An Improved Preimage Attack against HAVAL-3*

Jian Guo[1], Chunhua Su[2], and Wun-She Yap[3]

[1] Nanyang Technological University, Singapore
[2] Japan Advanced Institute of Science and Technology, Japan
[3] Universiti Tunku Abdul Rahman, Malaysia

ntu.guo@gmail.com, suchunhua@gmail.com, yapws@utar.edu.my

Abstract. Hash functions play an important role in constructing cryptographic schemes that provide security services, such as confidentiality in an encryption scheme, authenticity in an authentication protocol and integrity in a digital signature scheme and so on. Such hash function is needed to process a challenge, a message, an identifier or a private key. In this paper, we propose an attack against HAVAL-3 hash function, which is used in open source Tripwire and is included in GNU Crypto. Under the meet-in-the-middle (MITM) preimage attack framework proposed by Aoki and Sasaki in 2008, the one-wayness of several (reduced-) hash functions had been broken recently. However, most of the attacks are of complexity close to brute-force search. Focusing on reducing the time complexity of such MITM attacks, we improve the preimage attacks against HAVAL-3 hash function to with lower time complexity and memory requirement, compared with the best known attack proposed by Sasaki and Aoki in ASIACRYPT 2008. Besides the 256-bit variant of HAVAL-3, similar improvements can be applied to some truncated variants as well. Interestingly, due to the low complexity of our attack, the preimage attack applies to the 192-bit variant of HAVAL-3 for the first time.

Keywords: Hash function, HAVAL-3, cryptanalysis, meet-in-the-middle attack, splice-and-cut.

1 Introduction

A hash function H which is used in cryptography maps a message M of arbitrary length to a short fixed-length hash value h. It is a fundamental component of many cryptographic protocols and applications used in electronic devices such as digital signatures, random number generation in embedded chips, especially for RFID authentication system. To fulfil the security needs from these applications, there are three basic security requirements for hash functions of digest size n bits, *i.e.*,

Collision Resistance: it should be computationally difficult, up to a bound of $2^{n/2}$, to find (M, M') with $M \neq M'$ such that $H(M) = H(M')$.

Second-Preimage Resistance: given a message M, it should be computationally difficult, up to a bound of 2^{n-k} for message of 2^k blocks, to find M' with $M \neq M'$ such that $H(M) = H(M')$.

Preimage Resistance: given a target T, it should be computationally difficult, up to a bound of 2^n, to find M such that $H(M) = T$.

In this paper, we focus on preimage resistance. There are many known attack methods against the hash functions. We pay our attention to the splice-and-cut meet-in-the-middle preimage attack against cryptographic hash function HAVAL-3 designed by Zheng *et al.* in 1992 and improve the previous best time complexity of such attack. We note that HAVAL-3 has already many applications. For example, HAVAL is used in open source Tripwire, a free software security and data integrity tool useful for monitoring and alerting on specific file changes on a range of systems.

The meet-in-the-middle (MITM) attack was originated from an attack against double-DES by Diffie and Hellman [9], where they found double-DES (*i.e.*, encrypting a plaintext P using DES twice under different

* Appears in *Information Processing Letters 115 (2015), pp. 386-393.* A few errors from [12] have been corrected here. This document was last updated on 08 Nov 2014.

keys to produce a ciphertext $C = \texttt{DES}_{k2}(\texttt{DES}_{k1}(P)))$ did not provide security as expected, since one can compare $\texttt{DES}_{k2}^{-1}(C)$ and $\texttt{DES}_{k1}(P)$ to check whether both outputs match in the middle. One can view this attack works at the level of operating modes since the specification of DES do not play a crucial role in such a generic attack. Similar MITM attacks on operating modes of hash functions were presented in 1992 by Lai and Massey [22]. Later, this attack was generalized to analyze the security of block ciphers, and had been applied to round-reduced AES [7], DES [11] and IDEA [8] block ciphers.

In the CRYPTO 2008 rump session, Sasaki and Aoki [29] presented a MITM preimage attack framework to attack the compression functions of Davies-Meyer hash functions, named *splice-and-cut* MITM attacks. With a generic unbalanced MITM approach, one can convert these pseudo-preimages on compression functions to preimages of hash functions. For a given hash h and a compression function CF, pseudo-preimage is a pair of (v, M), $v \neq$ initial value IV, such that $CF(v, M) = h$. If we can construct many distinguished pairs (H_{i-1}, M_i), such that all produce the same value as the given target h, followed by a message block search, which links the IV to one of the H_{i-1}s, then a full preimage of the hash function is found. Subsequently, this framework has been applied to many narrow-pipe designs such as full or round-reduced MD4 [13], MD5 [30], SHA-0/1 [2, 21], SHA-2 [1], HAS-160 [27, 17], HAVAL [28, 25], RIPEMD [31, 34], Tiger [13] and also some SHA-3 candidates [19].

Many useful techniques have been developed through these attacks. In general, there are two streams, one aims to attack more steps, since MITM is generally difficult to achieve for full hash functions due to multiple passes and key schedules. On the other hand, most time complexities of such attacks are very close to brute-force search, thus few techniques have been developed to reduce the time complexities [1, 13].

Interestingly, these techniques developed on hash functions are found to be useful to analyze some existing block ciphers such as XTEA [33], also for very recent designs such as KTANTAN [4, 37].

In 2011, besides the Davies-Meyer construction, the MITM preimage framework is also applied to other modes of operation [26], with example of AES in different hashing modes. This attack is special since it does not use any key words (or message words in hash functions) as neutral words, but only some state values.

