Differential Path for SHA-1 with complexity $O(2^{52})$

Cameron McDonald[1,2], Philip Hawkes[2], and Josef Pieprzyk[1]

[1] Centre for Advanced Computing, Algorithms and Cryptography
Department of Computing, Macquarie University
{cmcdonal,josef}@ics.mq.edu.au
[2] Qualcomm Australia, Level 3, 77 King St,
Sydney 2000, Australia
phawkes@qualcomm.com

Abstract. Although SHA-1 has been theoretically broken for some time now, the task of finding a practical collision is yet to be completed. Using some new approaches to differential analysis, we were able to find a new differential path which can be used in a collision attack with complexity of $O(2^{52})$. This is currently the lowest complexity attack on SHA-1.

Keywords: Hash Functions, Differential Path, Boomerang Attack, SHA-1.

1 Introduction

The MD-SHA family of dedicated hash functions are the most well known hash functions to date. First introduced by R. Rivest in 1990 was MD4 [13], an improved version MD5 [14] soon followed in 1991 and was adopted as internet standard RFC-1321. In 1993, the NSA used similar principles to MD5 and designed SHA-0, which was published by NIST as a US standard [11]. It was withdrawn soon after publication and replaced by SHA-1 [12] in 1995. SHA-1 is identical to SHA-0 except for a 1-bit rotation in the message expansion. The MD-SHA family uses the iterative Merkle-Damgård structure with a compression function based on a block cipher in Davies-Meyer mode.

Chabaud and Joux [4] observed that the compression function of SHA-0/1 has a 6-step local collision. By interleaving multiple local collisions, they were able to build a probabilistic linear differential path. This provided the first theoretical collision attack on SHA-0, having complexity 2^{61}. Biham and Chen [1] introduced the notion of neutral bits and found near collisions in SHA-0, this was later extended to full collisions [2] at a complexity of 2^{51}. Wang et al. introduced modular addition differentials and message modification techniques and provided collisions for SHA-0 in 2^{39} [16] and the first theoretical collision attack on SHA-1 [17], having a complexity of 2^{69} message modifications.

Strategies similar to the neutral bit technique evolved to Klima's tunnels in MD5 [8] and Joux and Peyrin's Boomerang Attack [7]. These techniques take advantage of message bits that are "independent" with respect to the differential path and can be modified to produce new message pairs without effecting the conditions required to conform to the main differential path.

Manuel [9] classified disturbance vectors for SHA-1 and published a new low weight disturbance vector which could lead to a collision attack with complexity 2^{57} and suggested a complexity as low as 2^{51} might be possible when using techniques such as the boomerang attack.

Our goal was to find a non-linear differential path for Manuel's disturbance vector with lowest complexity. Our approach was a combination of automated path searching, analysis by hand and new SAT techniques. We found many new differential paths which could be used in a boomerang attack. The path with lowest complexity uses 5 auxiliary paths and has total complexity 2^{52}. This attack is much lower than the previous best attack of 2^{63}.

The paper outline is as follows. In §2, we introduce the notation used. A brief description of SHA-1 is given in §3. We describe collision attacks and previous work in §4. §5 briefly introduces the techniques used and discusses new results. Details on the techniques will be published at a conference in the near future.

2 Notation

SHA-1 was designed using 32-bit words, each variable will represent one 32-bit word unless otherwise stated. Let $\oplus$ denote bit-wise XOR and $+$ denote addition modulo 2^{32}. The symbol $\lll$ denotes rotation to the left. Let a_i^j represent the j-th least significant bit of variable a at step i. The bitwise complement of a is $\bar{a}$. Let (a, a') represent a pair of variables with XOR-difference $\Delta_{\oplus} a = a \oplus a'$ and modular add-difference $\Delta_{+} a = (a' - a) \bmod 2^{32}$. We adopt the following notation from [6] which describes in detail a difference ∇a, where $\nabla a^j \in \{., 0, 1, +, -, v, x\}$ such that:

$$\nabla a_i^j = \begin{cases} ., & \text{if } a_i^j = a_i'^j; \\ \mathtt{0}, & \text{if } a_i^j = a_i'^j = \mathtt{0}; \\ \mathtt{1}, & \text{if } a_i^j = a_i'^j = \mathtt{1}; \\ *, & \text{if } a_i^j \neq a_i'^j; \\ +, & \text{if } a_i^j = \mathtt{0} \text{ and } a_i'^j = \mathtt{1}; \\ -, & \text{if } a_i^j = \mathtt{1} \text{ and } a_i'^j = \mathtt{0}; \\ \mathtt{v}, & \text{if } a_i^j = a_{i+1}^j \text{ and } a_i'^j = a_{i+1}'^j; \\ \mathtt{x}, & \text{if } a_i^j \neq a_{i+1}^j \text{ and } a_i'^j \neq a_{i+1}'^j. \end{cases}$$

3 Description of SHA-1

SHA-1 is a dedicated hash function that takes a message less than 2^{64} bits in length and computes a 160-bit digest. The input message is padded and divided into 512-bit blocks. Each iteration takes a chaining variable and a new message block, employs a compression function and produces the next chaining variable. The initial chaining value is a specified constant and the final chaining value is used as the output.

