New Results on SHA-0 and SHA-1

Eli Biham

Department of Computer Science
Technion, Haifa 32000, Israel

Joint work with Rafi Chen

August 9, 2004

The SHA Family

SHA-0 was issued as a FIPS standard by NIST in 1993.

Designed by the NSA on the principles of Rivest's MD4 and MD5.

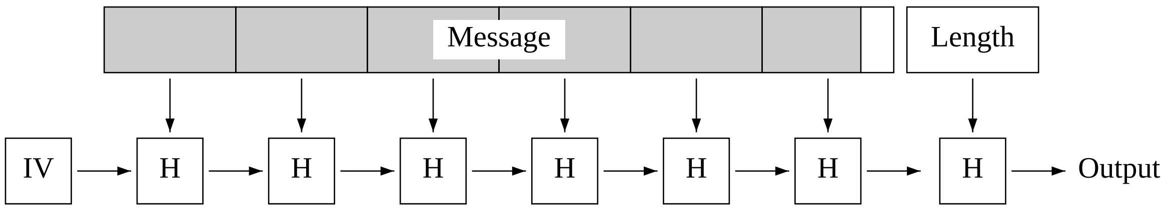
Outputs 160-bit hash values.

A technical revision, called SHA-1, specifies an additional rotate operation to the algorithm, replaced this standard.

Part I: Background

SHA-0 and SHA-1

Like in MD4 and most hash functions, the message is padded (where the padding include the length of the message), and then divided to 512-bit blocks, and a compression function is applied on the blocks iteratively:



The output of the last compression function becomes the output of the hash function.

The hash result is 160-bit long.

The Compression Functions of SHA-0 and SHA-1

1. Divide $X = M_i$ into 16 32-bit words: $W_0, W_1, W_2, \ldots, W_{15}$.

2. for $t = 16$ to 79 compute

$$W_t = (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) \underbrace{\lll 1}_{\uparrow \text{ Only in SHA-1}}.$$

3. Save A as AA, B as BB, C as CC, D as DD, and E as EE.

4. For $t = 0$ to 79 do

   (a) $T = A \lll 5 + f_t(B, C, D) + E + W_t + K_t$,

   (b) $E = D$, $D = C$, $C = B \lll 30$, $B = A$, $A = T$,

   where $X \lll Y$ (cyclicly) rotates X to the left by Y bits.

5. Perform $A = A + AA$, $B = B + BB$, $C = C + CC$, $D = D + DD$, and $E = E + EE$ (modulo 2^{32}).

The Compression Functions of SHA-0 and SHA-1 (cont.)

6. The function f_t and the values K_t used above are:

$$\begin{array}{rll} 0 \le t \le 19\colon & f_t(X,Y,Z) = XY \vee (\neg X)Z & K_t = \texttt{5A827999} \\ 20 \le t \le 39\colon & f_t(X,Y,Z) = X \oplus Y \oplus Z & K_t = \texttt{6ED9EBA1} \\ 40 \le t \le 59\colon & f_t(X,Y,Z) = XY \vee XZ \vee YZ & K_t = \texttt{8F1BBCDC} \\ 60 \le t \le 79\colon & f_t(X,Y,Z) = X \oplus Y \oplus Z & K_t = \texttt{CA62C1D6} \end{array}$$

Previous Works on the MD4/SHA Family

MD4: A collision (Dobbertin), many attacks on reduced variants

MD5: A pseudo-collision (Dobbertin)

These attacks use the fact the each message word (and each bit) is used only a few times by the message schedule.

In SHA-0 each message bit is used over 20 times.

Previous Works on the MD4/SHA Family (cont.)

SHA-0: In CRYPTO'98 Chabaud and Joux described an attack that can find a collision of SHA-0 in 2^{61} complexity.

They found a collision of 35-round reduced SHA-0.

SHA-1: No papers analyzing SHA-1 were published so far.

Chabaud and Joux' Attack – Observations

1. Observe that there is no mixing between columns (bits locations) in the W array: changing a bit in the message may affect only bits of the same columns in $W_{16}, \ldots, W_{79}$.

2. The avalanche effect in the main part of the compression function is due to two operations:

   (a) The rotation instructions cause each change of bit in W to affect bits 5 and 30 locations apart.