Another research line on finding preimages is to explore special properties and dedicated algorithms for both finding pseudo-preimages and conversion to preimages, *e.g.*, [23, 3, 5]. However, it is interesting to note that both attacks on MD4 [23] and HAVAL-3 [3] can also be re-explained under the MITM preimage framework. Yet, the dedicated algorithms work much faster when converting pseudo-preimages to preimages.

Besides attacks against primitives like hash functions and block ciphers, variants of MITM attacks have been used to analyze applications of these primitives, such as message authentication codes (MACs) based on block ciphers [38], and MACs based on hash functions [36, 35, 15, 14].

1.1 Our Contribution.

We observe that there are some new techniques, in particular `indirect partial matching` (IPM) from [1] and `multi-target pseudo-preimages` (MTPP) from [13], developed after the best known preimage attacks against HAVAL-3 were found in [25]. We apply some of these techniques together with the MITM preimage framework to HAVAL-3 and improve the attack complexities in terms of time and memory. These improvements apply to not only the 256-bit variant, but also some truncated variants of HAVAL-3 including 224-bit and 192-bit. Interestingly, due to the low time complexity of the attack, this is the first preimage attack against the 192-bit variant of HAVAL-3. A detailed comparison of our results and exsiting (pseudo-)preimage attacks against HAVAL-3 is shown in Table 1.

1.2 Organization.

The rest of the paper is organized as follows. Section 2 describles the details of HAVAL. Section 3 gives an introduction on the details of MITM preimage attacks. We apply the MITM preimage attack to HAVAL-3 and to its small variants in Section 4. Finally, we conclude and give some open problems in Section 5.

Table 1. Comparison on existing and our preimage attacks against some variants of HAVAL-3

	source	256-bit	224-bit	192-bit
Pseudo-preimage	Aumasson et al. [3]	2^{224}	-	-
	Sasaki et al. [28, 25]	2^{196}	2^{160}	2^{128}
	Ours	2^{160}	2^{135}	2^{112}
Preimage	Aumasson et al. [3]	2^{230}	-	-
	Sasaki et al. [28, 25]	2^{225}	2^{209}	-
	Sasaki et al. [32][d]	2^{244}	-	-
	Ours	2^{209}	$2^{196.5}$	2^{185}

2 Description of HAVAL-3

HAVAL [39] is a hash function family designed by Zheng *et al.* in 1992, which compresses a message up to $2^{64}-1$ bits into 128, 160, 192, 224, 256 bits digest (a.k.a. hash h). It consists of three versions for each hash function with 3, 4, 5 passes (HAVAL-x denotes the version with x passes), following Merkle-Damgård structure. A message is padded by a '1' and many '0's so that the length becomes 944 mod 1024. Then a 3-bit version number, 3-bit indicating number of passes used, 10-bit output length, 64-bit original message length are padded, the final length is multiple of 1024. Then the message is divided into blocks of 1024 bits $(B_0, B_1, \ldots, B_{n-1})$, and the hash can be computed from the initial value (IV) and compression function $(CF : \{0,1\}^{256} \times \{0,1\}^{1024} \to \{0,1\}^{256})$:

1. $H_0 \leftarrow IV$,
2. $H_{i+1} \leftarrow CF(H_i, B_i)$ for $0 \le i < n$.

H_n is the final hash output of the 256-bit variant. For each message block B_j, the block is divided into 32 words $M_0, M_1, \ldots, M_{31}$, and the compression function proceeds as follows:

1. $p_0 \leftarrow H_j$
2. $p_{i+1} \leftarrow R_i(p_i, M_{\pi(i)})$ for $0 \le i < 32x$ for x-pass version ($x = 3, 4, 5$). Here R_i is round function at step i and π is the message permutation as defined in Table 3.
3. $H_{j+1} \leftarrow p_0 + p_{32x}$, here '+' is word-wise addition modulo 2^{32}.

We denote Q_j, so that $p_j = Q_{j-7}||Q_{j-6}||Q_{j-5}||Q_{j-4}||Q_{j-3}||$
$Q_{j-2}||Q_{j-1}||Q_j$. $Q_{j+1} \stackrel{\text{def}}{=} (Q_{j-7} \ggg 11) + (F_j(Q_{j-6}, Q_{j-5},$
$Q_{j-4}, Q_{j-3}, Q_{j-2}, Q_{j-1}, Q_j) \ggg 7) + M_{\pi(j)} + K_{x,j}$, where $K_{x,j}$ are predefined constants and we ignore the details since these constants are not important to our attacks. F_j is composed as $f_j \circ \phi_{x,j}$, where $\phi_{x,j}$ are permutations defined in Table 2 and f_j are bitwise Boolean function defined in Equation 1.

$$
\begin{aligned}
0 \le j \le 31: \quad & f_j(x_6, x_5, x_4, x_3, x_2, x_1, x_0) \\
& = x_1x_4 \oplus x_2x_5 \oplus x_3x_6 \oplus x_0x_1 \oplus x_0. \\
32 \le j \le 63: \quad & f_j(x_6, x_5, x_4, x_3, x_2, x_1, x_0) \\
& = x_1x_2x_3 \oplus x_2x_4x_5 \oplus x_1x_2 \oplus x_1x_4 \oplus \\
& \quad x_2x_6 \oplus x_3x_5 \oplus x_4x_5 \oplus x_0x_2 \oplus x_0. \\
64 \le j \le 95: \quad & f_j(x_6, x_5, x_4, x_3, x_2, x_1, x_0) \\
& = x_1x_2x_3 \oplus x_1x_4 \oplus x_2x_5 \oplus x_3x_6 \oplus x_0x_3 \oplus \\
& \quad x_0.
\end{aligned}
\tag{1}
$$