The compression function processes one 512-bit message block per iteration. The 512-bit message is parsed into sixteen 32-bit words $(M_0, \ldots, M_{15})$, which are then expanded to 80 words using the following:

$$W_i = \begin{cases} M_i & \text{if } 0 \leq i \leq 15, \\ (W_{i-3} \oplus W_{i-8} \oplus W_{i-14} \oplus W_{i-16}) \lll 1 & \text{if } 16 \leq i \leq 79, \end{cases}$$

The compression function operates on a register of five 32-bit words $(A_i, B_i, C_i, D_i, E_i)$, initially loaded with the previous chaining value. The state is modified over 4 rounds, each consisting of 20 steps of the following process:

$$\begin{aligned} A_{i+1} &= (A_i \lll 5) + f_i(B_i, C_i, D_i) + E_i + W_i + k_i \\ B_{i+1} &= A_i \\ C_{i+1} &= B_i \lll 30 \\ D_{i+1} &= C_i \\ E_{i+1} &= D_i. \end{aligned}$$

The boolean functions f_i and the constants k_i are specified in Table 1. Note that the updated registers B, C, D and E are rotated copies of previous A registers, hence we can describe the update function by the following recurrence formula in terms of A registers only:

$$A_{i+1} = (A_i \lll 5) + f_i(A_{i-1}, (A_{i-2} \lll 30), (A_{i-3} \lll 30)) + (A_{i-4} \lll 30) + W_i + k_i.$$

Round	Step i	$f_i(B, C, D)$	k_i
1	$0 \leq i \leq 19$	$(B \wedge C) \vee (\bar{B} \wedge D)$	`0x5A827999`
2	$20 \leq i \leq 39$	$B \oplus C \oplus D$	`0x6ED6EBA1`
3	$40 \leq i \leq 59$	$(B \wedge C) \vee (B \wedge D) \vee (C \wedge D)$	`0x8FABBCDC`
4	$60 \leq i \leq 79$	$B \oplus C \oplus D$	`0xCA62C1D6`

Table 1. Boolean functions and constants in SHA-1

For a complete reference, see [12].

4 Collision Attacks on SHA-1

A large amount of research has contributed to the development and improvment of collision attacks on SHA-1 over the last 10 years. We aim to cover some of the important steps and knowledge required to explain the current situation and our contribution.

Chabaud and Joux [4] observed that SHA-0 (and consequently SHA-1) has a 6-step local collision for any step i and any starting bit position j. The local collision introduces a single bit difference in W_i^j followed by several more conditions to eventually remove all effects in 6 steps. Figure 1 traces the local collision as it passes through each of the 6 steps, showing the consequential effects and the adjustments necessary to remove them.

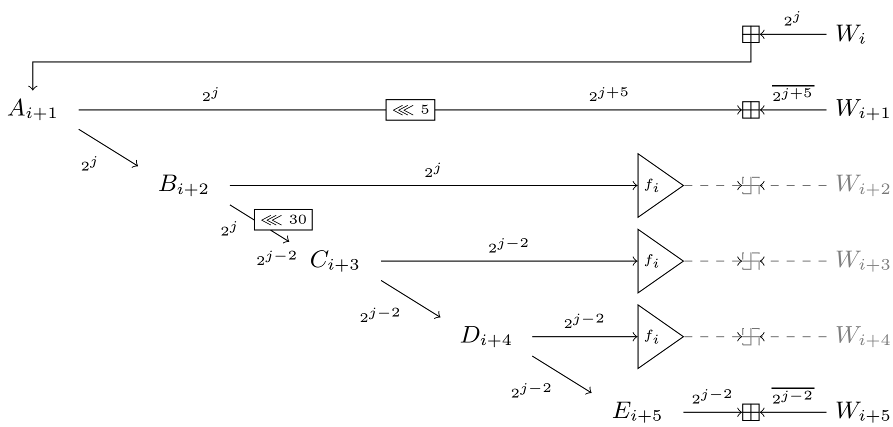

Fig. 1. Propagation of a single bit difference over 6 steps of the compression function. Only state variables that are affected by the difference are shown. The greyed section indicates where different variations of the local collision can occur due to the behaviour of the boolean function f_i.

