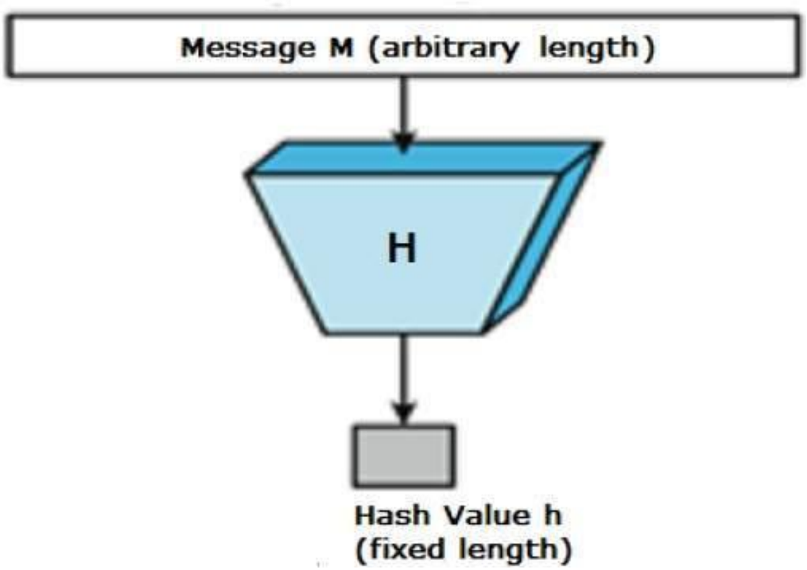
Collision Attack on SHA-1

Danish Malik, University of Minnesota Morris

What is a Hash function?

- Algorithm
- Input message of arbitrary bit-length
- Outputs fixed bit-length(hash value)



Collisions in Hash Functions

Collision occurs when two distinct inputs

x and y output to the same hash value z such that :

$h(x) = h(y) = z$

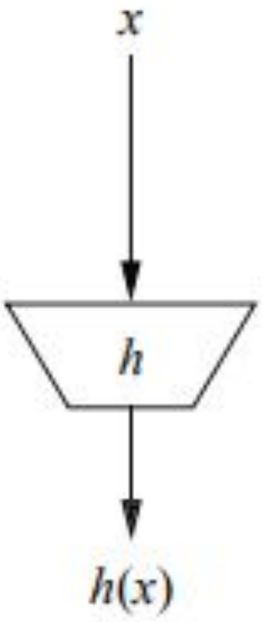
Properties of Hash Functions

- Preimage resistance (one-wayness).
- Second preimage resistance.
- Collision resistance.

Preimage Resistance (Property 1)

- One way

Given hash output z,
finding input x such that
z = h(x) should be infeasible.



preimage resistance

[1]

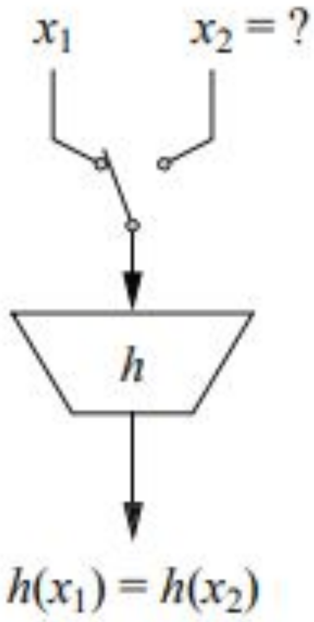
Second preimage resistance (Property 2)

Given input message x_1

finding x_2 such that

$h(x_1) = h(x_2)$

should be infeasible.



second preimage resistance

[1]

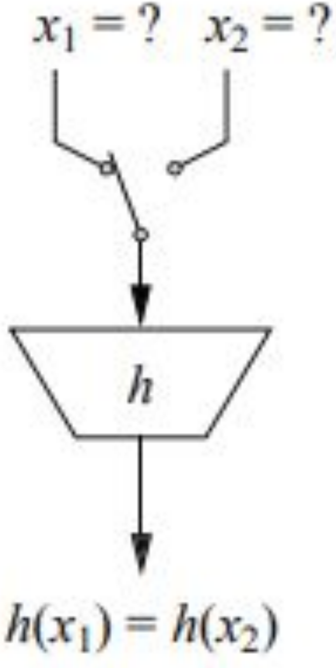
Collision resistance (Property 3)

Finding distinct input messages

x_1 and x_2 such that

that $h(x_1) = h(x_2)$

should be infeasible.



collision resistance

[1]

Why and Where are Hash functions used?

