0-1-
a_{11}	1-0-----	-------1	01111110	11----0-
a_{12}	0-1-----	--------	--------	-1--a---
a_{13}	1-0-----	--------	--------	-1---01-
a_{14}	1-------	--------	--------	-1---1--
a_{15}	0-------	--------	--------	----0--0
a_{16}	-1------	--------	--------	----a---
a_{17}	-0------	--------	--------	----100-
a_{18}	1-1-----	--------	--------	-----00-
a_{19}	--------	--------	--------	-------0
a_{20}	-C------	--------	--------	----A---
a_{21}	-b------	--------	--------	----a-1-
a_{22}	--------	--------	--------	-----A1-
a_{23}	--------	--------	--------	-------0
a_{24}	-c------	--------	--------	--------
a_{25}	-B------	--------	--------	----a---
a_{26}	--------	--------	--------	-----A1-
a_{27}	--------	--------	--------	-------1
a_{28}	-c------	--------	--------	----A---
a_{29}	-B------	--------	--------	----A-0-
a_{30}	--------	--------	--------	------0-
a_{31}	--------	--------	--------	--------
a_{32}	--------	--------	--------	----A---
a_{33}	--------	--------	--------	------1-
a_{34}	--------	--------	--------	--------
a_{35}	--------	--------	--------	--------
a_{36}	--------	--------	--------	----A---
a_{37}	--------	--------	--------	------1-
a_{38}	--------	--------	--------	----A---
a_{39}	B-------	--------	--------	------0-
a_{40}	C-------	--------	--------	----A---
a_{41}	B-------	--------	--------	------0-
a_{42}	C-------	--------	--------	----A---
a_{43}	B-------	--------	--------	------0-
a_{44}	C-------	--------	--------	--------
a_{45}	B-------	--------	--------	--------
a_i $(i \geq 46)$	--------	--------	--------	--------

Table 3. Sufficient condition on $\{m_{ij}\}$ and $\{a_{i,j}\}$ of 58-round SHA-1

Notation

In Table 2, 3

- ‘*a*’: $a_{i,j} = a_{i-1,j}$
- ‘A’: $a_{i,j} = a_{i-1,j} + 1$
- ‘b’: $a_{i,j} = a_{i-1,(j+2)\mathrm{mod}\ 32}$
- ‘B’: $a_{i,j} = a_{i-1,(j+2)\mathrm{mod}\ 32} + 1$
- ‘c’: $a_{i,j} = a_{i-2,(j+2)\mathrm{mod}\ 32}$
- ‘C’: $a_{i,j} = a_{i-2,(j+2)\mathrm{mod}\ 32} + 1$

Two message modification techniques

- Modification of *a*
  - Decode as codes defined by *a*
- Modification of *m*
  - Decode as codes defined on *m*
- We use modification of *a*

Relations in 0-15-round of *m*

- All conditions on 0-57-round of *m* can be rewritten by 0-15-round relations
  - Using the relations derived of key expansion

    $m_i = (m_{i\text{-}3} \oplus m_{i\text{-}8} \oplus m_{i\text{-}14} \oplus m_{i\text{-}16}) <<< 1$
  - Using Gaussian elimination
  - Introduce elimination order of $\{m_{i,j}\}$ $\{i = 0,1,\ldots,15,\ j = 0,1,\ldots,31\}$ by

    $m'_{i',j'} \leq m'_{i',j'}$ if $i' \leq i$ or ($i' = i$ and $j' \leq j$)