   (b) Carry bits of the addition operations.

   (c) Note that f_t is only used to increase non-linearity, but not to increase avalanche directly.

Chabaud and Joux' Attack – The Attack

1. Consider three column which are 5 and 30 bits apart (best are in bits 1, 6, 31 due to loss of the carry from bit 31)

2. Create a disturbance (difference) in bit 1 of some round

3. Correct the difference in the following rounds by modifying bits 1, 6, 31 of W in the disturbed locations

4. Elimination of the disturbance takes five rounds (due to the rotation of the value between the five registers)

5. Once all differences are eliminated, a collision is found

6. The process is probabilistic (correction is not always assured)

Chabaud and Joux' Attack – The Attack (cont.)

		Disturbance	Correction Rounds				
Round		i	$i+1$	$i+2$	$i+3$	$i+4$	$i+5$
Input	W^1	$0 \rightarrow 1$		$1 \rightarrow 0$			
Words	W^6		$1 \rightarrow 0$				
	W^{31}				$0 \leftrightarrow 1$	$0 \leftrightarrow 1$	$0 \leftrightarrow 1$
Desired	A^6		$0 \rightarrow 1$				
Results	f^1			$0 \rightarrow 1$			
	f^{31}				$0 \leftrightarrow 1$	$0 \leftrightarrow 1$	
	E^{31}						$0 \leftrightarrow 1$
Registers		A^1	B^1	C^{31}	D^{31}	E^{31}	
Differences		$0 \rightarrow 1$	$0 \rightarrow 1$	$0 \rightarrow 1$	$0 \rightarrow 1$	$0 \rightarrow 1$	None

Chabaud and Joux' Attack – The Attack (cont.)

The attack is initiated by a selection of a difference Δ, that is later used as the difference of the two colliding messages.

We call the 80-word vector $\mathcal{D}$ describing the rounds where disturbances are introduced due to that Δ a **disturbance vector** (each word is either 0 or 1).

Let $\mathrm{SR}^l(\mathcal{D})$ be the vector of 80 words received by prepending l zero words to the first $80-l$ words of $\mathcal{D}$ (i.e., a non-cyclic shift operation of the words).

Then, the corrections are made in bit 6 in the rounds which correspond to non-zero words in $\mathrm{SR}^1(\mathcal{D})$, in bit 1 in $\mathrm{SR}^2(\mathcal{D})$, and in bit 31 in $\mathrm{SR}^3(\mathcal{D})$ and $\mathrm{SR}^4(\mathcal{D})$ and $\mathrm{SR}^5(\mathcal{D})$. Thus, the expansion of Δ to 80 round can be written in the form

$$\begin{aligned}\exp(\Delta) \ = \ & \big(\big(\mathcal{D} \oplus \mathrm{SR}^2(\mathcal{D})\big) \lll 1\big) \oplus \\ & \big(\ \mathrm{SR}^1(\mathcal{D}) \lll 6\big) \oplus \\ & \big(\big(\mathrm{SR}^3(\mathcal{D}) \oplus \mathrm{SR}^4(\mathcal{D}) \oplus \mathrm{SR}^5(\mathcal{D})\big) \lll 31\big) \,,\end{aligned}$$

Note also that since each column is generated by the LFSR of the W's, so does $\exp(\Delta)$, and thus the disturbances and the corrections.

Chabaud and Joux' Attack – The Attack (cont.)

There are at most 2^{16} disturbance vectors (due to the 16 words of the message), of which

1. many have inappropriate forms (such as when $\exp(\Delta)$ starts with a correction in the first round)
2. many predict differences at the hash value (i.e., in the last rounds — these are near-collisions)
3. each disturbance vector has some probability for conformance of the pairs of messages, the higher probability the better for the attack
4. some have probability zero (for example, this probability is caused by consecutive disturbances in rounds 1–17)

Chabaud and Joux identified a disturbance vector that can be used to find collisions with complexity 2^{61}, and used it to find collisions of 35-round reduced SHA-0 (for 36–79 rounds it predicts only near-collisions).

Chabaud and Joux' Attack – The Attack (cont.)

