From: hash-forum@nist.gov on behalf of Danilo Gligoroski [danilo.gligoroski@gmail.com]
Sent: Monday, November 02, 2009 8:42 AM
To: Multiple recipients of list
Subject: OFFICIAL COMMENT:Blue Midnight Wish (Round 2)

Subject: Clarification on the rotation constant for the variable M_15

In the tweaked Blue Midnight Wish, we have introduced left rotations for j+1 positions on all message variables M_j, j=0, 1, ..., 15.

Both in the reference and optimized C code those rotations are correctly encoded. Additionally, in the written documentation on the page 30 we are explicitly saying that M_0 is rotated to the left by 1 position, M_1 by 2 positions, .., M_15 is rotated to the left by 16 positions.

However, in the written specification in the Table 1.3 we have given a short and imprecise expression for AddElement(j) and an equally imprecise explanation saying: "Note that for the function AddElement(j) index expressions involving the variable j for left rotations, M and H are computed modulo 16."

That explanation as such may be misleading and the consequence will be that one of the 16 message variables (the variable M_15) will be rotated to the left for 15 + 1 mod 16 = 0 positions (i.e. will not be rotated at all).

That is not what we write on page 30 of the written documentation and is not what is encoded in the reference and optimized C code.

So, the more precise description of the expression for AddElement(j) that will correspond with the other part of the documentation as well as with the submitted reference and optimized C code, in Table 1.3 should be the following:

AddElement(j) = (ROTL((j mod 16) + 1) (M_{j}) +
ROTL((j+3 mod 16) + 1)(M_{j+3}) - ROTL((j+10 mod 16) + 1)(M_{j+10}) +
K_{j+16}) xor H_{j+7}

where index expressions involving j in variables of M and H are computed modulo 16.

We would like to thank Daniel Otte and especially Stefan Tillich for spotting this ambiguity in the Table 1.3 of the written documentation.

Best Regards,
The Blue Midnight Wish Team

From: Danilo Gligoroski [danilo.gligoroski@gmail.com]
Sent: Tuesday, January 12, 2010 12:02 PM
To: hash-function@nist.gov
Cc: hash-forum@nist.gov
Subject: OFFICIAL COMMENT:Blue Midnight Wish (Round 2)

Hi,

We report about new optimized C versions (and one SSE2 version) of Blue Midnight
Wish hash function. They can be downloaded from:
http://people.item.ntnu.no/~danilog/Hash/BMW-SecondRound/BMW-2009-eBASH.tar.gz

The tarball is prepared for eBASH SUPERCOP, but you can also test the code independently.

With Intel C++ v11.1.46 for Windows, on the reference platform
(Intel Core 2 Duo, 2.4GHz, Windows Vista Ultimate 64-bit edition)
the new speeds are the following:

32-bit environment
BlueMidnightWish performance in Cycles/Byte with different message lengths in
BYTES

	1	8	64	576	1024	1536	4096	100000
BMW224/256	1129	142.63	25.33	9.42	8.52	8.14	7.69	7.45
BMW384/512	1321	165.13	22.33	6.29	5.73	5.28	4.73	4.40

64-bit environment
BlueMidnightWish performance in Cycles/Byte with different message lengths in
BYTES

	1	8	64	576	1024	1536	4096	100000
BMW224/256	1081	135.13	24.77	9.02	8.18	7.76	7.29	7.02
BMW384/512	1105	138.13	17.27	5.09	4.58	4.22	4.05	3.48

On behalf of the Blue Midnight Wish team,
Danilo Gligoroski

From: Danilo Gligoroski [danilo.gligoroski@gmail.com]
Sent: Tuesday, May 18, 2010 9:15 AM
To: ha sh-function@nist.gov
Cc: h ash-forum@nist.gov
Subject: OFFICIAL COMMENT:Blue Midnight Wish (Round 2)

Hi,

We report about new optimized SSSE3 implementations of Blue Midnight Wish - 256 hash function.

They can be downloaded from:
http://people.item.ntnu.no/~danilog/Hash/BMW-SecondRound/bmw256ssse3.tar.gz

The tarball is prepared for eBASH SUPERCOP, and we expect soon to be included in the new supercop version, but you can also test the code independently.