Relation of 0-15-round of *m*

$m_{15,31} = 1, m_{15,30} = 1, m_{15,29} = 0, m_{15,28} + m_{10,28} + m_{8,29} + m_{7,29} + m_{4,28} + m_{2,28} = 1, m_{15,27} + m_{14,25} + m_{12,28} + m_{12,26} + m_{10,28} + m_{9,27} + m_{9,25} + m_{8,29} + m_{8,28} + m_{7,28} + m_{7,27} + m_{6,26} + m_{5,28} + m_{4,26} + m_{3,25} + m_{2,28} + m_{1,25} + m_{0,28} = 1, m_{15,26} + m_{10,28} + m_{10,26} + m_{8,28} + m_{8,27} + m_{7,27} + m_{6,29} + m_{5,27} + m_{4,26} + m_{2,27} + m_{2,26} + m_{0,27} = 1, m_{15,25} + m_{11,28} + m_{10,27} + m_{10,25} + m_{9,28} + m_{8,27} + m_{8,26} + m_{7,26} + m_{6,29} + m_{6,28} + m_{5,26} + m_{4,25} + m_{3,28} + m_{2,28} + m_{2,26} + m_{2,25} + m_{1,28} + m_{0,28} + m_{0,26} = 0, m_{15,24} + m_{12,28} + m_{11,27} + m_{10,26} + m_{10,24} + m_{9,28} + m_{9,27} + m_{8,29} + m_{8,26} + m_{8,25} + m_{7,25} + m_{6,29} + m_{6,28} + m_{6,27} + m_{5,25} + m_{4,28} + m_{4,24} + m_{3,28} + m_{3,27} + m_{2,27} + m_{2,25} + m_{2,24} + m_{1,28} + m_{1,27} + m_{0,27} + m_{0,25} = 1, m_{15,23} + m_{12,28} + m_{12,27} + m_{11,26} + m_{10,25} + m_{10,23} + m_{9,27} + m_{9,26} + m_{8,28} + m_{8,25} + m_{8,24} + m_{7,29} + m_{7,24} + m_{6,28} + m_{6,27} + m_{6,26} + m_{5,24} + m_{4,27} + m_{4,23} + m_{3,27} + m_{3,26} + m_{2,26} + m_{2,24} + m_{2,23} + m_{1,27} + m_{1,26} + m_{0,26} + m_{0,24} = 1, m_{15,22} + m_{14,25} + m_{12,28} + m_{12,27} + m_{11,25} + m_{10,27} + m_{10,24} + m_{10,22} + m_{9,28} + m_{9,27} + m_{9,26} + m_{8,27} + m_{8,24} + m_{8,23} + m_{7,28} + m_{7,27} + m_{7,23} + m_{6,27} + m_{6,25} + m_{5,23} + m_{4,28} + m_{4,27} + m_{4,22} + m_{3,26} + m_{2,28} + m_{2,27} + m_{2,25} + m_{2,23} + m_{2,22} + m_{1,26} + m_{0,25} + m_{0,23} = 0, m_{15,6} = 1, m_{15,5} = 1, m_{15,4} + m_{12,5} + m_{10,4} + m_{4,5} + m_{4,4} + m_{2,5} + m_{2,4} =$

Advanced sufficient conditions of message

message variable	31 - 24	23 - 16	15 - 8	8 - 0
m_0	--0-----	--------	--------	--------
m_1	-01-----	--------	--------	--01--1-
m_2	L10-----	--------	--------	-1----11
m_3	-L0-----	--------	--------	-1------
m_4	000-----	--------	--------	-0----1-
m_5	L11-----	--------	--------	------1L
m_6	0L------	--------	--------	-------0
m_7	LL------	--------	--------	--1----L
m_8	LL------	--------	--------	------00
m_9	L0L-----	--------	--------	-0L1--1L
m_{10}	L0L-----	--------	--------	-0L----L
m_{11}	101-----	--------	--------	-1-1--1L
m_{12}	1L1-----	--------	--------	-------L
m_{13}	0LLLLL-L	LL------	--------	-0LLLLLL
m_{14}	LL0LLL-L	LLLL----	--------	--LLLLL0
m_{15}	LL0LLLLL	LL------	--------	-11LLLLL
m_{16}	0-------	--------	--------	-------0
m_{17}	-0------	--------	--------	-1----0-
m_{18}	00------	--------	--------	-1----01
m_{19}	-0------	--------	--------	--1---1-
m_{20}	--------	--------	--------	------11
m_{21}	-0------	--------	--------	-0----1-
m_{22}	01------	--------	--------	-0----10