WHY:

- Data Integrity

WHERE:

- Digital Signature Schemes.

- Message Authentication Code (MAC).

- Other authentication protocols.

Table of contents

- Introduction and Background on SHA-1
- Compression Algorithm of SHA-1
- Attack Overview
- Local Collisions Using Disturbance Vector(DV)
- Disturbance Vector Selection
- Differential Path(DP)
- Attack using DV and DP
- Computation
- Conclusion

SHA-1

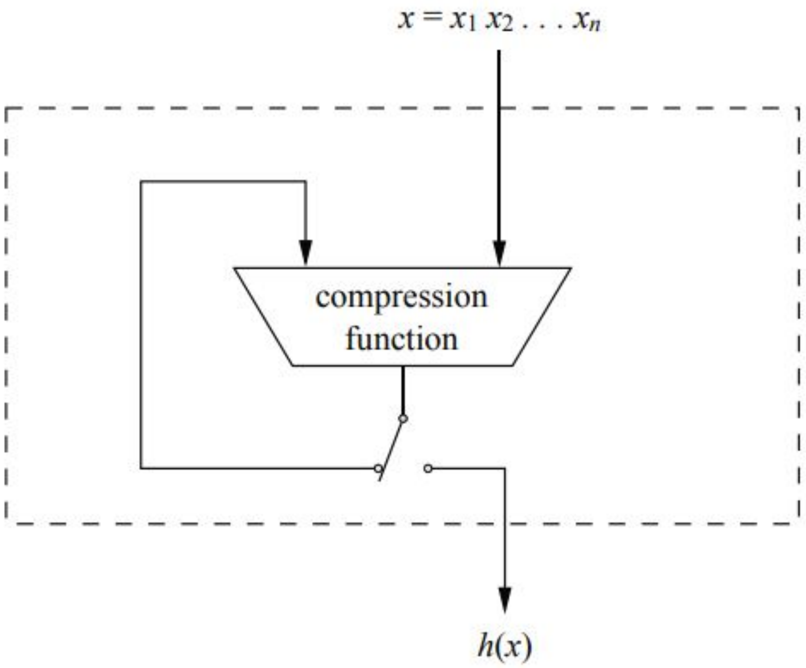
- Secure Hash Algorithm 1.
- Created in 1995 by NSA.
- Many theoretical attacks since 2005.
- Finally, broken by Google and CWI in 2017.

SHA-1 continued:

- Input of arbitrary bit-length.
- Outputs 160-bit hash value.
- Input is padded to obtain a multiple of 512 bits.

SHA-1's construction *(Merkle-Damgard)*

- After padding, input is segmented into 512-bit *message blocks* ($x_1,...,x_n$).
- Each message block (x_i) is fed into the *compression function.*
- Hash value *h(x)* is output after final iteration



[1]

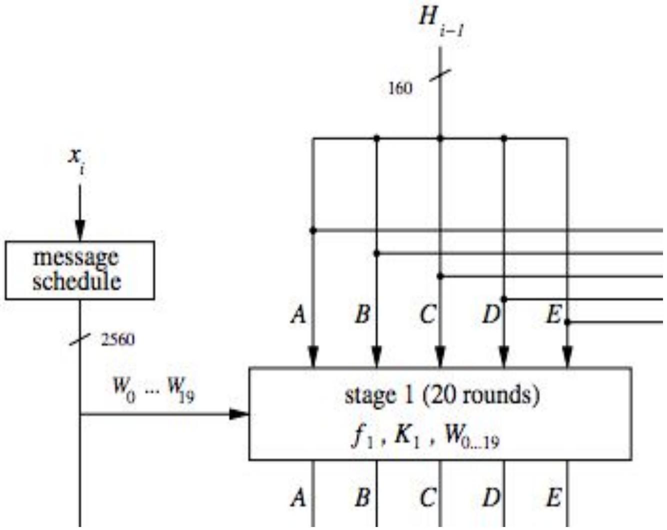
Compression Algorithm of SHA-1

- Message schedule expands the message blocks into eighty 32-bit strings $W_0, ..., W_{79}$ known as *message words.*

$$W_j = \begin{cases} x_i^{(j)} & 0 \leq j \leq 15 \\ (W_{j-16} \oplus W_{j-14} \oplus W_{j-8} \oplus W_{j-3}) \lll 1 & 16 \leq j \leq 79, \end{cases}$$