On the reference processor platform (Intel Core 2 Duo) the new speeds are the following:

***Core 2 Duo 65nm, performance in Cycles/Byte with different message lengths in BYTES:
32-bit mode:

	1	8	64	576	1536	4096	100000
MD Size: 256	1131.00	142.62	25.80	9.46	8.15	7.68	7.41

64-bit mode:

	1	8	64	576	1536	4096	100000
MD Size: 256	1001.00	125.12	22.52	8.11	6.99	6.58	6.33

*** Core 2 Duo 45nm, performance in Cycles/Byte with different message lengths in BYTES::
32-bit mode:

	1	8	64	576	1536	4096	100000
MD Size: 256	1055.00	129.75	23.66	8.84	7.68	7.20	6.96

64-bit mode:

	1	8	64	576	1536	4096	100000
MD Size: 256	927.00	111.62	20.86	7.44	6.45	6.07	5.73

The speed gain that Blue Midnight Wish is receiving from new technologies and new realizations of the same (similar) processor architectures is due to the improved internal parallelism in the new CPU editions and the inherent parallelism of the Blue Midnight Wish design.

We expect this technological trend of introducing more internal

parallelism in new editions of CPUs to continue (both in 32-bit and 64-bit world of processors and both for desktop and embedded processors), directly benefiting to even better performance of the Blue Midnight Wish on those processors.

On behalf of the Blue Midnight Wish team,
Danilo Gligoroski

From: hash-forum@nist.gov on behalf of Danilo Gligoroski [danilo.gligoroski@gmail.com]
Sent: Friday, August 27, 2010 12:50 PM
To: Multiple recipients of list
Subject: OFFICIAL COMMENT:Blue Midnight Wish (Round 2)
Attachments: FrameworkHowToEvaluateSecurityInBMW27-08-2010.pdf

Hi,

This note is in a direct compliance with the discussions that took over at the last SHA-3 conference in Santa Barbara on 23-24 August 2010, that there should be better classification on the growing number of attacks on all hash functions that do not follow the well established in cryptology definition for a distinguisher of a pseudo-random function (see for example Bellare and Rogaway "Introduction to Modern Cryptography", Ch. 3, Sec. 3.4, or Goldreich "Foundations of Cryptography - A Primer", Ch. 3, Sec. 3.3) and how these attacks can be observed from a global perspective of the security margins in the attacked functions.

Thus, as a response to the growing cryptanalytic work on Blue Midnight Wish hash function we define a framework that easily captures and classifies all those and future attacks both on the compression function and on the whole hash function.

On behalf of the Blue Midnight Wish team,
Danilo Gligoroski

A framework for Measuring and Evaluating the Progress of the Cryptanalysis of the Hash Function Blue Midnight Wish

The Blue Midnight Wish team

August 27, 2010

Attacker	Hash size	Type of attack	Compression function		Whole function	
			Attacked variables (rounds)	Complexity	Attacked variables (rounds)	Complexity
Aumasson[1]	All	pseudo-distinguisher	1 out of 16	2^{19}	0 out of 32	N/A
Nikolic et.al.[2]	512	pseudo-distinguisher on modified function	1 out of 16	$2^{-278.2}$	0 out of 32	N/A
Guo &Thomsen[3]	All	pseudo-distinguisher	1 out of 16	2^{1}	0 out of 32	N/A
Laurent[4]	256	pseudo-collision	3 out of 16	2^{32}	0 out of 32	N/A

Table 1: Evaluating the progress of the cryptanalysis of Blue Midnight Wish

This note is in a direct compliance with the discussions that took over at the last SHA-3 conference in Santa Barbara on 23-24 August 2010, that there should be better classification on the growing number of attacks on all hash functions that do not follow the well established in cryptology definition for a distinguisher of a pseudo-random function (see for example Bellare and Rogaway [5], Ch. 3, Sec. 3.4, or Goldreich [6], Ch. 3, Sec. 3.3) and how these attacks can be observed from a global perspective of the security margins in the attacked functions. Thus, as a response to the growing cryptanalytic work on Blue Midnight Wish hash function we define a framework for classification of all those and future attacks both on the compression function and on the whole hash function.

Blue Midnight Wish [7] hash function has no explicit rounds in its design. However, the compression function is producing 16 variables of the double-pipe chain with increased complexity beginning from the variable H_0 that has the lowest computational complexity, up

to the variable H_{15} that has the highest computational complexity. That is a very strong analogy with the designs that have rounds in their design and where the complexity of computed components in those designs is increasing in every round.