The round dependent boolean function requires different conditions for the local collision to uphold, leading to slight variations for each round. For example, an input difference to the *IF* function in the first round can result in either the difference being *absorbed* (no difference in the output) or *preserved* (a difference in the output).

The message differences introduced in the local collision are reintroduced at later steps due to the message expansion. By interleaving a sequence of local collisions, Chabaud and Joux built a linear differential path for the full SHA-0 compression function. The indices of the initial difference (perturbation) of each local collision is governed by a *disturbance vector*, found by analysis of the message expansion. The probability that the differential path holds is closely related to the number of disturbances made (the Hamming weight of the disturbance vector).

A *message pattern* describes the initial differences from the disturbance vector along with the consequent message differences required for each local collision. The differential path consists of a sequence of differences in the message (message pattern) along with the corresponding differences through the state variables, specified by each local collision. The differential path for SHA-0 could not be directly transfered to SHA-1 due to the rotation in the message expansion.

In 2005, Wang et al. published differential results on several hash functions, including SHA-0 [16] and SHA-1 [17]. This influential analysis contributed three significant findings: non-linear differentials, two-block collisions from near-collisions, and message modification. The non-linear differential models carry propagation from the modular addition as well as difference interaction in the boolean function f_i. This analysis provides differentials in the first round that hold with probability 1. The non-linear differential needs to converge to the probabilistic linear model at the end of round 1 which it follows for the remaining 3 rounds. The flexibility provided by the non-linear analysis allowed the restriction of no difference in the IV to be removed from the disturbance vector requirements.

Wang also argued that near collisions could be used to find two block collisions, this removed the requirement that the disturbance vector should have no difference in the output. These observations made available a whole new class of disturbance vectors to analyse, leading to an improved differential for SHA-0 [16] as well the first differential path for SHA-1 [17].

Along with this non-linear differential analysis, Wang provided a tool described as *message modification*, a method of manipulating message pairs so they conform to the differential path.

The current collision attacks on SHA-1 utilise results from 3 key areas of analysis. They include:

1. Finding a low cost disturbance pattern.
2. Building a non-linear differential in the first round.
3. Employing fast message generation techniques.

Generally, analysis on each area is conducted independently of the others. This is the case since it is difficult to specify the combined requirements that lead to the lowest complexity attack. There are currently several message generation techniques, each offering its own advantage, and there is not obvious benefit in choosing one method over another. It seems that best results are gained from a combination of all methods. The effectiveness of some message generation techniques cannot be established until a differential path is known and it is difficult to specify the requirements prior to path construction.

4.1 Message Generation

Once a differential path is known, one can start searching for a practical collision. This process involves generating pairs of messages that conform to the differential path in the hope of finding a pair conformant to step 80. The analyst has direct control over the first 16 message words which may be manipulated to obtain the conditions required for the differential to hold. Message modification, introduced in [17], is a technique for further message manipulation so the message pair is conformant to a few more steps, until approximately step 20. Advanced message modification builds on the same techniques to generate conformant message pairs to a few more steps once again, up to step 24 in practice. The details for applying these techniques to the differential published in [17] is given here [5]. In [3] and [10] the authors provide observations and implementation considerations when applying a practical collision attack.

Biham and Chen [1] introduced the notion of neutral bits. Informally, a neutral bit is message bit that is independent of the message bits constrained to the differential path. This means a neutral bit can be flipped in both messages and the new message pair will still conform to the differential path. Ideas similar to this led to Klima's tunnels in MD5 [8] and Joux and Peyrin's Boomerang attack [7].

4.2 Boomerang Attack

The Boomerang Attack was first used as a tool in the cryptanalysis of block ciphers [15]. It involves merging two independent partial differential characteristics to aid in an attack on the full cipher. Joux and Peyrin [7] adapt the amplified boomerang attack (chosen plaintext variant) to the hash function setting.

For the attack to be successful, a global main differential path must be found where t independent auxiliary paths can be placed such that the dynamic conditions (bits that are modified during the attack) are independent of the conditions on the main differential. One could say that the auxiliary path is orthogonal to the main differential. An m-step auxiliary path is basically a local collision that holds (collides) up to step m.

Table 2 illustrates the 24-step auxiliary published by Joux and Peyrin [7] generated by 3 perturbation points.

During the attack stage a message pair conformant to step m is found. Applying the conditions from the auxiliary path produces another message pair that is also conformant to the main differential until at least step m. Since the auxiliary paths are independent, there are 2^t combinations in which the auxiliary paths can be applied. This provides 2^t new message pairs, all conformant to at least step m of the main differential. This amplification of messages is achieved at no extra work effort. The disadvantage of this method is that extra conditions are placed on the main differential, lowering the number of free bits available in message modification. The conditions required for the Boomerang Attack to succeed can be specified prior to the differential path construction.