- The *chaining value* is initialized to a predefined value H_0 = *IV (initialization vector)*

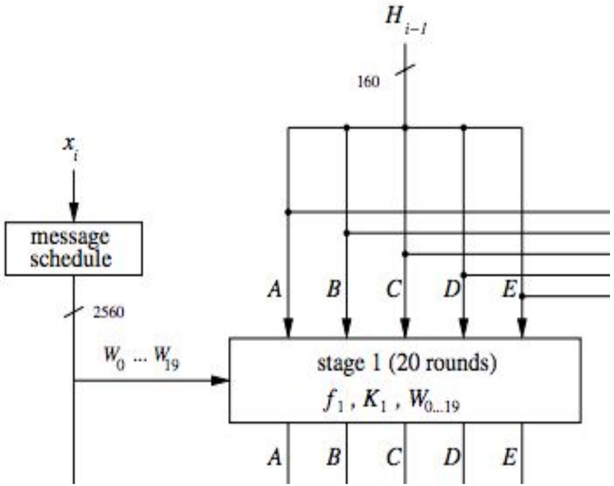
- Chaining value H_{i-1} is segmented into five 32-bit strings known as *state words* represented by letters A,B,C,D,E.



[1]

Compression Algorithm of SHA-1 (continued)

- There are 4 stages within the compression function.
- Each stage consists of 20 rounds.
- Each round updates the chaining value using a message word W_i such that : $H_i = Round(H_{i-1}, W_{i-1})$
- The final chaining value H_{80} is the hash output.



[1]

Compression Algorithm of SHA-1 (continued):

- Each stage use different bitwise boolean functions and constants.
- AND, OR, NOT
  XOR.

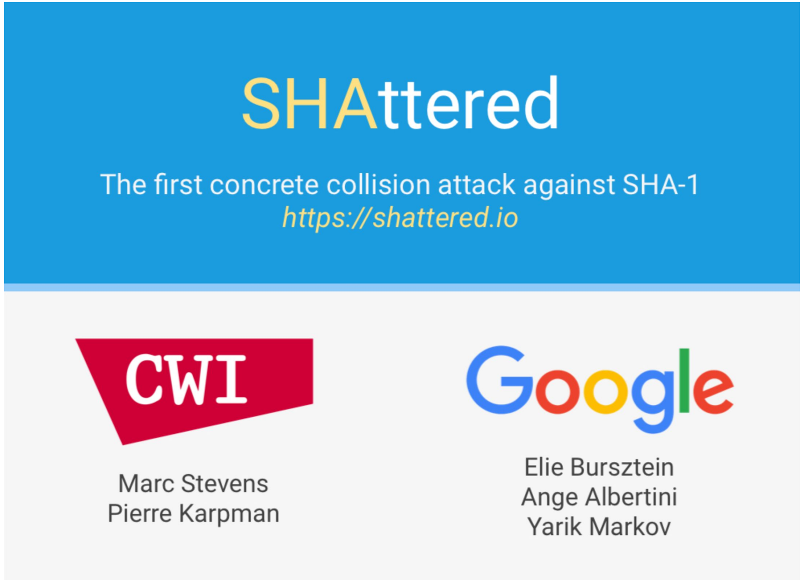
Stage t	Round j	Constant K_t	Function f_t
1	$0\ldots19$	$K_1 =$ 5A827999	$f_1(B,C,D) = (B \wedge C) \vee (\bar{B} \wedge D)$
2	$20\ldots39$	$K_2 =$ 6ED9EBA1	$f_2(B,C,D) = B \oplus C \oplus D$
3	$40\ldots59$	$K_3 =$ 8F1BBCDC	$f_3(B,C,D) = (B \wedge C) \vee (B \wedge D) \vee (C \wedge D)$
4	$60\ldots79$	$K_4 =$ CA62C1D6	$f_4(B,C,D) = B \oplus C \oplus D$

- A *local collision* occurs when the chaining values of two message words are equal such that :
  $A,B,C,D,E(W_i) = A,B,C,D,E(W_j)$

[1]

SHAttered Attack

PDF.1



PDF.2

SHA-1(PDF.1) = SHA-1(PDF.2)

[2]