We will say that a pair of messages **conforms to r rounds** if the differences of the registers during hashing are as predicted by the disturbance vectors in the first r rounds.

Part II: SHA-0

Our Observations — Observation 1

For each pair of messages there are many **neutral bits** — bits whose complementation in both messages does not change the differences of the registers in the following rounds.

In particular, there are neutral bits whose complementation does not affect the differences for many rounds.

Similarly, there are pairs (and larger sets) of bits whose simultaneous complementation does not affect the differences.

Take a pair of messages that conforms to 22 rounds. Complement a neutral bit in both messages. The new pair of messages conforms to 22 rounds as well.

Our Observations — Observation 1 (cont.)

Example: For the pair of messages given in the CRYPTO paper, the following bits are neutral — differences conform up to round 22:

Singles:	388	457	458	459	464	484	485	489
	490	491	494	495	496	499	501	506
	507							
Pairs:	301 264		461 424		493 456		497 460	
	500 463		502 428					
Triplets:	296 175 138		341 220 183		376 255 218		386 265 228	
	391 270 233		462 426 425		466 429 393		488 483 478	
	492 334 297							
Quadru-plets:	229 137 108 71		331 210 116 79		364 338 337 300			
	455 435 434 397		505 437 431 400					
Quintu-plets:	471 470 469 433 395				487 465 344 343 306			
	504 480 451 438 420							

Our Observations — Observation 1 (cont.)

Notice:

1. Some of the shown bits conform for further rounds (e.g., 25–26 rounds in case of single bits).
2. The quadruplet bits 229 137 108 71 are from rounds 2–7, i.e., they do not affect the differences for **15–20 rounds**.

Our Observations — Observation 2

A large fraction of the combinations of neutral bits is also neutral (about 1/8).

Moreover,

1. One composition of 10 bits from the 17 single neutral bits can leads to conformance of 49 rounds.
2. One composition of four bits and the above quadruplet leads to conformance of 54 rounds.
3. Thus, in less than 2^{18} complexity we can find pairs conforming to 54 rounds.
4. One composition of single neutral bits and pairs of bits leads to conformance of 58 rounds.

Our Analysis

The probability that a pairs conforms to 82 rounds, when it conforms to 22 is much higher than the probability that a random pair conform to 82 rounds.

Since we found an efficient method to generate pairs that conform to 22 rounds, we should only analyze the remaining rounds!

The probability that a pair that conforms to 22 rounds also conforms to 82 rounds is about 2^{-40}.

Thus, testing all the combinations of the 40 neutral bits shown should find a pairs that conforms to about 82 rounds.

Considering all the small factors, 82-round collisions should be found in about 2^{43} complexity.

Expected Complexities of Various Numbers of Rounds

The complexities for finding collisions are:

#Rounds	Complexity
50	– (*)
64	2^{29}
65	2^{29}
68	2^{43}
72	2^{50}
73	2^{50}
74	2^{50}

#Rounds	Complexity
75	2^{52}
76	– (*)
77	2^{66}
78	2^{56}
79	2^{56}
80	2^{56}
81	2^{43}

#Rounds	Complexity
82	2^{43}
83	2^{65}
84	2^{64}
85	2^{71}
86	2^{72}
87	– (*)
92	2^{74}

(*) There is no disturbance vector with non-zero probability that predicts collisions after 50, 76 or 87 rounds (and a few others).

Observation 3

The strength of reduced/extended SHA-0 is **not monotonous** with the number of rounds.

82-round SHA-0 is much less secure than 80-round SHA-0

The disturbance vector that predicts collisions for 82-round SHA-0 also

1. predicts collisions for 65-round SHA-0
2. predicts near-collisions for SHA-0 (80 rounds) with only about three differing bits.

Note: The best message pairs we found conform only to 77 rounds of the difference, thus we show pairs of messages with differences of about 18 bits

$\Longrightarrow$ Finding near-collisions is easier than finding collisions – the gained insight may be useful for finding collisions

Comparison of C&J Attack to Our Attack

	Chabaud and Joux		Our Results	
	Rounds	Complexity	Rounds	Complexity
Optimized for	80	2^{61}	82	2^{43}
Best collision found	35	2^{14}	65	2^{29} (*)
Conforming rounds found	≈ 56	–	77	2^{40} (**)
Near-collisions (18-bit diff)	–	–	80	2^{40} (**)

(*) About half an hour on a PC

(**) Our actual search took less than a day on a PC which is equivalent to a complexity of 2^{35}

Near-Collisions as a Tool for Finding Collisions

We mentioned that there is no disturbance vector for **50-round SHA-0**, as well as for some other reduced and extended numbers of rounds.