Table 2. Definitions of word-wise permutations $\phi_{x,j}$.

	x_6	x_5	x_4	x_3	x_2	x_1	x_0
	↓	↓	↓	↓	↓	↓	↓
$\phi_{3,1}$	x_1	x_0	x_3	x_5	x_6	x_2	x_4
$\phi_{3,2}$	x_4	x_2	x_1	x_0	x_5	x_3	x_6
$\phi_{3,3}$	x_6	x_1	x_2	x_3	x_4	x_5	x_0

Table 3. Message Expansion of HAVAL-3.

pass 1	0	1	2	3	4	5	6	7
	8	9	10	11	12	13	14	15
	16	17	18	19	20	21	22	23
	24	25	26	27	28	29	30	31
pass 2	5	14	26	18	11	28	7	16
	0	23	20	22	1	10	4	8
	30	3	21	9	17	24	29	6
	19	12	15	13	2	25	31	27
pass 3	19	9	4	20	28	17	8	22
	29	14	25	12	24	30	16	26
	31	15	7	3	1	0	18	27
	13	6	21	10	23	11	5	2

3 Description of the MITM Attacks

3.1 The Simple Version

Davies-Meyer construction refers to a method to construct a hash compression function from a block cipher by computing $CV' = E_B(CV) \oplus CV$, where $\oplus$ can be an exclusive-or, an addition or some other operations, CV and CV' denote the input and output respectively, $E_B(CV)$ denotes the ciphertext of encrypting a plaintext CV by the block cipher E under the key of message block B. The basic MITM pseudo-preimage attack works as in Fig. 1.

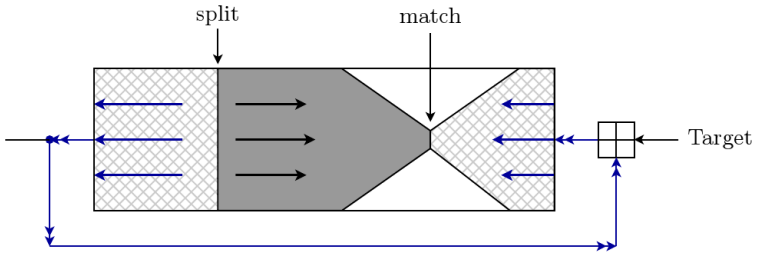

Fig. 1. MITM Pseudo-Preimage Attacks against Davies-Meyer Compression Functions.

1. Split the compression function into two chunks, and find some words/bits W_p and W_q not used in the other chunk, we call these words as neutral words.
2. Starting from the splitting point, fix to random values all state and message words except for W_p and W_q, and compute forward and backward utilizing all choices of W_q and W_p to get two lists L_q and L_p, independently.
3. Carry out match and filter out false choices. Note here only a partial matching is necessary at first, after which good candidates can be checked exhaustively for full match.

4. Repeat above process until a full match is found, which directly gives a pseudo-preimage.

To convert pseudo-preimages to preimages, one can repeat the above attack to get a few pseudo-preimages, and then find a message which links the IV to one of the pseudo-preimages. Without loss of generality, assume the sizes of W_p and W_q are l, and m bits are matched in step (iii). With computation of 2^l, one can compute 2^{2l} pairs and on average 2^{2l-m} good candidates left for further checking (in order not to let this part dominate the time complexity, usually $2l - m \leq l$ is required, *i.e.*, $m \geq l$). Hence, pseudo-preimage can be found in 2^{n-l}. One computes $2^{l/2}$ pseudo-preimages, then a linking message can be found in $2^{n-l/2}$ by random choices. Hence the overall complexity willl be $2^{n-l/2+1}$.

3.2 Initial Structures

Recently, cryptanalysts have replaced the starting state S, which can be at any step of the compression function, with a sophisticated construction called the initial structure [30, 2]. The initial structure can be informally defined as an overlapping of chunks, where neutral bits, although formally belonging to both chunks, are re-ordered so that they are involved in computation of the proper chunk only. It is found to be useful for significantly reducing attack complexity for the first. On the other hand, to improve the time complexity of a MITM preimage attack, the attackers usually need to find more neutral words.

Since dividing a whole compression function into two independent chunks is not easy in general, initial structure technique was invented to attack more steps by swapping message words around, in order to maintain the independence. As illustrated in Fig. 2, there are W_p and W_q originally located in chunk q and chunk p, which destroys the independence. Through the initial structure technique, one can swap the W_p and W_q so that the independence is re-established. Note this technique is useful to swap words close to the splitting point. Usually the swap can only be done under certain constraints and the attacker has to control some state bit values at the splitting point.

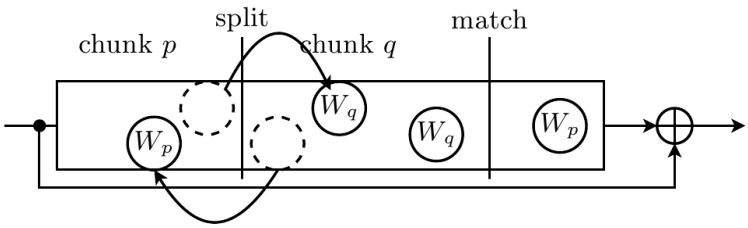

Fig. 2. Initial Structure in the MITM Preimage Attacks.