Control sequence (I)

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}	$a_{16,22}$	$m_{15,22} + m_{14,25} + m_{12,28} + m_{12,27} + m_{11,25}$ $+m_{10,27} + m_{10,24} + m_{10,22} + m_{9,28} + m_{9,27}$ $+m_{9,26} + m_{8,27} + m_{8,24} + m_{8,23} + m_{7,28} + m_{7,27}$ $+m_{7,23} + m_{6,27} + m_{6,25} + m_{5,23} + m_{4,28} + m_{4,27}$ $+m_{4,22} + m_{3,26} + m_{2,28} + m_{2,27} + m_{2,25} + m_{2,23}$ $+m_{2,22} + m_{1,26} + m_{0,25} + m_{0,23} = 0$
s_{111}	$a_{16,21}$	$a_{18,31} = 1$
[illegible]	[illegible]	[illegible]

Control Sequence (II)

Control sequence s_i	Control bit b_i	Controlled relation r_i
s_{82}	$a_{14,30}$	$m_{14,3} + m_{11,3} + m_{11,2}$ $+m_{8,2} + m_{7,4} + m_{7,2} + m_{7,1} + m_{6,2} + m_{5,3}$ $+m_{4,0} + m_{3,3} + m_{2,2} + m_{1,31} + m_{1,3} = 0$
s_{81}	$a_{15,2}$	$m_{14,2} + m_{12,5} + m_{12,3} + m_{10,4} + m_{9,2} + m_{7,4}$ $+m_{6,3} + m_{4,5} + m_{4,4} + m_{4,3} + m_{3,2} + m_{2,5}$ $+m_{2,4} + m_{1,2} = 1$
s_{80}	$a_{15,1}$	$m_{14,1} + m_{12,4} + m_{11,2} + m_{10,2} + m_{9,3} + m_{8,3}$ $+m_{7,2} + m_{6,2} + m_{5,5} + m_{5,2} + m_{4,4} + m_{3,31}$ $+m_{3,4} + m_{3,2} + m_{3,1} + m_{2,4} + m_{2,3} + m_{0,3} = 0$
s_{79}	$a_{14,27}$	$m_{14,0} = 0$
s_{78}	$a_{13,26}$	$m_{13,31} = 0$
s_{77}	$a_{13,25}$	$m_{13,30} = 0$
s_{76}	$a_{14,29}$	$m_{13,29} + m_{8,29} = 0$
s_{75}	$a_{14,28}$	$m_{13,28} + m_{8,28} + m_{2,28} + m_{0,28} = 0$
s_{74}	$a_{13,22}$	$m_{13,27} + m_{11,28} + m_{8,29} + m_{8,27} + m_{6,29}$ $+m_{5,28} + m_{3,28} + m_{2,27} + m_{0,27} = 1$
s_{73}	$a_{13,21}$	$m_{13,26} + m_{11,27} + m_{9,28} + m_{8,28} + m_{8,26}$ $+m_{6,28} + m_{5,27} + m_{3,28} + m_{3,27} + m_{2,26}$ $+m_{1,28} + m_{0,26} = 1$
s_{72}	$a_{14,24}$	$m_{13,24} + m_{12,28} + m_{11,27} + m_{11,25} + m_{10,28}$ $+m_{9,27} + m_{9,26} + m_{8,29} + m_{8,26} + m_{8,24}$ $+m_{7,29} + m_{7,28} + m_{6,26} + m_{5,25} + m_{4,28}$ $+m_{3,28} + m_{3,26} + m_{3,25} + m_{2,28} + m_{2,24}$ $+m_{1,28} + m_{1,26} + m_{0,24} = 0$
s_{71}	$a_{14,23}$	$m_{13,23} + m_{12,27} + m_{11,26} + m_{11,24} + m_{10,28}$ $+m_{10,27} + m_{9,26} + m_{9,25} + m_{8,29} + m_{8,28}$ $+m_{8,25} + m_{8,23} + m_{7,29} + m_{7,28} + m_{7,27}$ $+m_{6,25} + m_{5,28} + m_{5,24} + m_{4,28} + m_{4,27}$