The task of placing a maximum number of auxiliaries in a main differential is difficult due to the overlap of constraints. In [7], the authors force as much space between the auxiliary constraints. We have found in practice that placing auxiliary paths adjacent to each other causes no problem in either building a path or generating messages and applying the boomerang attack.

Our research is currently focussed on the generation of non-linear differential paths with the option of placing maximum auxiliary paths in a boomerang attack.

i	∇A_i	∇W_i
-1	`...........................v....`	
00	`................................`	`.............................a..`
01	`...........................v.a..`	`........................ā.......`
02	`...............................1`	`.............................b..`
03	`.............................b.0`	`........................b̄......ā`
04	`...............................0`	`...............................ā`
05	`...............................0`	`...............................ā`
06	`................................`	`...............................b̄`
07	`................................`	`...............................b̄`
08	`................................`	`................................`
09	`...........................v....`	`................................`
10	`................................`	`.............................c..`
11	`.............................c..`	`........................c̄.......`
12	`...............................0`	`................................`
13	`...............................0`	`................................`
14	`................................`	`...............................c̄`
15	`................................`	`...............................c̄`

Table 2. 24-step Auxiliary Path from [7]

5 New Differential Paths

In [9], Manuel classifies the current known disturbance vectors for SHA-1, and introduces a new disturbance vector with a complexity evaluation of 2^{57}, the lowest complexity published to date. Manuel did not provide a non-linear differential path corresponding to the new disturbance vector.

Our aim was to find a new differential path for Manuel's disturbance vector that contained the maximum number of auxiliary differentials for a boomerang attack.

We have used a variety of different methods and tools in approaching the problem of building non-linear differentials. We implemented a path searching tool, similar to the one described in [6] and [18]. We identified specific areas where the computer search was failing and built partial paths by hand to avoid the problems encountered. Areas where lots of difference interaction occurred were described by a corresponding SAT instance. We then employed a SAT solver to find solutions to this instance. The details of these methods will be published at a conference.

Using a combination of these methods, we were able to find a differential path with 5 auxiliary paths. The total complexity of a collision attack using this path is 2^{52} and is illustrated in Table 3. As mentioned in [7], the auxiliary paths place constraints on the IV. The following prepended message provides a chaining value with the required constraints:

W_0	0x1BD65216	W_1	0x46A40496	W_2	0x4E1942BD	W_3	0x4F37CC60
W_4	0x417801CE	W_5	0x0207CCC2	W_6	0x3ADB2203	W_7	0x65739B62
W_8	0x2A6F80CE	W_9	0x608B4726	W_{10}	0x35DAB724	W_{11}	0x19DF3347
W_{12}	0x57E6162C	W_{13}	0x08C6DF1C	W_{14}	0x0481F258	W_{15}	0x41EA26CF

We have found several similar paths to the one listed in Table 3, each varying slightly in the top 3 bits or bottom 2 bits. We are currently determining if any of the variations provide an advantage over the others during the message generation stage of the attack.

In the appendix, we give an example of how the boomerang attack is applied. Table 4 lists a message pair and corresponding state values that conform for 40 steps of the differential path illustrated in 3. Applying all 5 auxiliary paths at once to the first message pair is shown in Table 5. Finally, Table 6 lists the new amplified message pair, showing it also conforms to the differential path for at least the first 24 steps.

6 Conclusion

Using the techniques described we successfully found new non-linear differential paths for SHA-1. The path which yields the best attack has complexity of $O(2^{52})$ when used in a boomerang attack. This is a significant reduction to the previous best result of 2^{63}. We believe that practical collisions are now within reach of a dedicated system. We are continuing our search for more differential paths with a maximum number of auxiliary paths.

References

1. Eli Biham and Rafi Chen. Near-collisions of sha-0. In *CRYPTO*, volume 3152 of *LNCS*, pages 290–305. Springer, 2004.
2. Eli Biham, Rafi Chen, Antoine Joux, Patrick Carribault, Christophe Lemuet, and William Jalby. Collisions of sha-0 and reduced sha-1. In *EUROCRYPT*, volume 3494 of *LNCS*, pages 36–57. Springer, 2005.
3. Christophe De Canneére, Florian Mendel, and Christian Rechberger. Collisions for 70-step sha-1: On the full cost of collision search. In *Selected Areas in Cryptography*, volume 4876 of *LNCS*, pages 56–73. Springer, 2007.
4. Florent Chabaud and Antoine Joux. Differential collisions in sha-0. In *CRYPTO*, volume 1462 of *LNCS*, pages 253–261. Springer, 1998.
5. Martin Cochran. Notes on the wang et al. 2^{63} sha-1 differential path. Cryptology ePrint Archive, Report 2007/474, 2007. http://eprint.iacr.org/.
6. Philip Hawkes, Michael Paddon, and Gregory G. Rose. Musings on the wang et al. md5 collision. Cryptology ePrint Archive, Report 2004/264, 2004. http://eprint.iacr.org/.
7. Antoine Joux and Thomas Peyrin. Hash functions and the (amplified) boomerang attack. In *CRYPTO*, volume 4622 of *LNCS*, pages 244–263. Springer, 2007.
8. Vlastimil Klima. Tunnels in hash functions: Md5 collisions within a minute. Cryptology ePrint Archive, Report 2006/105, 2006. http://eprint.iacr.org/.