3.3 Indirect Partial Matching

The indirect partial matching [1] was invented also to extend number of attacked steps. The partial-matching technique makes use of the property that the state update function does not update all chaining variables of an internal state at each step, and usually updates only one word and leaves other words unchanged in a generalized Feistel network. In such case, several steps between two independent message chunks can be skipped, and yet some common state bits values are known in order to carry out the match. If the computation of a matching point can further be decomposed into a sum of independent functions of the neutral words, the indirect partial matching technique can be applied.

As illustrated in Fig. 3, when there are neutral words mixed at point close to the matching point, one needs not to stop the independent computation immediately. Instead, a few bits/words of the state can be still computed after few more steps assuming no knowledge on the values of the neutral words from the opposite chunk. These bits/words are used to carry out the match, instead of the full state. One can actually

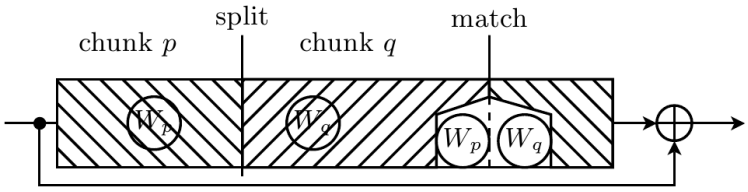

Fig. 3. Partial Matching in the MITM Preimage Attacks.

denote these bits in chunk p as a function of W_p f_p, similarly f_q for chunk q. Hence the MITM is to find matches for f_p of W_p and f_q of W_q.

Although the independence at the matching point is stopped by the neutral words from the opposite chunks, sometimes the matching bits can be expressed as $f_p(W_p)+\psi_p(W_q)$ and $f_q(W_q)+\psi_q(W_p)$. Instead of matching directly, one can compute $f_p(W_p)-\psi_q(W_p)$ and $f_q(W_q)-\psi_p(W_q)$ independently, then match. This is called indirect partial matching, and usually it extends partial matching for 2 steps with details shown later.

Assume one knows $p_j = Q_{j-7}||Q_{j-6}||Q_{j-5}||Q_{j-4}||Q_{j-3}||Q_{j-2}||Q_{j-1}||Q_j$, and note that $p_{j+7} = Q_j||Q_{j+1}||Q_{j+2}||Q_{j+3}||Q_{j+4}||Q_{j+5}||Q_{j+6}||Q_{j+7}$, where Q_j have not been modified 7 steps later from p_j to p_{j+7}. One can use these properties to partially match one register for two state words at most 7 steps away from each other. Indirect partial matching extends it to 9 steps. More generally, one can find r registers unmodified after $8-r$ steps, and indirectly match these r words for $10-r$ steps.

4 Improved MITM Attacks against HAVAL-3

4.1 Results by Sasaki et al.

In [25], Sasaki presented the best preimage attacks against HAVAL-3 known so far using the MITM preimage attacks, besides the one-block preimage attacks in [32]. The separation and neutral words choices are shown in Table 4. They used M_5 and M_{11} as the neutral word W_p, M_0 and M_1 as the neutral word W_q. The splitting point is between step 39 and 40, and the partial matching worked for 5 steps, *i.e.*, step $1, 0, 95, 94, 93$.

Table 4. Separation of neutral words of HAVAL-3 in the attack by Sasaki [25]

Step	0	1	2	3	4	5	6	7	8	9	10	11	12	13	$\cdots$	21	22	23	24	25	26	27	28	29	30	31
index	⓪	①	2	3	4	⑤	6	7	8	9	10	⑪	12	13	$\cdots$	21	22	23	24	25	26	27	28	29	30	31
	skip		first chunk																								
Step	32	33	34	35	36	37	38	39	40	41	42	43	44	45	$\cdots$	53	54	55	56	57	58	59	60	61	62	63
index	⑤	14	26	18	⑪	28	7	16	⓪	23	20	22	①	10	$\cdots$	24	29	6	19	12	15	13	2	25	31	27
	first chunk								second chunk																		
Step	64	65	66	67	68	69	70	71	72	73	74	75	$\cdots$	83	84	85	86	87	88	89	90	91	92	93	94	95
index	19	9	4	20	28	17	8	22	29	14	25	12	$\cdots$	3	①	⓪	18	27	13	6	21	10	23	⑪	⑤	2
	second chunk																							skip		

4.2 Our Improvements

To increase the success probability of the attacks against HAVAL-3 based cryptographic schemes, we have to reduce the time complexity of the attack significantly first. To reduce the time complexity for this attack,

larger neutral words will be needed, and in the meanwhile sufficiently many bits should be matched during the partial match stage, so that the final full-match will not dominate the time complexity. Currently $8-5=3$ words (96 bits) are matched. If one needs more neutral words, it will probably appear in the partial matching stage. At least one more word has to be added to each of W_p and W_q, then partial matching has to bypass 7 steps, and only $8-7=1$ word can be matched, which becomes insufficient. This is probably the difficulty faced in [25].

When indirect partial matching is considered, $10-7=3$ words can be matched for 7 steps, which is just enough. With this insight, we carry out the search of good neutral words pairs with the following criteria.

- There should be at least 3 words for both W_p and W_q.
- At most 7 steps can be bypassed by the indirect partial matching.

While the neutrality should be maintained, we found the only solution $W_p = (M_5, M_{11}, M_{23})$ and $W_q = (M_0, M_1, M_2)$. Chunk p covers steps $3-39$, chunk q covers steps $42-91$, initial structure covers steps $40, 41$ and indirect partial matching covers steps $92-95, 0-2$, as demonstrated in Table 5.