Control Sequence (III)

Control sequence s_i	Control bit b_i	Controlled relation r_i
s_{22}	$a_{5,25}$	$m_{5,30} = 1$
s_{21}	$a_{6,29}$	$m_{5,29} = 1$
s_{20}	$a_{6,1}$	$m_{5,1} = 1$
s_{19}	$a_{3,27}$	$m_{5,0} + m_{3,0} + m_{1,31} = 1$
s_{18}	$a_{4,26}$	$m_{4,31} = 0$
s_{17}	$a_{4,25}$	$m_{4,30} = 0$
s_{16}	$a_{5,29}$	$m_{4,29} = 0$
s_{15}	$a_{5,6}$	$m_{4,6} = 0$
s_{14}	$a_{5,1}$	$m_{4,1} = 1$
s_{13}	$a_{3,25}$	$m_{3,30} = 1$
s_{12}	$a_{3,24}$	$m_{3,29} = 0$
s_{11}	$a_{4,6}$	$m_{3,6} = 1$
s_{10}	$a_{2,26}$	$m_{2,31} = 0$
s_9	$a_{2,25}$	$m_{2,30} = 1$
s_8	$a_{2,24}$	$m_{2,29} = 0$
s_7	$a_{3,5}$	$m_{2,6} = 1$
s_6	$a_{2,6}$	$m_{2,6} = 1$
s_5	$a_{3,1}$	$m_{2,1} = 1$
s_4	$a_{2,5}$	$m_{1,5} = 0$
s_3	$a_{1,28}$	$m_{1,1} = 1$
s_2	$a_{1,25}$	$m_{1,30} = 0$
s_1	$a_{1,24}$	$m_{1,29} = 1$
s_0	$a_{1,23}$	$m_{1,29} = 1$

Table 6 Control bit and controlled relations of 58-round SHA-1 (III)

Improvement of Message Modification technique

- Success probability is not 1
  - Control sequences sometimes rotate and do not end
  - Changing control bits may not affect leading term properly
- New method
  - Multiple control bits
    - Use iterative decoding technique
    - Use list decoding technique
  - Controlling non-leading terms

Advanced sufficient conditions of chaining variables *a*

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	00XVrrvV	11000100	00000000	101-1-1y
a_{10}	0w1-rv-v	11111011	11100000	00hW0-1r
a_{11}	1w0--V-V	-------1	01111110	11x---0Y
a_{12}	0w1-rV-V	--------	--------	-1XWa-Wh
a_{13}	1w0--vv-	-rr-----	--------	-1---01y
a_{14}	1rhhvvVh	hh------	--------	-1hhh1hh
a_{15}	0rwhhhVh	hhhh----	--------	--hh0hh0
a_{16}	W1whhhhh	hhq-q-q-	q--q-qqq	-WWhahhh
a_{17}	-0------	--------	--------	----1-0-
a_{18}	1-1-----	--------	--------	------0-
a_{19}	--------	--------	--------	-------0
a_{20}	--------	--------	--------	--------
a_{21}	--------	--------	--------	------1-

Advanced 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	00XVrrvV	11000100	00000000	101-1-1y
a_{10}	0w1-rv-v	11111011	11100000	00hW0-1r
a_{11}	1w0--V-V	-------1	01111110	11x---0Y
a_{12}	0w1-rV-V	--------	--------	-1XWa-Wh
a_{13}	1w0--vv-	-rr-----	--------	-1---01y
a_{14}	1rhhvvVh	hh------	--------	-1hhh1hh
a_{15}	0rwhhhVh	hhhh----	--------	--hh0hh0
a_{16}	W1whhhhh	hhq-q-q-	q--q-qqq	-WWhahhh
a_{17}	-0------	--------	--------	----1-0-
a_{18}	1-1-----	--------	--------	------0-
a_{19}	--------	--------	--------	-------0
a_{20}	--------	--------	--------	--------
a_{21}	--------	--------	--------	------1-