Type of Attack

Chosen-prefix attack by constructing two files x and y : where,

$x = (P \quad M_1^{(x)} \quad M_2^{(x)})$ and $y = (P \quad M_1^{(y)} \quad M_2^{(y)})$

and

P is the identical prefix

$M_1^{(x)}$, $M_1^{(y)}$ is the first near-collision block pair

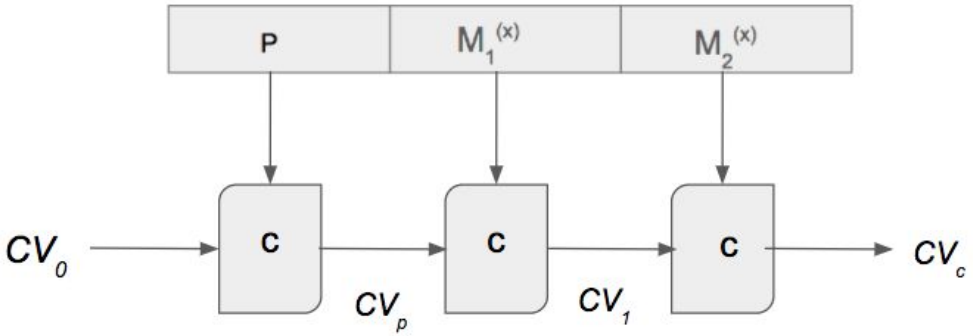
$M_2^{(x)}$, $M_2^{(y)}$ is the second near-collision block pair

such that x and y collide for any suffix S

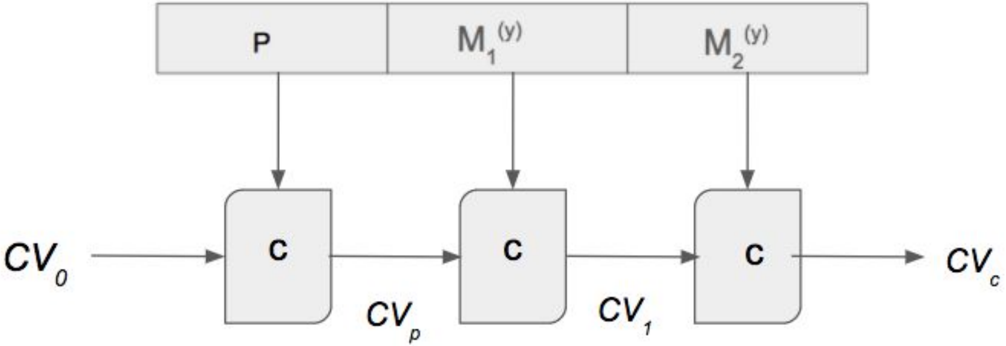
$$\text{SHA-1}(P \,||\, M_1^{(x)} \,||\, M_2^{(x)} \,||\, S) = \text{SHA-1}(P \,||\, M_1^{(y)} \,||\, M_2^{(y)} \,||\, S)$$

Attack Overview

Computation SHA-1(x) = CV_c



Computation SHA-1(y) = CV_c



Construction of the files x and y

$x = (\underline{P} \;\; M_1^{(x)} \;\; M_2^{(x)})$ and $y = (\underline{P} \;\; M_1^{(y)} \;\; M_2^{(y)})$

P in Hexidecimal

```
25 50 44 46 2d 31 2e 33 0a 25 e2 e3 cf d3 0a 0a | %PDF-1.3.%......
0a 31 20 30 20 6f 62 6a 0a 3c 3c 2f 57 69 64 74 | .1 0 obj.<</Widt
68 20 32 20 30 20 52 2f 48 65 69 67 68 74 20 33 | h 2 0 R/Height 3
20 30 20 52 2f 54 79 70 65 20 34 20 30 20 52 2f |  0 R/Type 4 0 R/
53 75 62 74 79 70 65 20 35 20 30 20 52 2f 46 69 | Subtype 5 0 R/Fi
6c 74 65 72 20 36 20 30 20 52 2f 43 6f 6c 6f 72 | lter 6 0 R/Color
53 70 61 63 65 20 37 20 30 20 52 2f 4c 65 6e 67 | Space 7 0 R/Leng
74 68 20 38 20 30 20 52 2f 42 69 74 73 50 65 72 | th 8 0 R/BitsPer
43 6f 6d 70 6f 6e 65 6e 74 20 38 3e 3e 0a 73 74 | Component 8>>.st
72 65 61 6d 0a ff d8 ff fe 00 24 53 48 41 2d 31 | ream......$SHA-1
20 69 73 20 64 65 61 64 21 21 21 21 21 85 2f ec |  is dead!!!!!./.
09 23 39 75 9c 39 b1 a1 c6 3c 4c 97 e1 ff fe 01 | .#9u.9...<L.....
```

