Classification and Generation of Disturbance Vectors for Collision Attacks against SHA-1

Stéphane Manuel[1]

CRI - Paris Rocquencourt
stephane.manuel@inria.fr

Abstract. In this paper, we present a deterministic algorithm to produce disturbance vectors for collision attacks against SHA-1. We show that all published disturbance vectors can be classified into two types of vectors, *type-I* and *type-II*. We define a cost function, close to those described in [9], to evaluate the complexity of a collision attack for a given disturbance vector. Using the classification and the cost function we made an exhaustive search which allowed us to retrieve all known vectors. We also found new vectors which have lower cost. This may lead to the best collision attack against SHA-1, with a theoretical attack complexity of 2^{51} hash function calls.

Key words: Hash Functions, SHA-1, Collision Attack, Disturbance Vector.

1 Introduction

SHA-1 has been a widely used hash function since it was published by NIST as a Federal Processing Standard in 1995 [11]. SHA-1 is an evolution of a previous standard named SHA-0 [10]. SHA-1 and SHA-0 only differ in their message expansion.

Many researches have discussed collision attacks against SHA-0 and SHA-1 [5, 1, 2, 15–17, 3, 14, 4, 6, 8]. Chabaud and Joux [5] pointed out the weakness of the state update transformation common to SHA-0 and SHA-1. They described a linear differential path composed of interleaved 6-steps local collisions. The core of this differential path is represented by a disturbance vector (so-called L-characteristic) which indicates where the 6-step local collisions are initiated. Once a disturbance vector is chosen, one can evaluate the complexity of a collision attack against SHA-0 or SHA-1 directly from this vector. The critical factor for choosing a disturbance vector is considered to be the Hamming weight of its last 60 coordinates. A lot of work has been spent in order to find good vectors [16, 7, 13, 12, 18]. The algorithms proposed are essentially probabilistic algorithms based on coding theory tools.

This article presents a new algorithm able to produce efficient disturbance vectors for collision attacks against SHA-1. Based on the experiments done using this algorithm, we present a classification for these vectors. First, we will describe our algorithm and present what we will call type-I and type-II classes. We then show that all the previously proposed and/or used disturbance vectors can be classified into one of these classes. We define a cost function in order to compare known vectors, and show that these vectors are not optimal. We therefore introduce a new vector which is optimal according to this cost function. This

The author is supported in part by the french Agence Nationale de la Recherche under the project designation EDHASH, DPg/ANR-CI FA/VB 2007-010.

vector may be used in order to build a collision attack against SHA-1 with a theoretical complexity of 2^{51} hash function calls.

We organized the paper as follows. In Section 2, we give a brief description of SHA-1. Then, in Section 3, we describe our new algorithm, define type-I and type-II families and show that all known vectors are elements of these classes. In Section 4, we define the cost function we used in order to evaluate the complexity of collision attacks against SHA-1. We then provide a new disturbance vector and give a complexity comparison with previous disturbance vectors. Finally, we draw conclusions in Section 5.

2 Short description of SHA-1

SHA-1 [11], is a 160-bit dedicated hash function based on the design principle of MD4. It applies the Merkle-Damgård paradigm to a dedicated compression function. The input message is padded and split into k 512-bit message blocks. At each iteration of the compression function h, a 160-bit chaining variable H_t is updated using one message block M_{t+1}, *i.e* $H_{t+1} = h(H_t, M_{t+1})$. The initial value H_0 (also called IV) is predefined and H_k is the output of the hash function.

The SHA-1 compression function is build upon the Davis-Meyer construction. It uses a function E as a block cipher with H_t for the message input and M_{t+1} for the key input, a feed-forward is then needed in order to break the invertibility of the process:

$$H_{t+1} = E(H_t, M_{t+1}) \boxplus H_t,$$

where $\boxplus$ denotes the addition modulo 2^{32} 32-bit words by 32-bit words. This function is composed of 80 steps (4 rounds of 20 steps), each processing a 32-bit message word W_i to update 5 32-bit internal registers (A, B, C, D, E). The feed-forward consists in adding modulo 2^{32} the initial state with the final state of each register. Since more message bits than available are utilized, a message expansion is therefore defined.