9. Stephane Manuel. Classification and generation of disturbance vectors for collision attacks against sha-1. Cryptology ePrint Archive, Report 2008/469, 2008. http://eprint.iacr.org/.
10. Florian Mendel, Norbet Pramstaller, Christian Rechberger, and Vincent Rijmen. The impact of carries on the complexity of collision attacks on sha-1. In *Fast Software Encryption*, volume 4047 of *LNCS*, pages 278–292. Springer, 2006.
11. National Institute of Standards and Technology. Fips 180: Secure hash standard, 1993. http://csrc.nist.gov.
12. National Institute of Standards and Technology. Fips 180-1: Secure hash standard, 1995. http://csrc.nist.gov.
13. R. Rivest. Rfc 1320: The md4 message-digest algorithm, 1992. http://www.ietf.org/rfc/rfc1320.txt.
14. R. Rivest. Rfc 1321: The md5 message-digest algorithm, 1992. http://www.ietf.org/rfc/rfc1321.txt.
15. David Wagner. The boomerang attack. In *FSE*, volume 1636 of *LNCS*, pages 156–170. Springer, 1999.
16. Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Efficient collision search attacks on sha-0. In *CRYPTO*, volume 3621 of *LNCS*, pages 1–16. Springer, 2005.
17. Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding collisions in the full sha-1. In *CRYPTO*, volume 3621 of *LNCS*, pages 17–36. Springer, 2005.
18. Jun Yajima, Yu Sasaki, Yusuke Naito, Terutoshi Iwasaki, Takeshi Shimoyama, Noboru Kunihiro, and Kazuo Ohta. A new strategy for finding a differential path of sha-1. In *Information Security and Privacy*, volume 4586 of *LNCS*, pages 45–58. Springer, 2007.

i	∇A_i	∇W_i
-4	....	
-3	....	
-2	....	
-1	1v.1v....v..vv....v........v..xv	
00	1..0.................10.........	..+++-a....d..gj....m........+..
01	---+0+a..v.dvvgj11v.m01...vv.+..	-ā-+--d̄..ḡj̄....m̄...........++.+.
02	0.-00+.01...11..11....1+-...10x1	..+...b....e..hk....n......+....
03	1v+1.-b00..e00hk00+-n.0.101.++.1	.b̄+..-ē.āh̄k̄..d̄.n̄ḡj̄....m̄....-++..
04	-.+.00v101vvv0+.00..1100110.00+-	+--.--..ā....d̄..ḡj̄....m̄....-.-..
05	1-++01.+0+...0..00..00010.-.0000	....--..ā....d̄..ḡj̄....m̄.......-.
06	+110.1-++++++++.1.......1-+111--	+-......b̄....ē..h̄k̄....n̄....+....
07	++-010..1.1.11111......0v1-100++	-.++.-..b̄....ē..h̄k̄....n̄....-+-..
08	0-.01...010011111..0...1...+--.-	..+.+-.......................-..
09	0++11....v..vv....v1v0vvv+-.001-	+.+--+.....................++...
10	0.+01..............1.+...00010--	+.-+..c....f..il....p......+....
11	--.1..c....f..il....p-+++++101+-	.c̄....f̄..īl̄....p̄............-+..
12	+.+01...0....0..00....01111-+010	+.---......................-....
13	++010...0....0..00....00110111-+	....+......................++...
14	-+-10.......................0111	.+-++...c̄....f̄..īl̄....p̄....-....
15	++-.1.........................-+	....-...c̄....f̄..īl̄....p̄....+.+..
16	+...............................	.***.......................*....
17	***.............................	*.***......................***..
18	....	***.*...........................
19	..*.............................	*.**.........................*..
20	*...............................	.*.**......................*....
21	.**.............................	*.**........................**..
22	....	.*..*...........................
23	....	*.**............................
24	*...............................	*.***......................*....
25	*...............................	*..**......................*....
26	....	*.*.............................
27	....	....
28	....	....
29	....	..*.............................
30	....	*...............................
31	*...............................	...........................*....
32	....	....
33	*...............................	..*........................*....
34	....	..*.............................
35	*...............................	...........................*....
36	....	.**.............................
37	**..............................	...........................**...
38	....	***.............................
39	....	*..*............................
40	*...............................	..**.......................*....