Table 5. Separation of neutral words of our improved MITM preimage attacks.

Step	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Index	⓪	①	②	3	4	⑤	6	7	8	9	10	⑪	12	13	14	15
	IPM			chunk p												
Step	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Index	16	17	18	19	20	21	22	㉓	24	25	26	27	28	29	30	31
	chunk p															
Step	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
Index	⑤	14	26	18	⑪	28	7	16	⓪	㉓	20	22	①	10	4	8
	chunk p								IS		chunk q					
Step	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Index	30	3	21	9	17	24	29	6	19	12	15	13	②	25	31	27
	chunk q															
Step	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
Index	19	9	4	20	28	17	8	22	29	14	25	12	24	30	16	26
	chunk q															
Step	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
Index	31	15	7	3	①	⓪	18	27	13	6	21	10	㉓	⑪	⑤	②
	chunk q												IPM			

Initial Structure. The purpose of the initial structure is to derive an equivalent function when moving some message words, i.e., the same output will be produced when the input is the same. This is sometimes only possible when the attacker has control over some intermediate state values. As demonstrated in Fig. 4, one can move M_0 from step 40 to step 41, if the value of Q_{41} does not influences the result of F at step 41. Note

$$\begin{aligned}
&F_{41}(Q_{35}, Q_{36}, Q_{37}, Q_{38}, Q_{39}, Q_{40}, Q_{41}) \\
=\ &f_{41}(Q_{37}, Q_{39}, Q_{40}, Q_{41}, Q_{36}, Q_{38}, Q_{35}) \\
=\ &Q_{36}Q_{38}Q_{41} \oplus Q_{36}Q_{39}Q_{40} \oplus Q_{36}Q_{38} \oplus Q_{38}Q_{40} \oplus \\
&Q_{36}Q_{37} \oplus Q_{39}Q_{41} \oplus Q_{39}Q_{40} \oplus Q_{35}Q_{36} \oplus Q_{35}.
\end{aligned}$$

When we set $Q_{38} = Q_{39} = 0$ so that $Q_{36}Q_{38}Q_{41} = Q_{39}Q_{41} = 0$ and the result of F_{41} becomes independent of Q_{41}, note we have the freedom to set all the chaining values at the cutting line, i.e., the 8 registers including Q_{41} at the dotted line as in Fig. 4. In this way, we derived an equivalent function, in the meanwhile, the order of M_0 and M_{23} are swapped.

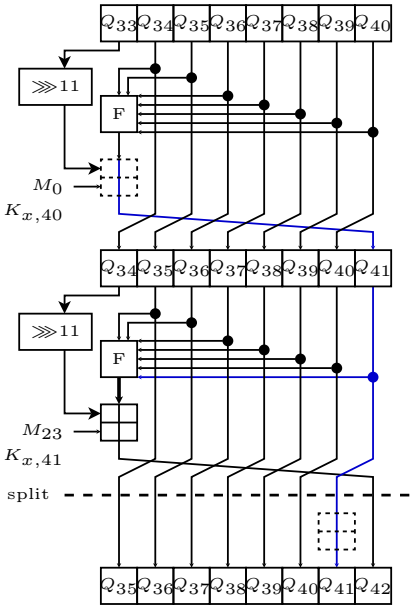

Fig. 4. 2-step initial structure for `HAVAL-3`

Indirect Partial Matching. As demonstrated in Fig. 5, given p_{92} and p_3, we are to match 3 words. Note $Q_{93} = (Q_{85} \ggg 11) + (F_{92}(Q_{86}, Q_{87}, Q_{88}, Q_{89}, Q_{90}, Q_{91}, Q_{92}) \ggg 7) + M_{23} + K_{x,92} = f_q(p_{92}) + M_{23}$, for some f_q. We denote the target T as $T_0||T_1||\cdots||T_7$. To match on T_4, we are to find $Q_{-3} + Q_{93} = f_q(p_{92}) + M_{23} + Q_{-3} = T_4$. One can compute $f_q(p_{92})$ dependent on W_q and $M_{23} + Q_{-3}$ dependent on W_p, independently.

Similarly, $Q_{-5} = (f_p(p_3) - M_2) \lll 11 = (f_p(p_3) \lll 11) - (M_2 \lll 11) - c_0 + (c_1 \lll 11)$ [6, Theorem 4.13], for some carries $c_0, c_1 \in \{0, 1\}$. We can match T_2 indirectly, and we deal with the carries by exhaustively searching all 4 choices.

We directly match T_3, and hence we matched 3 words in total.

Attack Complexities. The attack then follows exactly the Meet-in-the-Middle preimage attack framework explained in Section 3. In pseudo-preimage attack, we utilize the 2^{96} candidates of W_p and W_q independently, and after matching 96 bits, we are left with $2^{96+96-96} = 2^{96}$ candidates for further match. Hence, with computation of 2^{96} we checked 2^{192} pairs, and hence we need to repeat the process $2^{256-192} = 2^{64}$ times to expect one full match. Overall, it costs $2^{96+64} = 2^{160}$ time to find one pseudo-preimage, and the memory complexity is 2^{96}. By repeating the pseudo-preimage attack procedure, one finds 2^{48} pseudo-preimages, then it costs $2^{256-48} = 2^{208}$ to find a message block linking to one of the pseudo-preimages. Hence, the overall attack complexities are 2^{209} for time, and 2^{96} for memory.