By the described technique, **no collision can be found**, but it is still possible to find near-collisions.

We now describe a **simple tool** that uses these near-collisions for finding collisions.

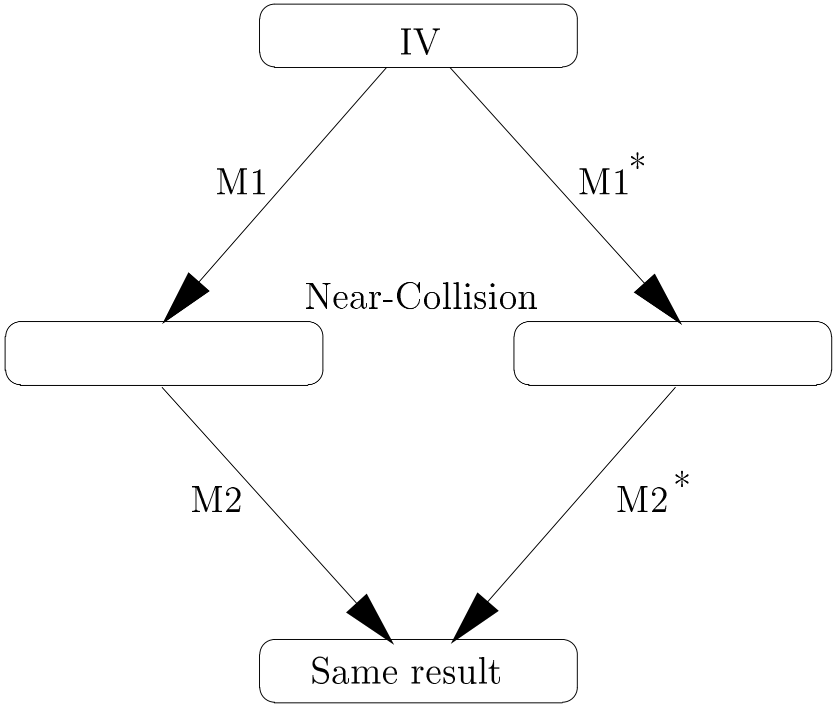
Near-Collisions as a Tool for Finding Collisions (cont.)

The following pair of blocks form a near-collision of 50-round SHA-0 with a difference of one bit:

B91F6175	66BB87FB	8946CC3E	41ED923C	508360CA	6179277A	67BD9099	7C895FD4
A7A145DE	91A17B62	A6FA374E	EB9E2734	B1C0F51E	4821B72B	CCA776A9	4D73A779
B91F6175	66BB87FB	8946CC3C	41ED927C	508360C8	E1792778	E7BD90D9	FC895FD6
27A145DE	11A17B62	26FA374C	EB9E2774	B1C0F51C	C821B72B	4CA776A9	CD73A779

Near-Collisions as a Tool for Finding Collisions (cont.)

With this near-collision at hand, it is possible to start with the resulting **two slightly different initial values** of a second block, and find collisions after the second block using another disturbance vector:



The two colliding messages are

$$M = M1\|M2, \qquad M^* = M1^*\|M2^*$$

Near-Collisions as a Tool for Finding Collisions (cont.)