1, 0, a: Wang's sufficient conditions
w: adjust $a_{i+1,j}$ so as $m_{i,j} = 0$
W: adjust $a_{i+1,j}$ so as$m_{i,j} = 1$
v: adjust $a_{i,j-5}$ so as $m_{i,j} = 0$
V: adjust $a_{i,j-5}$ so as $m_{i,j} = 1$
…

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

message variable	31 - 24	23 - 16	15 - 8	8 - 0
m_0	--0-----	--------	--------	--------
m_1	-01-----	--------	--------	--01--1-
m_2	L10-----	--------	--------	-1----11
m_3	-L0-----	--------	--------	-1------
m_4	000-----	--------	--------	-0----1-
m_5	L11-----	--------	--------	------1L
m_6	0L------	--------	--------	-------0
m_7	LL------	--------	--------	--1----L
m_8	LL------	--------	--------	------00
m_9	L0L-----	--------	--------	-0L1--1L
m_{10}	L0L-----	--------	--------	-0L----L
m_{11}	101-----	--------	--------	-1-1--1L
m_{12}	1L1-----	--------	--------	-------L
m_{13}	0LLLLL-L	LL------	--------	-0LLLLLL
m_{14}	LL0LLL-L	LLLL----	--------	--LLLLL0
m_{15}	LL0LLLLL	LL------	--------	-11LLLLL
m_{16}	0-------	--------	--------	-------0
m_{17}	-0------	--------	--------	-1----0-
m_{18}	00------	--------	--------	-1----01
m_{19}	-0------	--------	--------	--1---1-
m_{20}	--------	--------	--------	------11
m_{21}	-0------	--------	--------	-0----1-
m_{22}	01------	--------	--------	-0----10
m_{23}	11------	--------	--------	--1---0-
m_{24}	--------	--------	--------	-------0
m_{25}	-1------	--------	--------	------1-
m_{26}	10------	--------	--------	-0----10
m_{27}	-1------	--------	--------	-01---0-
m_{28}	1-------	--------	--------	-------0
m_{29}	-1------	--------	--------	-1----0-
m_{30}	-0------	--------	--------	-1----0-
m_{31}	-1------	--------	--------	------0-
m_{32}	--------	--------	--------	------1-
m_{33}	--------	--------	--------	-0------
m_{34}	0-------	--------	--------	------1-
m_{35}	0-------	--------	--------	--------
m_{36}	1-------	--------	--------	------1-
m_{37}	1-------	--------	--------	-0------
m_{38}	--------	--------	--------	--------
m_{39}	0-------	--------	--------	-1------
m_{40}	1-------	--------	--------	--------
m_{41}	--------	--------	--------	-1------
m_{42}	1-------	--------	--------	--------
m_{43}	--------	--------	--------	-1------
m_{44}	1-------	--------	--------	------1-
m_{45}	--------	--------	--------	--------
m_{46}	1-------	--------	--------	--------
m_{47}	0-------	--------	--------	--------
m_i $(i \geq 48)$	--------	--------	--------	--------