Message expansion. First, the message block M_t is split into 16 32-bit words $W_0, \ldots, W_{15}$. These 16 words are then expanded linearly, as follows:

$$W_i = (W_{i-16} \oplus W_{i-14} \oplus W_{i-8} \oplus W_{i-3}) \lll 1 \text{ for } 16 \leq i \leq 79.$$

State update. First, the chaining variable H_t is divided into 5 32-bit words to fill the 5 registers $(A_0, B_0, C_0, D_0, E_0)$. Then the following transformation is applied 80 times:

$$STEP_{i+1} := \begin{cases} A_{i+1} = (A_i \ll 5) + f_i(B_i, C_i, D_i) + E_i + K_i + W_i, \\ B_{i+1} = A_i, \\ C_{i+1} = B_i \gg 2, \\ D_{i+1} = C_i, \\ E_{i+1} = D_i. \end{cases}$$

where K_i are predetermined constants and f_i are boolean functions defined in Table 1.

round	step i	$f_i(B, C, D)$	K_i
1	$1 \leq i \leq 20$	$f_{IF} = (B \wedge C) \oplus (\overline{B} \wedge D)$	`0x5a827999`
2	$21 \leq i \leq 40$	$f_{XOR} = B \oplus C \oplus D$	`0x6ed6eba1`
3	$41 \leq i \leq 60$	$f_{MAJ} = (B \wedge C) \oplus (B \wedge D) \oplus (C \wedge D)$	`0x8fabbcdc`
4	$61 \leq i \leq 80$	$f_{XOR} = B \oplus C \oplus D$	`0xca62c1d6`

Table 1. Boolean functions and constants in `SHA-1`.

Feed-forward. The sums modulo 2^{32}: $(A_0 + A_{80})$, $(B_0 + B_{80})$, $(C_0 + C_{80})$, $(D_0 + D_{80})$, $(E_0 + E_{80})$ are concatenated to form the chaining variable H_{t+1}.

Note that all updated registers but A_{i+1} are just rotated copies, so we only need to consider the register A at each step. Thus, we have:

$$A_{i+1} = (A_i \ll 5) + f_i(A_{i-1}, A_{i-2} \gg 2, A_{i-3} \gg 2) + A_{i-4} \gg 2 + K_i + W_i. \qquad (1)$$

3 Disturbance vectors searching algorithm

3.1 Principle of the algorithm

The search algorithm we used is mainly based on the simple observation that the message expansion of `SHA-1` can be defined in two directions: forward expansion and backward expansion. Namely, one can fix $W_0, \ldots, W_{15}$ and then expand them forward to obtain $W_{16}, \ldots, W_{79}$:

– Forward expansion : $W_i = (W_{i-16} \oplus W_{i-14} \oplus W_{i-8} \oplus W_{i-3}) \lll 1$.

This is the standard way defined in `SHA-1` specifications. We can also expand backward to obtain $W_{-64}, \ldots, W_{-1}$ defined by:

– Backward expansion : $W_i = (W_{i+16} \ggg 1) \oplus W_{i+13} \oplus W_{i+8} \oplus W_{i+2}$.

Any sequence of 80 consecutive 32-bits words $W_i, \ldots, W_{i+79}$ with $-64 \leq i \leq 0$ is a valid expanded message. For all $W_0, \ldots, W_{15}$ we define an extended expanded message (EEM) consisting of 144 32-bits words:

$$W_{-64}, \ldots, W_{-1}, W_0, \ldots, W_{15}, W_{16}, \ldots, W_{79},$$

each EEM is composed of 65 valid expanded message. For all $W_0, \ldots, W_{15}$ of total binary Hamming weight of 3 or less, we generated the corresponding EEM and computed the cost of its 65 expanded messages.

We experimentally observed some facts:

– fact 1: all efficient vectors ”looked” similar in some way,
– fact 2: we were able to find all previously known vectors,
– fact 3: all efficient vectors mainly have their disturbances on low weight or heavy weight bits of the W_i,
– fact 4: lot of efficient vectors were the same under cyclic shift of their 32-bits words.