The second block in our case is:

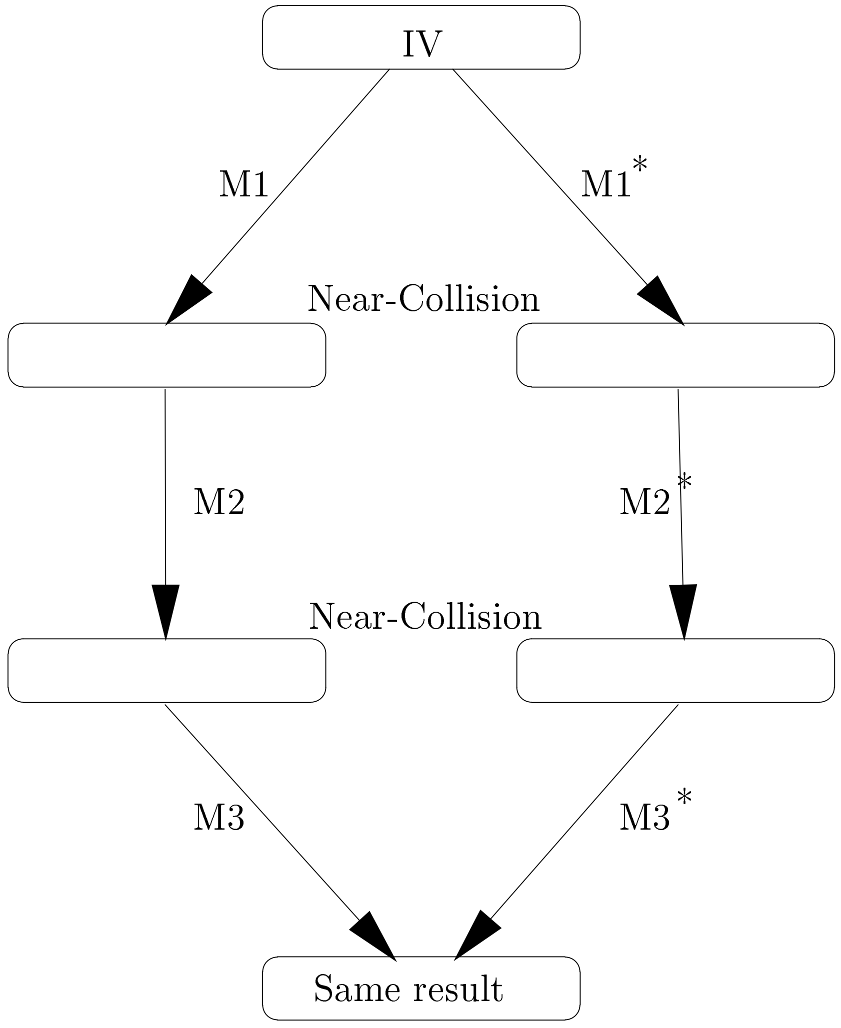
D284C629	8465BA04	2043ECDF	4AEBA300	DB2EA81A	F24E0E28	CA682AFD	6B9C328E
27A4B3C0	16F7D878	A7A004CB	46159612	992C22A5	D300E109	EB7F7D70	9FEC2DA7
D284C62B	0465BA04	A043ECDD	CAEBA340	DB2EA81A	724E0E68	4A682AFD	6B9C32CE
A7A4B3C2	16F7D878	27A004CB	C6159612	992C22A5	D300E109	EB7F7D70	9FEC2DA7

Thus, the following two two-block messages collide under 50-round SHA-0:

B91F6175	66BB87FB	8946CC3E	41ED923C	508360CA	6179277A	67BD9099	7C895FD4
A7A145DE	91A17B62	A6FA374E	EB9E2734	B1C0F51E	4821B72B	CCA776A9	4D73A779
D284C629	8465BA04	2043ECDF	4AEBA300	DB2EA81A	F24E0E28	CA682AFD	6B9C328E
27A4B3C0	16F7D878	A7A004CB	46159612	992C22A5	D300E109	EB7F7D70	9FEC2DA7
B91F6175	66BB87FB	8946CC3C	41ED927C	508360C8	E1792778	E7BD90D9	FC895FD6
27A145DE	11A17B62	26FA374C	EB9E2774	B1C0F51C	C821B72B	4CA776A9	CD73A779
D284C62B	0465BA04	A043ECDD	CAEBA340	DB2EA81A	724E0E68	4A682AFD	6B9C32CE
A7A4B3C2	16F7D878	27A004CB	C6159612	992C22A5	D300E109	EB7F7D70	9FEC2DA7

Near-Collisions as a Tool for Finding Collisions (cont.)

Note that in some cases 3-block, 4-block, or a longer collision may be necessary or may have the least complexity, where near-collisions occur after the first and second blocks (and further blocks, except for the last block).