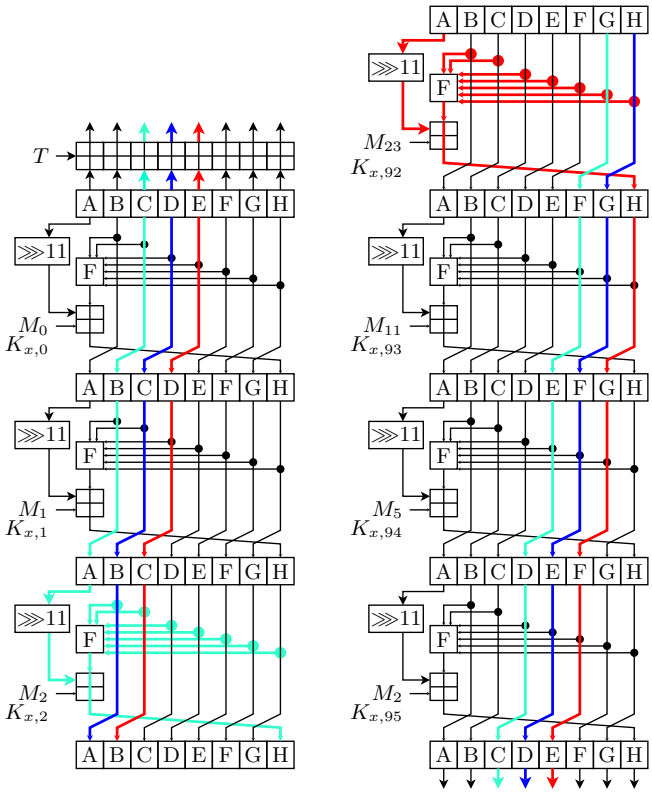

Fig. 5. 7-step indirect partial matching for `HAVAL-3`

4.3 Application to Truncated Variants

There are smaller variant of `HAVAL-3` with digest sizes L with $L \in \{224, 192, 160, 128\}$. Previous attacks [28, 25] apply to 256-bit variant of 3−, 4−, and 5−pass `HAVAL`, as well as preimages for 224-bit variant of `HAVAL-3`. Our improvements on the attack applies to the variants with $L = 224, 192$, which includes complexity improvements over 224-bit variant, as well as the first preimage attack against the 192-bit variant.

Attack on 224-bit Output. Denote H_n as $D_7||D_6||\cdots||D_0$, and further D_7 is divided as follows.

$$D_7 = X^{[5]}_{7,6}||X^{[5]}_{7,5}||X^{[4]}_{7,4}||X^{[5]}_{7,3}||X^{[4]}_{7,2}||X^{[5]}_{7,1}||X^{[4]}_{7,0}\,,$$

where $X^{[a]}$ denotes word X of a bits. The final digest is $D_6 + X^{[4]}_{7,0}||D_5 + X^{[5]}_{7,1}||\cdots||D_0 + X^{[5]}_{7,6}$.

As done in [25], denote $X^{U[a]}$ as the upper a bits of word X. Since $X^{[5]}_{7,5}$ has only 5 bits, $(D_5 + X^5_7)^{U[27]} = D_5^{U[27]} + C$ with $C \in \{0, 1\}$. Hence, we still match the upper $27, 28, 27$ bits of D_5, D_4 and D_3, respectively with 3 bit carries overhead. In other words, we lost $5 + 4 + 5 = 14$ matching bits, which results in more neutral bits than matching bits. To make a re-balance between the number of neutral bits and the matching bits, we set $l_p = l_q = m = 89$ by fixing 7 least significant bits of M_{23} of W_q and 7 bits of M_2 of W_p. Note that with the knowledge of the 7 LSBs of M_{23}, we are able to compute the 7 LSBs of T_5, hence increasing the number of matched bits by 7 to $96 - 14 + 7 = 89$ bits.

The pseudo-preimage attack works the same as before, and it costs $2^{224-89} = 2^{135}$ time and 2^{89} memory. Hence a full preimage costs $2^{(256+135)/2+1} = 2^{196.5}$ time and memory requirement remains the same.

Attack on 192-bit Output. In case of $L = 192$, D_7 and D_6 are divided as follows.

$$D_7 = X_{7,5}^{[6]}||X_{7,4}^{[5]}||X_{7,3}^{[5]}||X_{7,2}^{[6]}||X_{7,1}^{[5]}||X_{7,0}^{[5]}$$
$$D_6 = X_{6,5}^{[6]}||X_{6,4}^{[5]}||X_{6,3}^{[5]}||X_{6,2}^{[6]}||X_{6,1}^{[5]}||X_{6,0}^{[5]}$$

The digest is given by $D_5 + (X_{7,5}^{[6]}||X_{6,4}^{[5]})||D_4 + (X_{7,4}^{[5]}||X_{6,3}^{[5]})||D_3 + (X_{7,3}^{[5]}||X_{6,2}^{[6]})||\cdots||D_0 + (X_{7,0}^{[5]}||X_{6,5}^{[6]})$.

The attack here is similar to that for the 224-bit variant, but here we lost $(6+5)+(5+5)+(5+6) = 32$ matching bits. Hence, we fix 16 bits of M_2 and M_{23}, which re-balances the neutral and matching bits to 80. This results in a pseudo-preimage attack in $2^{192-80} = 2^{112}$ time and 2^{80} memory, then a full preimage in $2^{(256+112)/2+1} = 2^{185}$.