We remark here that some of these observations were already present in the literature. Rijmen and Oswald [13], noticed that the codewords they found have a large amount of W_i in common. Jutla and Patthak [7] indicated that their first codeword was earlier reported by Wang *et al.* in [16]. Pramstaller *et al.* [12] also pointed that their vector were the same disturbance vector as the one used by Wang *et al.* with a shifted version of the indices.

Considering fact 3 and fact 4 lead us to add some heuristics to our algorithm. With these heuristics, we were able to extand our search for efficient vectors to $W_0, \ldots, W_{15}$ of total binary Hamming weight of 6 or less. This new search confirmed the previous constated facts. The interpretation of these facts lead us to conclude that there were only two types of efficient vectors.

3.2 Classification of disturbance vectors

We say that two disturbances vectors are equivalent if:

– the vectors are globally invariant under cyclic shift of their 32-bits words $W_0, \ldots, W_{79}$ or,
– the vectors are generated by the same extented expanded message.

We experimentally verified that all efficient disturbance vectors lie in only two different classes. We note these classes type-I and type-II. The type-I class contains the vector first reported by Wang *et al.* in [16]. Type-II class contains the vector first reported as *Codeword*2 by Jutla and Patthak in [7]. Table 4 and Table 5 present in a synthetic way all previously known vectors and show that they are of type-I or type-II.

Whereas vectors of the same class are equivalent, the complexity of the associated collision attacks can vary. We discuss in the next section how we evaluated this complexity.

4 Complexity evaluation

4.1 Cost function

In order to compare the complexity of collision attacks based on different disturbance vectors we used a cost function. This cost function is based on *the probabilities for local collisions in* `SHA-1` described in Table 5 of the article proposed by Mendel *et al.* at FSE 2006 [9]. In the article of Wang *et al.* [16], conditions are counted from step 22 to 78. In section 3.5 of the article of Mendel *et al.*, the probabilities are computed, on the same disturbance vector, from step 22 to 76. That's why we arbitrarily chose to build our cost function on the basis of a probability computation from step 22 to 76. We also used the strict differential bit compression described by Yajima *et al.* [18]. Table 2 gives a complexity comparison based on this cost function, of all disturbance vectors we were aware of.

We are not claiming that this is the best way ever to evaluate the complexity of a collision attack against `SHA-1`. The article of Mendel *et al.* demonstrates that this is an estimation and that the accurate probability is lower than the estimation. Furthermore, Heuristics like advanced message modifications or neutral bit techniques modify the complexity computation. For example by allowing to start from steps higher than 22. Also, early stopping techniques can lower the global complexity of the attack. However, this *simple* cost function is a convenient way to make a raw comparison between proposed disturbance vectors.

We also point out that our algorithm may use any other cost function. This is just an evaluation function used to sort good candidates.

Disturbance vector	Complexity evaluation log
Wang *et al.* CRYPTO'05 [16]	66
Rijmen & Oswald CT-RSA 2005 [13]	
*Codeword*1	84
*Codeword*2	84
*Codeword*3	95
Jutla & Patthak Eprint 2005 [7]	
*Codeword*1	67
*Codeword*2	60
*Codeword*3	68
Pramstaller *et al.* IMA 2005 [12]	71
De Cannière *et al.* SAC 2007 [4]	79
Yajima *et al.* ASIACCS 2008 [18]	68

Table 2. Complexity comparison of known disturbance vectors.

4.2 New disturbance vector

Using our algorithm, we were able to find an optimal disturbance vector. It is a type-II vector which has a complexity evaluation of 2^{57} with respect to the cost function we defined. This vector is given in Table 3, with its estimated probability computation.

This vector may be used in a two block collision attack against SHA-1, the same way that have been done in previous attacks. One may use an automatic generator to produce the needed non-linear characteristics [3]. Then use boomerangs in the neutral bits framework [6] or advanced message modifications such as those given in [17] to start complexity evaluation from step 25. Assuming the above simplifications, which is arguable, we estimate the complexity of the attack to be close to 2^{51} hash function calls.