i	∇A_i	∇W_i
41	....	*.**............................
42	....	..*.............................
43	....	..*.............................
44	....	*.*.............................
45	*...............................	...........................*....
46	....	*...............................
47	....	..*.............................
48	....	..*.............................
49	....	..*.............................
50	....	*...............................
51	*...............................	...........................*....
52	....	....
53	*...............................	..*........................*....
54	....	*.*.............................
55	....	....
56	....	..*.............................
57	....	..*.............................
58	....	....
59	....	....
60	....	....
61	....	....
62	....	....
63	....	....
64	....	...............................*
65	...............................*	..........................*.....
66	....	...............................*
67	....	.*............................*.
68	..............................*.	.*.......................*.....*
69	...............................*	.*........................*...*.
70	....	*............................*.*
71	.............................*..	**......................*.....*.
72	..............................*.	**.......................*...**.
73	..............................*.	.*.......................*..*.**
74	............................*...	*......................*.....***
75	.............................*..	........................*...*..*
76	....	...........................*.*..
77	...........................*..*.	*.....................*..*..*.**
78	............................*...	.......................*...**.**
79	............................*.*.	*......................*.**.**.*
80	..........................*.....	

Table 3. 80-steps of a new differential path with 5 auxiliary paths

i	A_i	A'_i	∇A_i	W_i	W'_i	∇W_i
-4	0xADACF0F7	0xADACF0F7	....			
-3	0x0710F3A4	0x0710F3A4	....			
-2	0xBD0A728F	0xBD0A728F	....			
-1	0xBF2CDD4E	0xBF2CDD4E	....			
00	0xAEAEC5A0	0xAEAEC5A0	....	0x457E50F8	0x797E50FC	..+++-.......................+..
01	0xE09EE24B	0x149EE24F	---+.+.......................+..	0xEDA308C1	0x51A308DB	-.-+--.....................++.+.
02	0x229CF689	0x069CF709	..-..+.................+-.......	0x034BAA0F	0x234BAA1F	..+........................+....
03	0x9E431CB3	0xBA432CBF	..+..-............+-........++..	0x87804392	0xA380438E	..+..-.....................-++..
04	0x83593CD1	0x235B3CD2	-.+...........+...............+-	0x6C9B7F5D	0x809B7F49	+--.--.....................-.-..
05	0xC6190130	0xB7590110	.-++...+.+................-.....	0x8C806A7A	0x80806A78	....--........................-.
06	0x6E00C7DF	0xEDFEC7BC	+.....-++++++++..........-+...--	0x671C100C	0xA71C101C	+-.........................+....
07	0x2BEF8A70	0xCBEF8A53	++-.......................-...++	0x8E7511D7	0x3A7511CB	-.++.-.....................-+-..
08	0x6F4FE90D	0x2F4FE910	.-.........................+--.-	0x5767A73D	0x7B67A739	..+.+-.......................-..
09	0x183DD233	0x783DD252	.++......................+-....-	0x1BFFA0C5	0xA7FFA0DD	+.+--+.....................++...
10	0x4A8E920B	0x6A8E9608	..+..................+........--	0x2FD70C8D	0x9FD70C9D	+.-+.......................+....
11	0xD27FEC15	0x127FEBF6	--...................-+++++...+-	0x0196CEBB	0x0196CEB7	............................-+..
12	0x0A7225F2	0xAA7225EA	+.+........................-+...	0x3CBC5639	0x84BC5629	+.---......................-....
13	0x177B3CDE	0xD77B3CDD	++............................-+	0xD6FF95E0	0xDEFF95F8	....+......................++...
14	0xB78CC0C7	0x578CC0C7	-+-.............................	0xA72301B0	0xDF2301A0	.+-++......................-....
15	0x287E041E	0xC87E041D	++-...........................-+	0xBF703C02	0xB7703C16	....-......................+.+..
16	0x23EFFDA4	0xA3EFFDA4	+...............................	0x8F5B9055	0xFF5B9045	.+++.......................-....
17	0x985D131F	0x785D131F	-++.............................	0xADFFD44D	0x15FFD451	-.-+-......................+--..
18	0x68130225	0x68130225	....	0xFEEFCBBB	0x16EFCBBB	---.-...........................
19	0x54114D3B	0x74114D3B	..+.............................	0x0B9AEE0D	0xBB9AEE09	+.++.........................-..
20	0xD3628D64	0x53628D64	-...............................	0x3589DA4B	0x6D89DA5B	.+.-+......................+....
21	0x01B43B64	0x61B43B64	.++.............................	0x55CA4BEC	0xE5CA4BE0	+.+-........................--..
22	0x3DA50777	0x3DA50777	....	0x3984B119	0x7184B119	.+..-...........................
23	0x976D179D	0x976D179D	....	0x3EE6AEB6	0x8EE6AEB6	+.--............................
24	0x79788C4E	0xF9788C4E	+...............................	0x4442E013	0xFC42E003	+.+++......................-....
25	0x6F705175	0xEF705175	+...............................	0x1C241655	0x84241645	+..--......................-....
26	0x4D3FC7F0	0x4D3FC7F0	....	0xA6C47F73	0x06C47F73	-.-.............................
27	0x21F5E27B	0x21F5E27B	....	0x3162AA8B	0x3162AA8B	....
28	0xCD921C35	0xCD921C35	....	0x6465372F	0x6465372F	....
29	0x8D46D476	0x8D46D476	....	0x35033AFB	0x15033AFB	..-.............................
30	0xBF345F21	0xBF345F21	....	0x413D14EE	0xC13D14EE	+...............................
31	0xE05202A7	0x605202A7	-...............................	0x9018E3AC	0x9018E3BC	...........................+....
32	0x40B20A18	0x40B20A18	....	0x01EB020C	0x01EB020C	....
33	0x27395834	0xA7395834	+...............................	0xF6F871F7	0xD6F871E7	..-........................-....
34	0xC7BAB6B2	0xC7BAB6B2	....	0xFB751A5F	0xDB751A5F	..-.............................
35	0xA0744E3B	0x20744E3B	-...............................	0xDDB21ACC	0xDDB21ADC	...........................+....
36	0x3182B0C3	0x3182B0C3	....	0x3D205B15	0x5D205B15	.+-.............................
37	0x04D1973C	0xC4D1973C	++..............................	0x4AB58BFD	0x4AB58BE5	...........................--...
38	0xC601C40C	0xC601C40C	....	0xC292BE51	0x2292BE51	--+.............................
39	0xC44001B8	0xC44001B8	....	0x1FF400B5	0x8FF400B5	+..-............................
40	0x0A4044EF	0x8A4044EF	+...............................	0x53B02D23	0x63B02D33	..+-.......................+....