5 Conclusion and Open Problems

In this paper, we presented an improved meet-in-the-middle preimage attack against `HAVAL-3` from the previous best time complexity 2^{225} to 2^{209} for attacking `HAVAL-3` based cryptographic schemes. Similar improvements have been done for truncated variants of output sizes 192 and 224 as well, with time complexity of $2^{196.5}$ and 2^{185}, respectively. Notably, besides faster than the existing preimage attack on `HAVAL-3` around 65.5 thousands time, this is the first preimage attack on the 192-bit variant of `HAVAL-3`.

We tried to apply these techniques to `HAVAL-4`, however the current best results in [28] seems difficult to be improved.

It is noted that the key to improve the time complexity of such MITM preimage attacks are to find larger neutral words. While partial matching works for certain steps, it is the main obstacle to extend the attack further. The key improvement in the paper is due to indirect partial matching which extends the partial matching for 2 steps.

Acknowledgements

We would like to thank the anonymous reviewers of *Information Processing Letters* for the helpful comments. Part of the work was done while Jian Guo was visiting Tsinghua University, China. Wun-She Yap would like to acknowledge UTAR for financially funding his research through the UTAR Research Fund number UTARRF 6200/Y33.

References

1. Kazumaro Aoki, Jian Guo, Krysitan Matusiewicz, Yu Sasaki, and Lei Wang. Preimages for Step-Reduced SHA-2. In Mitsuru Matsui, editor, *ASIACRYPT*, volume 5912 of *LNCS*, pages 578–597. Springer, 2009.
2. Kazumaro Aoki and Yu Sasaki. Meet-in-the-Middle Preimage Attacks Against Reduced SHA-0 and SHA-1. In Halevi [16], pages 70–89.
3. Jean-Philippe Aumasson, Willi Meier, and Florian Mendel. Preimage Attacks on 3-Pass HAVAL and Step-Reduced MD5. In Roberto Avanzi, Liam Keliher, and Francesco Sica, editors, *Selected Areas in Cryptography*, volume 5381 of *LNCS*, pages 120–135. Springer, 2008.
4. Andrey Bogdanov and Christian Rechberger. A 3-Subset Meet-in-the-Middle Attack: Cryptanalysis of the Lightweight Block Cipher KTANTAN. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, *Selected Areas in Cryptography*, volume 6544 of *LNCS*, pages 229–240. Springer, 2010.
5. Christophe De Cannière and Christian Rechberger. Preimages for Reduced SHA-0 and SHA-1. In David Wagner, editor, *CRYPTO*, volume 5157 of *LNCS*, pages 179–202. Springer, 2008.
6. Magnus Daum. *Cryptanalysis of Hash Functions of the MD4-Family.* PhD thesis, Ruhr-Universität Bochum, May 2005. `http://www.cits.rub.de/imperia/md/content/magnus/dissmd4.pdf`.
7. Hüseyin Demirci and Ali Aydin Selçuk. A Meet-in-the-Middle Attack on 8-Round AES. In Nyberg [24], pages 116–126.