5 Conclusion

In this paper, we introduce a new algorithm to produce disturbance vectors to be used in collision attacks against SHA-1. By identifying two types of efficient disturbance vectors, we show that all known disturbance vectors can be classified into one of these types. We managed to find a better candidate that the previously proposed vectors. This new disturbance vector may be used as an L-characteristic for a collision attack against SHA-1. This attack has a theoretical complexity of 2^{51} hash function calls.

References

1. E. Biham and R. Chen. Near-Collisions of SHA-0. In M.K. Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 290–305. Springer-Verlag, 2004.
2. E. Biham, R. Chen, A. Joux, P. Carribault, C. Lemuet and W. Jalby. Collisions of SHA-0 and Reduced SHA-1. In R. Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 36–57. Springer-Verlag, 2005.
3. C. De Cannière and C. Rechberger. Finding SHA-1 Characteristics: General Results and Applications. In X. Lai and K. Chen, editors, *Advances in Cryptology – ASIACRYPT 2006*, volume 4284 of *Lecture Notes in Computer Science*, pages 1–20. Springer-Verlag, 2006.
4. C. De Cannière, F. Mendel and C. Rechberger. Collisions for 70-step SHA-1: On the Full Cost of Collision Search. in C. Adams and A. Miri and M. Wiener, editors, *Selected Areas in Cryptography – SAC 2007*, volume 4876 of *Lecture Notes in Computer Science*, pages 56–73. Springer-Verlag, 2007.
5. F. Chabaud and A. Joux. Differential Collisions in SHA-0. In H. Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 56–71. Springer-Verlag, 1998.
6. A. Joux and T. Peyrin. Hash Functions and the (Amplified) Boomerang Attack. In A. Menezes, editor, *Advances in Cryptology – CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 244–263. Springer-Verlag, 2004.
7. C.S. Jutla and A.C. Patthak. A Matching Lower Bound on the Minimum Weight of SHA-1 Expansion Code. *Cryptology ePrint Archive*, Report 2005/266, 2005. Available from: `http://eprint.iacr.org`.
8. S. Manuel and T. Peyrin. Collisions on SHA-0 in one hour. In K. Nyberg, editor, *Fast Software Encryption – FSE 2008*, volume 5086 of *Lecture Notes in Computer Science*, pages 16–35. Springer-Verlag, 2008.
9. F. Mendel, N. Pramstaller, C. Rechberger and V. Rijmen. The Impact Of Carries on the Complexity of Collision Attacks on SHA-1. In M.J.B. Robshaw, editor, *Fast Software Encryption – FSE 2006*, volume 4047 of *Lecture Notes in Computer Science*, pages 278–292. Springer-Verlag, 2006.
10. National Institute of Standards and Technology. FIPS 180: Secure Hash Standard, May 1993. Available from: `http://csrc.nist.gov`.
11. National Institute of Standards and Technology. FIPS 180-1: Secure Hash Standard, April 1995. Available from: `http://csrc.nist.gov`.
12. N. Pramstaller, C. Rechberger and V. Rijmen. Exploiting Coding Theory for Collision Attacks on SHA-1. in N.P. Smart, editor, *Cryptography and Coding 2005*, volume 3796 of *Lecture Notes in Computer Science*, pages 78–95. Springer-Verlag, 2005.
13. V. Rijmen and E. Oswald. Update on SHA-1. in A;J. Menezes, editor, *The Cryptographers'Track at the RSA conference – CT-RSA 2005*, volume 3376 of *Lecture Notes in Computer Science*, pages 58–71. Springer-Verlag, 2005.
14. M. Sugita, M. Kawazoe, L. Perret and H. Imai. Algebraic Cryptanalysis of 58-Round SHA-1. In A. Biryukov, editor, *Fast Software Encryption – FSE 2007*, volume 4593 of *Lecture Notes in Computer Science*, pages 349–365. Springer-Verlag, 2007.
15. X. Wang, H. Yu and Y.L. Yin. Efficient Collision Search Attacks on SHA-0. In V. Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 1–16. Springer-Verlag, 2005.
16. X. Wang, Y.L. Yin, and H. Yu. Finding Collisions in the Full SHA-1. In V. Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 17–36. Springer-Verlag, 2005.
17. X. Wang, Y.L. Yin, and H. Yu. New Collision Search for SHA-1. In Proceedings of *NIST Cryptographic Hash Workshop*, 2005. Available from: `http://csrc.nist.gov`.