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
a_{20}	-C------	--------	--------	----A---
a_{21}	-b------	--------	--------	----a-1-
a_{22}	--------	--------	--------	-----A1-
a_{23}	--------	--------	--------	-------0
a_{24}	-c------	--------	--------	--------
a_{25}	-B------	--------	--------	----a---
a_{26}	--------	--------	--------	-----A1-
a_{27}	--------	--------	--------	-------1
a_{28}	-c------	--------	--------	----A---
a_{29}	-B------	--------	--------	----A-0-
a_{30}	--------	--------	--------	------0-
a_{31}	--------	--------	--------	--------
a_{32}	--------	--------	--------	----A---
a_{33}	--------	--------	--------	------1-
a_{34}	--------	--------	--------	--------
a_{35}	--------	--------	--------	--------
a_{36}	--------	--------	--------	----A---
a_{37}	--------	--------	--------	------1-
a_{38}	--------	--------	--------	----A---
a_{39}	B-------	--------	--------	------0-
a_{40}	C-------	--------	--------	----A---
a_{41}	B-------	--------	--------	------0-
a_{42}	C-------	--------	--------	----A---
a_{43}	B-------	--------	--------	------0-
a_{44}	C-------	--------	--------	--------
a_{45}	B-------	--------	--------	--------
a_i $(i \geq 46)$	--------	--------	--------	--------

Table 6. 'Advanced' sufficient condition on $\{m_{i,j}\}$ and $\{a_{i,j}\}$

Notation

In Table 6,

- ‘w’: adjust $a_{i,j}$ so that $m_{i+1,j} = 0$
- ‘W’: adjust $a_{i,j}$ so that $m_{i+1,j} = 1$
- ‘v’: adjust $a_{i,j}$ so that $m_{i,(j+27)\mathrm{mod}32} = 0$
- ‘V’: adjust $a_{i,j}$ so that $m_{i,(j+27)\mathrm{mod}32} = 1$
- ‘h’: adjust $a_{i,j}$ so that corresponding controlled relation including $m_{i+1,\mathrm{j}}$ as leading term holds
- ‘r’: adjust $a_{i,j}$ so that corresponding controlled relation including $m_{i,(j+27)\mathrm{mod}32}$ as leading term holds

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

Algorithm 1

Algorithm 1 (Basic Message Modification) *Procedures for message modification: Preset the maximal number of trials M.*

1. *Set* $r = 0$.
2. *Generate* $(a_1, a_2, \cdots, a_{16})$ *randomly.*
3. *Set* $i = 0$.
4. *Increment* i *until the controlled relation* r_i *of* s_i *is not satisfied. If all relations are satisfied go to final step. If* $r > M$, *give up and return to Step 2.*
5. *Adjust control bits* $a_{i,j}$ *of* s_i *so that corresponding controlled relation and sufficient condition on* $\{a_{i,j}\}$ *hold. After adjusting, set* $i = 0$ *and* $r = r + 1$ *and go to Step 3 and repeat the process until all controlled relations hold.*
6. *If all controlled relations are satisfied, check whether modified message yields collision or not. If it does not generate collision, return to Step 2. If it generates collision, finish.*

Algorithm 2

Algorithm 2 (Improved Message Modification) *Procedures for message:*

1. *Generate* $(a_1, a_2, \cdots, a_{16})$ *randomly.*
2. *Using the basic message modification described in Algorithm 1, modify* $(a_1, a_2, \cdots, a_{16})$ *so that all message conditions and some chaining variable conditions from the 17-th round to the 23-rd round hold. If this step fails, return to Step 1.*
3. *If remaining changing variable conditions from the 17-th round to the 23-th round are not satisfied, return to Step 1 and repair until all conditions are satisfied (It can be satisfied probabilistically).*
4. *Change values of semi-neutral bits and modify chaining variables using our control sequence, and check whether chaining variable conditions from the 24-th round to the final round are satisfied.*
5. *Repeat all procedure above until all chaining variable conditions are satisfied.*

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

Algorithm 3

Algorithm 3 *Procedures for message modification: Preset the maximal number of trials M.*