Table 4. Message pair that conforms to step 40 of the differential

i	A_i	A_i^*	∇A_i	W_i	W_i^*	∇W_i
-4	0xADACF0F7	0xADACF0F7	....			
-3	0x0710F3A4	0x0710F3A4	....			
-2	0xBD0A728F	0xBD0A728F	....			
-1	0xBF2CDD4E	0xBF2CDD4E	....			
00	0xAEAEC5A0	0xAEAEC5A0	....	0x457E50F8	0x476D58F8	......+....-..-+....+...........
01	0xE09EE24B	0xE28DEA4B	......+....-..-+....+...........	0xEDA308C1	0xAFC208C1	.-....+..+-....-................
02	0x229CF689	0x229CF689	....	0x034BAA0F	0x0158A20F	......-....+..--....-...........
03	0x9E431CB3	0x9C5014B3	......-....+..--....-...........	0x87804392	0xC5658192	.+....-.-++..+.++-....-.........
04	0x83593CD1	0x83593CD1	....	0x6C9B7F5D	0x6C1FBD5D	........-....+..+-....-.........
05	0xC6190130	0xC6190130	....	0x8C806A7A	0x8C04A87A	........-....+..+-....-.........
06	0x6E00C7DF	0x6E00C7DF	....	0x671C100C	0x6798D20C	........+....-..++....+.........
07	0x2BEF8A70	0x2BEF8A70	....	0x8E7511D7	0x8EF1D3D7	........+....-..++....+.........
08	0x6F4FE90D	0x6F4FE90D	....	0x5767A73D	0x5767A73D	....
09	0x183DD233	0x183DD233	....	0x1BFFA0C5	0x1BFFA0C5	....
10	0x4A8E920B	0x4A8E920B	....	0x2FD70C8D	0x2DC4048D	......-....-..--....-...........
11	0xD27FEC15	0xD06CE415	......-....-..--....-...........	0x0196CEBB	0x43F7CEBB	.+....+..++....+................
12	0x0A7225F2	0x0A7225F2	....	0x3CBC5639	0x3CBC5639	....
13	0x177B3CDE	0x177B3CDE	....	0xD6FF95E0	0xD6FF95E0	....
14	0xB78CC0C7	0xB78CC0C7	....	0xA72301B0	0xA7A7C3B0	........+....+..++....+.........
15	0x287E041E	0x287E041E	....	0xBF703C02	0xBFF4FE02	........+....+..++....+.........
16	0x23EFFDA4	0x23EFFDA4	....	0x8F5B9055	0x8F5B9055	....
17	0x985D131F	0x985D131F	....	0xADFFD44D	0xADFFD44D	....
18	0x68130225	0x68130225	....	0xFEEFCBBB	0xFEEFCBBB	....
19	0x54114D3B	0x54114D3B	....	0x0B9AEE0D	0x0B9AEE0D	....
20	0xD3628D64	0xD3628D64	....	0x3589DA4B	0x3589DA4B	....
21	0x01B43B64	0x01B43B64	....	0x55CA4BEC	0x55CA4BEC	....
22	0x3DA50777	0x3DA50777	....	0x3984B119	0x3984B119	....
23	0x976D179D	0x976D179D	....	0x3EE6AEB6	0x3EE6AEB6	....
24	0x79788C4E	0x79788C4E	....	0x4442E013	0x4064F013	.....-....+..+-....+............