8. Hüseyin Demirci, Ali Aydin Selçuk, and Erkan Türe. A New Meet-in-the-Middle Attack on the IDEA Block Cipher. In Mitsuru Matsui and Robert J. Zuccherato, editors, *Selected Areas in Cryptography*, volume 3006 of *LNCS*, pages 117–129. Springer, 2003.
9. Whitfield Diffie and Martin. E. Hellman. Exhaustive Cryptanalysis of the NBS Data Encryption Standard. *Computer*, 10:74–84, June 1977.
10. Orr Dunkelman, editor. *Fast Software Encryption, 16th International Workshop, FSE 2009, Leuven, Belgium, February 22-25, 2009, Revised Selected Papers*, volume 5665 of *LNCS*. Springer, 2009.
11. Orr Dunkelman, Gautham Sekar, and Bart Preneel. Improved Meet-in-the-Middle Attacks on Reduced-Round DES. In K. Srinathan, C. Pandu Rangan, and Moti Yung, editors, *INDOCRYPT*, volume 4859 of *LNCS*, pages 86–100. Springer, 2007.
12. Jian Guo. *ANALYSIS OF CRYPTOGRAPHIC HASH FUNCTIONS*. PhD thesis, Nanyang Technological University, Singapore, 2011.
13. Jian Guo, San Ling, Christian Rechberger, and Huaxiong Wang. Advanced Meet-in-the-Middle Preimage Attacks: First Results on Full Tiger, and Improved Results on MD4 and SHA-2. In Masayuki Abe, editor, *ASIACRYPT*, volume 6477 of *LNCS*, pages 56–75. Springer, 2010.
14. Jian Guo, Yu Sasaki, Lei Wang, Meiqin Wang, and Long Wen. Equivalent Key Recovery Attacks against HMAC and NMAC with Whirlpool Reduced to 7 Rounds. In Calos Cid and Christian Rechberger, editors, *Fast Software Encryption*, London, UK, 2014. Springer. To appear.
15. Jian Guo, Yu Sasaki, Lei Wang, and Shuang Wu. Cryptanalysis of HMAC/NMAC-Whirlpool. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT (2)*, volume 8270 of *LNCS*, pages 21–40. Springer, 2013.
16. Shai Halevi, editor. *Advances in Cryptology - CRYPTO 2009: 29th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 2009, Proceedings*, volume 5677 of *LNCS*. Springer, 2009.
17. Deukjo Hong, Bonwook Koo, and Yu Sasaki. Improved Preimage Attack for 68-Step HAS-160. In Donghoon Lee and Seokhie Hong, editors, *ICISC*, volume 5984 of *LNCS*, pages 332–348. Springer, 2009.
18. Antoine Joux, editor. *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, volume 5479 of *LNCS*. Springer, 2009.
19. Dmitry Khovratovich, Ivica Nikolic, and Ralf-Philipp Weinmann. Meet-in-the-Middle Attacks on SHA-3 Candidates. In Dunkelman [10], pages 228–245.
20. Aggelos Kiayias, editor. *Topics in Cryptology - CT-RSA 2011 - The Cryptographers' Track atthe RSA Conference 2011, San Francisco, CA, USA, February 14-18,2011. Proceedings*, volume 6558 of *LNCS*. Springer, 2011.
21. Simon Knellwolf and Dmitry Khovratovich. New Preimage Attacks against Reduced SHA-1. In Reihaneh Safavi-Naini and Ran Canetti, editors, *CRYPTO*, volume 7417 of *LNCS*, pages 367–383. Springer, 2012.
22. Xuejia Lai and James L. Massey. Hash Function Based on Block Ciphers. In Rainer A. Rueppel, editor, *EUROCRYPT*, volume 658 of *LNCS*, pages 55–70. Springer, 1992.
23. Gaëtan Leurent. MD4 is Not One-Way. In Nyberg [24], pages 412–428.
24. Kaisa Nyberg, editor. *Fast Software Encryption, 15th International Workshop, FSE 2008, Lausanne, Switzerland, February 10-13, 2008, Revised Selected Papers*, volume 5086 of *LNCS*. Springer, 2008.
25. Yu Sasaki. Meet-in-the-Middle Attacks Using Output Truncation in 3-Pass HAVAL. In Pierangela Samarati, Moti Yung, Fabio Martinelli, and Claudio Agostino Ardagna, editors, *ISC*, volume 5735 of *LNCS*, pages 79–94. Springer, 2009.
26. Yu Sasaki. Meet-in-the-Middle Preimage Attacks on AES Hashing Modes and an Application to Whirlpool. In Antoine Joux, editor, *FSE*, volume 6733 of *LNCS*, pages 378–396. Springer, 2011.
27. Yu Sasaki and Kazumaro Aoki. A Preimage Attack for 52-Step HAS-160. In Pil Joong Lee and Jung Hee Cheon, editors, *ICISC*, volume 5461 of *LNCS*, pages 302–317. Springer, 2008.
28. Yu Sasaki and Kazumaro Aoki. Preimage Attacks on 3, 4, and 5-Pass HAVAL. In Josef Pieprzyk, editor, *ASIACRYPT*, volume 5350 of *LNCS*, pages 253–271. Springer, 2008.
29. Yu Sasaki and Kazumaro Aoki. Preimage Attacks on MD, HAVAL, SHA, and Others. CRYPTO 2008 Rump Session, 2008. `http://rump2008.cr.yp.to/efa237568f229268803b82ed02e217ca.pdf`.
30. Yu Sasaki and Kazumaro Aoki. Finding Preimages in Full MD5 Faster Than Exhaustive Search. In Joux [18], pages 134–152.
31. Yu Sasaki and Kazumaro Aoki. Meet-in-the-Middle Preimage Attacks on Double-Branch Hash Functions: Application to RIPEMD and Others. In Colin Boyd and Juan Manuel González Nieto, editors, *ACISP*, volume 5594 of *LNCS*, pages 214–231. Springer, 2009.
32. Yu Sasaki, Wataru Komatsubara, Yasuhide Sakai, Lei Wang, Mitsugu Iwamoto, Kazuo Sakiyama, and Kazuo Ohta. Meet-in-the-Middle Preimage Attacks Revisited - New Results on MD5 and HAVAL. In Pierangela Samarati, editor, *SECRYPT*, pages 111–122. SciTePress, 2013.

33. Gautham Sekar, Nicky Mouha, Vesselin Velichkov, and Bart Preneel. Meet-in-the-Middle Attacks on Reduced-Round XTEA. In Kiayias [20], pages 250–267.
34. Lei Wang, Yu Sasaki, Wataru Komatsubara, Kazuo Ohta, and Kazuo Sakiyama. (Second) Preimage Attacks on Step-Reduced RIPEMD/RIPEMD-128 with a New Local-Collision Approach. In Kiayias [20], pages 197–212.
35. Xiaoyun Wang, Wei Wang, Keting Jia, and Meiqin Wang. New Distinguishing Attack on MAC Using Secret-Prefix Method. In Dunkelman [10], pages 363–374.
36. Xiaoyun Wang, Hongbo Yu, Wei Wang, Haina Zhang, and Tao Zhan. Cryptanalysis on HMAC/NMAC-MD5 and MD5-MAC. In Joux [18], pages 121–133.
37. Lei Wei, Christian Rechberger, Jian Guo, Hongjun Wu, Huaxiong Wang, and San Ling. Improved Meet-in-the-Middle Cryptanalysis of KTANTAN (Poster). In Udaya Parampalli and Philip Hawkes, editors, *ACISP*, volume 6812 of *LNCS*, pages 433–438. Springer, 2011. full version available: `http://eprint.iacr.org/2011/201`.
38. Zheng Yuan, Wei Wang, Keting Jia, Guangwu Xu, and Xiaoyun Wang. New Birthday Attacks on Some MACs Based on Block Ciphers. In Halevi [16], pages 209–230.
39. Yuliang Zheng, Josef Pieprzyk, and Jennifer Seberry. HAVAL - A One-Way Hashing Algorithm with Variable Length of Output. In Jennifer Seberry and Yuliang Zheng, editors, *AUSCRYPT*, volume 718 of *LNCS*, pages 83–104. Springer, 1993.