1. *Set $r = 0$.*
2. *Generate $(a_1, a_2, \cdots, a_{16}) \in (\mathbb{F}_2^{32})^{16}$ randomly.*
3. *Set $i = 0$.*
4. *Increment i until $f_i \not\equiv 0 \mod I$. If all f_i are contained in I, go to the final step. If $r > M$, give up and return to Step 2.*
5. *For control polynomials $\{g_{j,l}\}$ associated to f_i, replace appropriate $g_{j,l}(X_{j,l})$ by $g_{j,l}(X_{j,l} + 1)$ in I to satisfy $f_i \equiv 0 \mod I$. After adjusting, set $r = r + 1$ and go to Step 3.*
6. *Solve a system of polynomial equations in R_2 consists of all equations with respect to advanced sufficient conditions on $\{a_{i,j}\}$ by using Gröbner basis algorithm.*
7. *Check whether modified message yields collision or not. If it does not generate collision, return to Step 2. If it generates collision, finish.*

In the case of full round SHA-1

- Success probability of message modification is smaller?
  - Control bits are insufficient
  - Success probability is very small?
- No semi-neutral bit remained?
- Complexity is 2^{63} message modification, not 2^{63} SHA-1
  - Message modification is too heavy？
- Message modification can be improved?

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	$07ff8000$
$i = 8$	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
$i = 23$	00000003	00000003	00000000

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

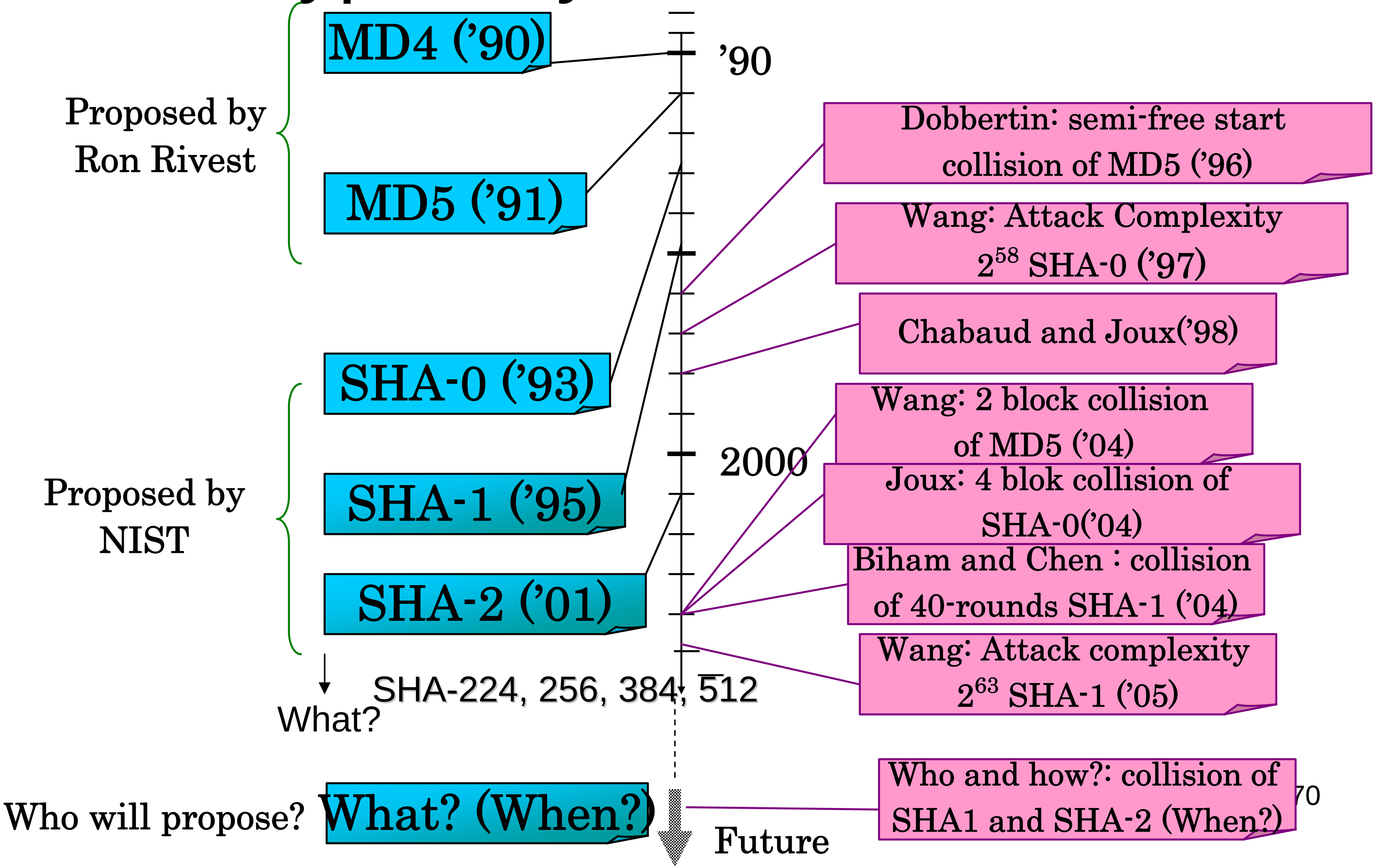
- The above M satisfies all message conditions of 0-80 rounds and all chaining variable conditions of 0-28 rounds