$P = W_1,\ldots,W_{48}$ (48 message words) (1536 bits)

[2]

Construction of the files x and y (continued)

$x = (P \quad \underline{M}_1^{(x)} \quad \underline{M}_2^{(x)})$ and $y = (P \quad \underline{M}_1^{(y)} \quad \underline{M}_2^{(y)})$

$M_1^{(x)} = W_{49}, \ldots, W_{64}$

$M_2^{(x)} = W_{65}, \ldots, W_{80}$

First and Second near-collision blocks of x:

$M_1^{(1)}$	7f	46	dc	93	a6	b6	7e	01	3b	02	9a	aa	1d	b2	56	0b
	45	ca	67	d6	88	c7	f8	4b	8c	4c	79	1f	e0	2b	3d	f6
	14	f8	6d	b1	69	09	01	c5	6b	45	c1	53	0a	fe	df	b7
	60	38	e9	72	72	2f	e7	ad	72	8f	0e	49	04	e0	46	c2
$M_2^{(1)}$	30	57	0f	e9	d4	13	98	ab	e1	2e	f5	bc	94	2b	e3	35
	42	a4	80	2d	98	b5	d7	0f	2a	33	2e	c3	7f	ac	35	14
	e7	4d	dc	0f	2c	c1	a8	74	cd	0c	78	30	5a	21	56	64
	61	30	97	89	60	6b	d0	bf	3f	98	cd	a8	04	46	29	a1

$M_1^{(y)} = W_{49}, \ldots, W_{64}$

$M_2^{(y)} = W_{65}, \ldots, W_{80}$

First and Second near-collision blocks of y:

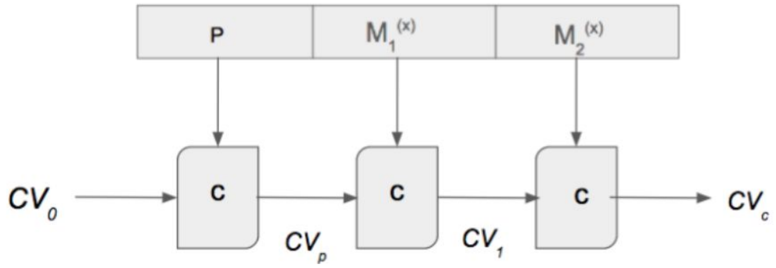
$M_1^{(2)}$	73	46	dc	91	66	b6	7e	11	8f	02	9a	b6	21	b2	56	0f
	f9	ca	67	cc	a8	c7	f8	5b	a8	4c	79	03	0c	2b	3d	e2
	18	f8	6d	b3	a9	09	01	d5	df	45	c1	4f	26	fe	df	b3
	dc	38	e9	6a	c2	2f	e7	bd	72	8f	0e	45	bc	e0	46	d2
$M_2^{(2)}$	3c	57	0f	eb	14	13	98	bb	55	2e	f5	a0	a8	2b	e3	31
	fe	a4	80	37	b8	b5	d7	1f	0e	33	2e	df	93	ac	35	00
	eb	4d	dc	0d	ec	c1	a8	64	79	0c	78	2c	76	21	56	60
	dd	30	97	91	d0	6b	d0	af	3f	98	cd	a4	bc	46	29	b1

[2]

Attack Overview

Computation SHA-1(*x*)

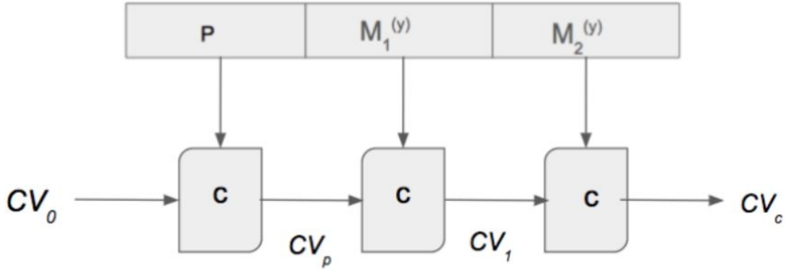
$x = W_{1} \ldots W_{48} \mid W_{49} \ldots W_{64} \mid W_{65} \ldots W_{80}$