Extensions of SHA-0

"Stronger" versions with more rounds are insecure:

As we mentioned already, 81-round and 82-round extensions of SHA-0 are weaker than (80-round) SHA-0.

Extensions of SHA-0 (cont.)

- Antoine Joux inserted our ideas to his program and found:
- **Collision of 81-round SHA-0**, which is also
- A near-collision of the comp. func. of SHA-0 with one differing bit!
- Collision of 128-bit truncated output of the comp. func. of SHA-0!
- A free-start collision for 82 rounds, where we add one round at the beginning and one at the end!
- A free-start collision for 80 rounds, where we remove one round at the beginning and add one at the end instead!

$M_1 =$

```
0EA10361 4D722EBD 9B61C37E 8ACA5B37 9D1B4004 BB489C0B D3387A74 7A39DAF8
5E533273 946EDE61 E45D47A8 FDF1D379 E70C5964 2051DBC8 68BA52A8 F4D3D1D1
```

$M_2 =$

```
0EA10361 4D722EBD 9B61C37C 8ACA5B77 9D1B4006 3B489C0B 53387A74 FA39DAFA
5E533233 946EDE61 645D47E8 7DF1D37B E70C5966 A051DB88 E8BA52AA 74D3D1D1
```

Part III: SHA-1

Application to SHA-1

In SHA-1, it is still possible to find disturbance vectors, but due to the additional rotate operation in the message schedule we have much less freedom in their selection:

1. It is not possible to enforce all the disturbances to appear in the same bit location in all rounds.
2. Necessarily at least 23 different bit locations are active during the 80 rounds.
3. The disturbances and corrections cannot be limited to bits 1, 6, 31.
4. $\Longrightarrow$ The disturbance vectors have considerably higher Hamming weights.
5. $\Longrightarrow$ The probabilities are considerably smaller.
6. $\Longrightarrow$ The technique is not applicable to the full SHA-1.

34-Round SHA-1

For attacking 34-round SHA-1 we use the following differences (HW= 2 from round 20 and on):

Rnd	D.Vec	D&C	Rnd	D.Vec	D&C	Rnd	D.Vec	D&C
-5	00000000		8	00000000	80000003	21	00000000	00000040
-4	00000000		9	00000002	40000002	22	00000002	00000000
-3	00000000		10	00000000	C0000040	23	00000000	80000040
-2	00000000		11	00000000	C0000002	24	00000000	80000002
-1	00000000		12	00000000	80000000	25	00000000	00000000
0	00000002	00000002	13	00000000	80000000	26	00000000	80000000
1	00000000	00000040	14	00000002	80000002	27	00000000	80000000
2	00000002	00000000	15	00000000	00000040	28	00000000	00000000
3	00000000	80000040	16	00000000	00000002	29	00000000	00000000
4	00000002	80000000	17	00000000	80000000	30	00000000	00000000
5	00000000	00000040	18	00000000	80000000	31	00000000	00000000
6	00000003	80000001	19	00000000	80000000	32	00000000	00000000
7	00000000	00000060	20	00000002	00000002	33	00000000	00000000

Collisions (after the round) are marked in **pink**. The message difference is marked in blue.

34-Round SHA-1 (cont.)

1. The following messages collide after 34 rounds of SHA-1:

F1641C2B	242BFDB5	EAE01E30	F4BBBA6F	18D45E8E	DE68AEBA	74EC8CF9	FC204957
45AAA8BF	1CD3AE7D	D845C2F3	AC737464	F25BEBBB	BE5FFF1D	2ADB2818	0B1D13FB
F1641C29	242BFDF5	EAE01E30	74BBBA2F	98D45E8E	DE68AEFA	F4EC8CF8	FC204937
C5AAA8BC	5CD3AE7F	1845C2B3	6C737466	725BEBBB	3E5FFF1D	AADB281A	0B1D13BB

   We found **millions** of such collisions

2. We also have **over 100** near-collisions with 40 conforming rounds, including three-bit difference after 40 rounds, 9 after 43, and 17 after 44