18. J. Yajima, T. Iwasaki, Y. Naito, Y. Sasaki, T. Shimoyama, N. Kunihiro and K. Ohta. A Strict Evaluation Method on the Number of Conditions for the SHA-1 Collision Search. In proceedings *ASIACCS'08* March 18-20, Tokyo, Japan, 2008.

i	DV_i	Complexity evaluation
0:	oo-o----------------------------	–
1:	o-------------------------------	–
2:	ooo-----------------------------	–
3:	o-o-----------------------------	–
4:	---o----------------------------	–
5:	o-------------------------------	–
6:	ooo-----------------------------	–
7:	--o-----------------------------	–
8:	oo------------------------------	–
9:	o-------------------------------	–
10:	-oo-----------------------------	–
11:	o-------------------------------	–
12:	oo------------------------------	–
13:	o-------------------------------	–
14:	o-o-----------------------------	–
15:	o-------------------------------	–
16:	ooo-----------------------------	–
17:	--------------------------------	–
18:	--o-----------------------------	–
19:	o-------------------------------	–
20:	-oo-----------------------------	–
21:	--------------------------------	0
22:	--------------------------------	0
23:	o-------------------------------	3
24:	o-------------------------------	3
25:	--------------------------------	0
26:	--------------------------------	0
27:	--------------------------------	0
28:	--------------------------------	0
29:	--------------------------------	0
30:	o-------------------------------	3
31:	--------------------------------	0
32:	o-------------------------------	3
33:	--------------------------------	0
34:	o-------------------------------	3
35:	--------------------------------	0
36:	oo------------------------------	4
37:	--------------------------------	0
38:	--------------------------------	0
39:	o-------------------------------	3
40:	--------------------------------	0
41:	--------------------------------	0
42:	--------------------------------	0
43:	--------------------------------	0
44:	o-------------------------------	3
45:	--------------------------------	0
46:	--------------------------------	0
47:	--------------------------------	0
48:	--------------------------------	0
49:	--------------------------------	0
50:	o-------------------------------	3
51:	--------------------------------	0
52:	o-------------------------------	3
53:	--------------------------------	0
54:	--------------------------------	0
55:	--------------------------------	0
56:	--------------------------------	0
57:	--------------------------------	0
58:	--------------------------------	0
59:	--------------------------------	0
60:	--------------------------------	0
61:	--------------------------------	0
62:	--------------------------------	0
63:	--------------------------------	0
64:	-------------------------------o	4
65:	--------------------------------	0
66:	--------------------------------	0
67:	------------------------------o-	2
68:	-------------------------------o	4
69:	--------------------------------	0
70:	-----------------------------o--	4
71:	------------------------------o-	2
72:	------------------------------o-	2
73:	----------------------------o---	4
74:	-----------------------------o--	4
75:	--------------------------------	0
76:	---------------------------o--o-	–
77:	----------------------------o---	–
78:	----------------------------o-o-	–
79:	--------------------------o-----	–

Table 3. New disturbance vector.