CV_0	4e a9 62 69 7c 87 6e 26 74 d1 07 f0 fe c6 79 84 14 f5 bf 45
$M_1^{(2)}$	73 46 dc 91 66 b6 7e 11 8f 02 9a b6 21 b2 56 0f
	f9 ca 67 cc a8 c7 f8 5b a8 4c 79 03 0c 2b 3d e2
	18 f8 6d b3 a9 09 01 d5 df 45 c1 4f 26 fe df b3
	dc 38 e9 6a c2 2f e7 bd 72 8f 0e 45 bc e0 46 d2
$CV_1^{(2)}$	8d 64 c8 21 ff ed 52 e2 eb c8 59 15 5e c7 eb 36 73 8a 5a 7b
$M_2^{(2)}$	3c 57 0f eb 14 13 98 bb 55 2e f5 a0 a8 2b e3 31
	fe a4 80 37 b8 b5 d7 1f 0e 33 2e df 93 ac 35 00
	eb 4d dc 0d ec c1 a8 64 79 0c 78 2c 76 21 56 60
	dd 30 97 91 d0 6b d0 af 3f 98 cd a4 bc 46 29 b1
CV_2	1e ac b2 5e d5 97 0d 10 f1 73 69 63 57 71 bc 3a 17 b4 8a c5

Computation SHA-1(*y*)

$y = W_{1} \ldots W_{48} \mid W_{49} \ldots W_{64} \mid W_{65} \ldots W_{80}$

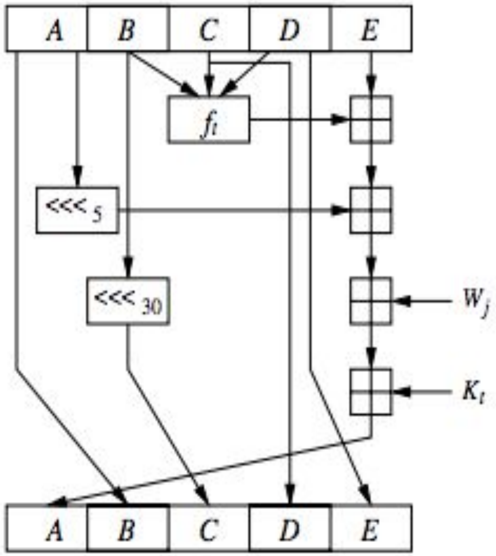


CV_0	4e a9 62 69 7c 87 6e 26 74 d1 07 f0 fe c6 79 84 14 f5 bf 45
$M_1^{(1)}$	7f 46 dc 93 a6 b6 7e 01 3b 02 9a aa 1d b2 56 0b
	45 ca 67 d6 88 c7 f8 4b 8c 4c 79 1f e0 2b 3d f6
	14 f8 6d b1 69 09 01 c5 6b 45 c1 53 0a fe df b7
	60 38 e9 72 72 2f e7 ad 72 8f 0e 49 04 e0 46 c2
$CV_1^{(1)}$	8d 64 d6 17 ff ed 53 52 eb c8 59 15 5e c7 eb 34 f3 8a 5a 7b
$M_2^{(1)}$	30 57 0f e9 d4 13 98 ab e1 2e f5 bc 94 2b e3 35
	42 a4 80 2d 98 b5 d7 0f 2a 33 2e c3 7f ac 35 14
	e7 4d dc 0f 2c c1 a8 74 cd 0c 78 30 5a 21 56 64
	61 30 97 89 60 6b d0 bf 3f 98 cd a8 04 46 29 a1
CV_2	1e ac b2 5e d5 97 0d 10 f1 73 69 63 57 71 bc 3a 17 b4 8a c5

[2]

Local Collisions using Disturbance Vector

- *Disturbance Vector* (DV) is a set of expanded message words that aim to cause a local collision.
- Every "1" bit of a DV marks the start of a local collision by creating a disturbance.
- The disturbance is then corrected within the next 5 rounds to obtain the same chaining value such that :
  $A,B,C,D,E(W_i) = A,B,C,D,E(W_{i+5})$



[1]

Disturbance Vector Selection

- The message expansion schedule is defined in two directions: *Forward and Backward expansion*.