Table 5. Application of Auxiliary paths to first message

i	A_i^*	$A_i'^*$	∇A_i^*	W_i^*	$W_i'^*$	∇W_i^*
-4	0xADACF0F7	0xADACF0F7	....			
-3	0x0710F3A4	0x0710F3A4	....			
-2	0xBD0A728F	0xBD0A728F	....			
-1	0xBF2CDD4E	0xBF2CDD4E	....			
00	0xAEAEC5A0	0xAEAEC5A0	....	0x476D58F8	0x7B6D58FC	..+++-.......................+..
01	0xE28DEA4B	0x168DEA4F	---+.+.......................+..	0xAFC208C1	0x13C208DB	-.-+--.....................++.+.
02	0x229CF689	0x069CF709	..-..+.................+-.......	0x0158A20F	0x2158A21F	..+........................+....
03	0x9C5014B3	0xB85024BF	..+..-............+-........++..	0xC5658192	0xE165818E	..+..-.....................-++..
04	0x83593CD1	0x235B3CD2	-.+...........+...............+-	0x6C1FBD5D	0x801FBD49	+--.--.....................-.-..
05	0xC6190130	0xB7590110	.-++...+.+................-.....	0x8C04A87A	0x8004A878	....--........................-.
06	0x6E00C7DF	0xEDFEC7BC	+.....-++++++++..........-+...--	0x6798D20C	0xA798D21C	+-.........................+....
07	0x2BEF8A70	0xCBEF8A53	++-.......................-...++	0x8EF1D3D7	0x3AF1D3CB	-.++.-.....................-+-..
08	0x6F4FE90D	0x2F4FE910	.-.........................+--.-	0x5767A73D	0x7B67A739	..+.+-.......................-..
09	0x183DD233	0x783DD252	.++......................+-....-	0x1BFFA0C5	0xA7FFA0DD	+.+--+.....................++...
10	0x4A8E920B	0x6A8E9608	..+..................+........--	0x2DC4048D	0x9DC4049D	+.-+.......................+....
11	0xD06CE415	0x106CE3F6	--...................-+++++...+-	0x43F7CEBB	0x43F7CEB7	............................-+..
12	0x0A7225F2	0xAA7225EA	+.+........................-+...	0x3CBC5639	0x84BC5629	+.---......................-....
13	0x177B3CDE	0xD77B3CDD	++............................-+	0xD6FF95E0	0xDEFF95F8	....+......................++...
14	0xB78CC0C7	0x578CC0C7	-+-.............................	0xA7A7C3B0	0xDFA7C3A0	.+-++......................-....
15	0x287E041E	0xC87E041D	++-...........................-+	0xBFF4FE02	0xB7F4FE16	....-......................+.+..
16	0x23EFFDA4	0xA3EFFDA4	+...............................	0x8F5B9055	0xFF5B9045	.+++.......................-....
17	0x985D131F	0x785D131F	-++.............................	0xADFFD44D	0x15FFD451	-.-+-......................+--..
18	0x68130225	0x68130225	....	0xFEEFCBBB	0x16EFCBBB	---.-...........................
19	0x54114D3B	0x74114D3B	..+.............................	0x0B9AEE0D	0xBB9AEE09	+.++.........................-..
20	0xD3628D64	0x53628D64	-...............................	0x3589DA4B	0x6D89DA5B	.+.-+......................+....
21	0x01B43B64	0x61B43B64	.++.............................	0x55CA4BEC	0xE5CA4BE0	+.+-........................--..
22	0x3DA50777	0x3DA50777	....	0x3984B119	0x7184B119	.+..-...........................
23	0x976D179D	0x976D179D	....	0x3EE6AEB6	0x8EE6AEB6	+.--............................
24	0x79788C4E	0xF9788C4E	+...............................	0x4064F013	0xF864F003	+.+++......................-....

Table 6. Amplified message pair