3. And even messages with **43 conforming rounds**

4. There is a lot of freedom for the attacker in this 34-round attack, and thus we can also find...

Collisions of 34-Round SHA-1 in ASCII

The following colliding pair of messages consists entirely of letters in ASCII:

Message 1: `IkGDqVMwISGGcBMpNHMYavPTsmUlykPTzokJOkwnrSgJSfDmlpeqsmDzWbAjmNxP`

Message 2: `IkgDqRMwISGGcFEpNHEYarPTsmMlymPTzoSJOkSnrWkJSfhmlpmqsmLzWbijmJxP`

In hex the messages are:

496B4744	71564D77	49534747	63424D70	4E484D59	61765054	736D556C	796B5054
7A6F6B4A	4F6B776E	7253674A	5366446D	6C706571	736D447A	5762416A	6D4E7850
496B6744	71524D77	49534747	63464570	4E484559	61725054	736D4D6C	796D5054
7A6F534A	4F6B536E	72576B4A	5366686D	6C706D71	736D4C7A	5762696A	6D4A7850

To get this effect we had to rotate the locations of the disturbances, which caused some increase in the complexity.

Collisions of 34-Round SHA-1 in ASCII (cont.)

The following collision contains a more meaningful text (case-sensitive):

Message 1: `I Am OilMANgujnPay916472136314$USAkNOWwTkjepMFXGlmfHNGcpodElGfvL`

Message 2: `I am KilMANgunfPay11607213.312$USASNOWSTknipMFtGlmnHNGkpodmlGbvL`

And also this pair:

Message 1: `OhG, not this mess,age notThat onenot U, oh noHRtBMTkKllLlIluvpB`

Message 2: `Ohg, jot this$eess$aga notLhar oneVot q, kd nodRtBETkKdlLlalurpB`

A 36-Round Collision

The following messages collide after 36 rounds of SHA-1 (two blocks each):

00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
9F29DE8D	BBD58270	1F11EB22	A6637C3E	7E6FB0C0	63E9BF5E	C4FF7010	073174B3
3133689A	579A753E	2D17124D	7D37E853	5B5BBB01	F0371FBB	025A725C	8FB9FE33
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
1F29DE8D	BBD58260	9F11EB23	86637C1E	5E6FB0C2	03E9BF3C	84FF7052	87317493
313368DB	579A7536	2D17120F	2D37E811	4B5BBB11	60371FFB	225A725C	AFB9FE73

The IV of SHA-1 does not allow to apply this attack. The first block is used to **replace** the IV by a good one.

Larger Numbers of Rounds

The following table summarizes Hamming weights (from round 20 and on) for the best differences we could find:

Rounds	HW	w/o IF	Rounds	HW	w/o IF	Rounds	HW	w/o IF
28–34	**2** found	2	42	17	13	50	35	**14**
35	7	4 found	43	17	17	51	42 (32)	15
36	**7** found	5	44	19	15	52	42	16
37	11	5 found	45	25	15	53	42	**16**
38	12	8	46	25	23	54	42	36
39	12	11	47	27	25	55	63	42
40	19 (15)	11 (10)	48	31 (25)	14	56	41	41
41	17	12	49	34	14	57		42

The probabilities are expected to be on the order of $2^{-3\mathrm{HW}}$ (depending on various details).

If SHA-1 did not have the IF rounds at the beginning, the HW would decrease to the "w/o IF" figures, and attacks could be found for up to about **53 rounds** (e.g., the last 53 rounds).

Note that in most of these cases a two-block method using a near-collision at the first block may have (slightly) lower complexities. See examples of 40, 48, 51 rounds.

Summary

1. Observed the existence of neutral bits
2. Found collisions of 65-round SHA-0
3. Found near-collisions of 80-round SHA-0 (difference of 18 bits)
4. Shown that the security of SHA-0 is not monotonous with the number of rounds
5. Simple changes to SHA-0, such as increasing the number of rounds to 82, or changing the order of the f_t functions, would make it weaker
6. Found many collisions of 34-round SHA-1, and a collision of 36-round SHA-1
7. Collisions of more rounds may be found faster than with the birthday attack

Summary (cont.)

8. **SHA-0 should not be used any more**

9. At this time, we do not expect to find collisions of the full SHA-1

The End