Summary of Part II

- Proposed the novel method for finding the differential pattern, method for determining sufficient conditions and the novel method for the message modification using Gröbner-like method
- Succeeded in finding collisions of 58-step SHA-1
  - Showed by experiments the efficiency of proposed method

Part III
Hash Functions:
What's the Future?".

A history of hash function proposals and cryptanalysis of hash functions



Hash functions in the future

- NIST admit to use SHA-1 for 5 years as it is
- NIST is considering SHA-256 as a replacement of SHA-1 and to be secure until 2015
- Timeline was published by NIST

Timeline published by NIST

- **Year 1 (2008?):**
  - 1Q Draft and publish the minimum acceptability requirements, evaluation criteria, and submission requirements for public comments. Announce a public workshop to discuss these requirements.
  - 2Q Public comment period ends.
  - 2Q Host a workshop to discuss these requirements.
  - 3Q Finalize and publish the minimum acceptability requirements, evaluation criteria and submission requirements. Request submissions for new hash algorithms.
- **Year 2 (2009?):**
  - 2Q Review submitted algorithms, and select candidates that meet basic submission requirements.
  - 3Q Host the First Hash Function Candidate Conference. A nnounce first round candidates
  - 3Q Call for public comments on the first round candidates.
- **Year 3 (2010?):**
  - 1Q Hold the Second Hash Function Candidate Conference. Discuss analysis results on the first round candidates.
  - 2Q Public comment period on the first round candidates ends.
  - 3Q Address public comments; select the second round finalists. Prepare a report to explain the selection.
  - 3Q Announce the second round finalists. Publish the selection report, and call for public comments on the second round candidates.
- **Year 4 (2011?):**
  - 2Q Host the Third Hash Function Candidate Conference. Submitters of the second round finalists discuss comments on their algorithms. 2QPublic comment period ends.
  - 3Q Address public comments, and select the finalist. Prepare a report to describe the final selection(s).
  - 4Q Announce the new hash function(s).
- **Year 5 (2012?):**
  - 1Q Publish draft standard for public comments.
  - 2Q Public comment period ends
  - 3Q Address public comments.
  - 4Q Publish new hash function standard.

What's the difficulty to find collision of 58-round reduced SHA-1?

- Wang found the collisions of 58-round
- Many researcher in the world failed to find similar collisions, why?
  - Wang does not publish all the details of her attack
  - Attack is essentially mathematical
  - Need the knowledge of Gröbner basis
  - Need the programming technique
    - Sometimes need super programmer
  - Need so many human resources
    - I spent 2000 hours to experiment and implement

What's the problem in standardization of hash function?

- No one could not implement Wang's attack of SHA-1 properly
  - Therefore no one can evaluate the complexity accurately
  - No one knows whether Wang's attack can be applicable to SHA-2 or not
  - No one can propose new algorithms immune to Wang's attack

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