DV_i	Wang *et al.* [16, 17, 14, 6]	Rijmen & Oswald [13]			Jutla & Patthak [7]		Pramstaller *et al.* [12]
	$\ggg 2$	$\ggg 1$			*Codeword*1	*Codeword*3	$\ggg 2$
⋮							
--------------------------------					0		
-oo-----------------------------					1	0	
--------------------------------					2	1	0
-o-o----------------------------	0				3	2	1
o-------------------------------	1				4	3	2
o-------------------------------	2				5	4	3
o-o-----------------------------	3				6	5	4
-o------------------------------	4	0			7	6	5
--------------------------------	5	1			8	7	6
-oo-----------------------------	6	2			9	8	7
o-------------------------------	7	3			10	9	8
o-------------------------------	8	4	0		11	10	9
o-------------------------------	9	5	1		12	11	10
--------------------------------	10	6	2	0	13	12	11
--------------------------------	11	7	3	1	14	13	12
-o------------------------------	12	8	4	2	15	14	13
--------------------------------	13	9	5	3	16	15	14
o-o-----------------------------	14	10	6	4	17	16	15
o-------------------------------	15	11	7	5	18	17	16
o-o-----------------------------	16	12	8	6	19	18	17
--------------------------------	17	13	9	7	20	19	18
o-------------------------------	18	14	10	8	21	20	19
--------------------------------	19	15	11	9	22	21	20
oo------------------------------	20	16	12	10	23	22	21
--------------------------------	21	17	13	11	24	23	22
o-------------------------------	22	18	14	12	25	24	23
o-------------------------------	23	19	15	13	26	25	24
-o------------------------------	24	20	16	14	27	26	25
--------------------------------	25	21	17	15	28	27	26
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
--------------------------------	61	57	53	51	64	63	62
--------------------------------	62	58	54	52	65	64	63
--------------------------------	63	59	55	53	66	65	64
-------------------------------o	64	60	56	54	67	66	65
--------------------------------	65	61	57	55	68	67	66
--------------------------------	66	62	58	56	69	68	67
------------------------------o-	67	63	59	57	70	69	68
--------------------------------	68	64	60	58	71	70	69
--------------------------------	69	65	61	59	72	71	70
-----------------------------o--	70	66	62	60	73	72	71
--------------------------------	71	67	63	61	74	73	72
------------------------------o-	72	68	64	62	75	74	73
----------------------------o---	73	69	65	63	76	75	74
--------------------------------	74	70	66	64	77	76	75
--------------------------------	75	71	67	65	78	77	76
---------------------------o----	76	72	68	66	79	78	77
--------------------------------	77	73	69	67		79	78
----------------------------o-o-	78	74	70	68			79
--------------------------o-----	79	75	71	69			
-----------------------------oo-		76	72	70			
--------------------------------		77	73	71			
-------------------------o------		78	74	72			
----------------------------o---		79	75	73			
--------------------------o-o---			76	74			
------------------------o-------			77	75			
----------------------------o---			78	76			
--------------------------------			79	77			
-----------------------o----o---				78			
--------------------------------				79			
⋮							

Table 4. Known disturbance vectors of type-I.

DV_i	Yajima *et al.* [18] $\ggg 2$	De Cannière *et al.* [4] $\ggg 2$	Jutla & Patthak [7] *Codeword*2
⋮			
oo-o----------------------------	0		
o-------------------------------	1		
ooo-----------------------------	2		
o-o-----------------------------	3		
---o----------------------------	4		
o-------------------------------	5		0
ooo-----------------------------	6		1
--o-----------------------------	7		2
oo-o----------------------------	8		3
o-------------------------------	9		4
ooo-----------------------------	10		5
o-o-----------------------------	11	0	6
---o----------------------------	12	1	7
o-------------------------------	13	2	8
ooo-----------------------------	14	3	9
--o-----------------------------	15	4	10
oo------------------------------	16	5	11
o-------------------------------	17	6	12
-oo-----------------------------	18	7	13
o-------------------------------	19	8	14
oo------------------------------	20	9	15
o-------------------------------	21	10	16
o-o-----------------------------	22	11	17
o-------------------------------	23	12	18
ooo-----------------------------	24	13	19
--------------------------------	25	14	20
⋮	⋮	⋮	⋮
--------------------------------	61	51	57
--------------------------------	62	52	58
--------------------------------	63	53	59
--------------------------------	64	54	60
--------------------------------	65	55	61
--------------------------------	66	56	62
--------------------------------	67	57	63
--------------------------------	68	58	64
--------------------------------	69	59	65
--------------------------------	70	60	66
-------------------------------o	71	61	67
--------------------------------	72	62	68
--------------------------------	73	63	69
------------------------------o-	74	64	70
-------------------------------o	75	65	71
--------------------------------	76	66	72
-----------------------------o--	77	67	73
------------------------------o-	78	68	74
------------------------------o-	79	69	75
----------------------------o---		70	76
-----------------------------o--		71	77
--------------------------------		72	78
---------------------------o--o-		73	79
----------------------------o---		74	
----------------------------o-o-		75	
--------------------------o-----		76	
---------------------------o-oo-		77	
--------------------------------		78	
--------------------------o--o-o		79	
⋮			

Table 5. Known disturbance vectors of type-II.