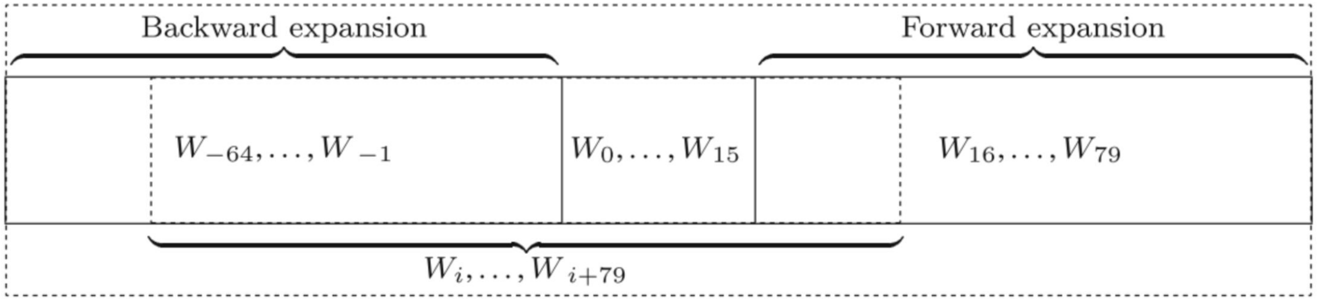
- The first 16 message words $W_0,...,W_{15}$ is known as the *information window.*

- These 16 message words can be expanded forward to obtain the remaining 64 message words $W_{16},...,W_{79}$ using the standard SHA-1 recursive equation.

- Similarly, we can expand those 16 message words backwards to obtain $W_{-64},...,W_{-1}$ using the recursive equation below :

$$W_i = (W_{i+16} \ggg 1) \oplus W_{i+13} \oplus W_{i+8} \oplus W_{i+2} \quad \text{for } -64 \leq i \leq -1$$

Disturbance Vector Selection (continued)



- For a given information window, we can construct 144 message words through forward and backward expansion.

- These 144 expanded message words $W_{-64},\ldots W_{-1}, W_0,\ldots,W_{15}, W_{16},\ldots,W_{79}$ is known as the *extended expanded message* (EEM).

- For each EEM, there are 65 valid expanded message words, each of which is a potential candidate as disturbance vector.

[3]

Differential Path (DP)

- *Differential Path* (DP) is a form of cryptanalysis that is the study of how differences in input can affect the resultant differences in output.

- Allows us to obtain a precise description of bit differences in state words and message words.

- Helps us understand how those differences should propagate over the 80 rounds of the compression function.

Collision Attack using DV and DP

- We can connect the chaining value differences with the local collision positions of a DV.

- This is achieved by constructing a non-linear differential path over the first 16 rounds that unite the chaining value bit differences to the “1” bit positions of the DV.

$$W_j = \begin{cases} x_i^{(j)} & 0 \leq j \leq 15 \\ (W_{j-16} \oplus W_{j-14} \oplus W_{j-8} \oplus W_{j-3}) \lll 1 & 16 \leq j \leq 79, \end{cases}$$

- After which we can determine a system of equations over the 80 rounds.

- The solution to these equations is in the form of a near-collision message block pair.

[1]

Computation of Attack

- Required 9,223,372,036,854,775,808 SHA-1 computations.
- 6,500 years of single-CPU computations.
- 110 years of single-GPU computations.
- 100,000 times faster than the Brute force attack.

Conclusion

- Certification Authority (CA) not allowed to issue SHA-1 certificates anymore.
- Websites protected by SHA-1 certificates are now considered insecure by both Chrome and Firefox.
- Don’t use SHA-1 anymore!
- Consider using better alternatives, SHA-256 or SHA-3.

References

[1] - Christof Paar Jan Pelzl. Understanding Cryptography: A Textbook for Students and Practitioners. Springer Publishing Company, 2009. ISBN: 3642041000 9783642010006

[2] - Marc Stevens Elie Bursztein Pierre Karpman Ange Albertini and Yarik Markov. “The first collision for full SHA-1”. In: Springer 14.3 (2017), pp. 1-7

[3] - Stephane Manuel. “Classification and generation of disturbance vectors for collision attacks against SHA-1”. In: Springer Science and Business Media, LLC 2010 14.3 (2011), pp. 250–251

Acknowledgements

Thank you for your time!

Special Thanks to Elena Machkasova for her guidance and feedback.