By setting the output variables $H_i,\ i = 0, \ldots, 15$ of the compression function of Blue Midnight Wish to denote an equivalent notion to the "rounds" in classical designs, we will enable independent cryptographers to evaluate and measure the success of their attack and the strength of the function in accordance of the cryptanalytic progress. Since the whole hash function has additional blank final invocation of the compression function this implies that in this framework the number of rounds that will correspond for the whole hash function is at least by 16 more than in the compression function.

Thus from cryptanalytic point of view we can talk about two values that are determining the security margins in Blue Midnight Wish:

1. Number of variables (rounds) with ever-increasing computational complexity as security margin on the compression function.

The security margin in the compression function has a value 16 as an analogy with the designs that have 16 rounds in their compression functions.

2. Number of variables (rounds) with ever-increasing computational complexity as security margins for the whole hash function.

The security margin for the whole hash function has a value 32 as an analogy with the designs that have 32 rounds in their compression functions. The rationale for setting the security margin as 32 is that for the whole hash function, the minimal number of produced variables out of the compression function calls in Blue Midnight Wish is 32 (one call to the compression function and one finalization call). Thus any successful attack on the compression function, in order to be transferred to the whole function will have at least 32 produced variables out of the compression function with each of them produced in a series of ever increasing computational complexity.

All independent cryptanalysis for Blue Midnight Wish that has happened so far (and the new one that has been recently announced in the Rump session of CRYPTO 2010) are naturally fitting in this framework for measuring and evaluating the progress of the cryptanalysis of the hash function. They are presented in Table 1.

Additionally, so far all attacks have gone in the direction of taking the control over both H and M variable. According to the taxonomy of the attacks on hash functions developed in the PhD thesis of Preneel (see [8] Ch. 2.5), all these attacks are "pseudo-attacks"[1]. This fact is automatically making these attack techniques non-applicable and non-effective against the whole function because the final invocation of the compression function is such that it excludes attack techniques that assume control over both H and M.

The actual situation that all attacks so far are pseudo-attacks is a direct confirmation of the soundness of the design rationale to incorporate big number of entangled bijections that will force the attacks to be only pseudo-attacks. Additionally, from the described framework and Table 1 it is clear that Blue Midnight Wish is not just the best performer, but also a candidate with the biggest security margin among Second Round SHA-3 candidates.

[1] According to Miriam-Webster online dictionary (`http://www.merriam-webster.com/dictionary/pseudo` - accessed Aug 27 2010) the meaning of the word "pseudo" is: being apparently rather than actually as stated.

References

[1] J.-P. Aumasson: "Practical distinguisher for the compression function of Blue Midnight Wish", Feburary 2010. Available: http://131002.net/data/papers/Aum10.pdf (2009/08/27).

[2] I. Nikolić, J. Pieprzyk, P. Sokołowski, and R. Steinfeld: "Rotational Cryptanalysis of (Modified) Versions of BMW and SIMD", March 2010. Available: https://cryptolux.org/mediawiki/uploads/0/07/Rotational_distinguishers_(Nikolic,_Pieprzyk,_Sokolowski,_Steinfeld).pdf (2010/08/27).

[3] J. Guo and S. S. Thomsen: "Distinguishers for the Compression Function of Blue Midnight Wish with Probability 1", March 2010. Available: http://www2.mat.dtu.dk/people/S.Thomsen/bmw/bmw-distinguishers.pdf (2010/08/27).

[4] G. Laurent, "Self-Defence Against Fresh Fruit", CRYPTO 2010 Rump Session. Available: http://rump2010.cr.yp.to/c659ebaf681758e01ccf824fd58f3c42.pdf (2010/08/27). The details of the attack are not yet fully disclosed but according to the author, more details will be posted soon.

[5] Mihir Bellare and Phillip Rogaway, "Introduction to Modern Cryptography," Department of Computer Science and Engineering, University of California, September 2005

[6] Oded Goldreich, "Foundations of cryptography: a primer," New Publishers Inc., USA, 2005

[7] D. Gligoroski, V. Klima, S. J. Knapskog, M. El-Hadedy, J. Amundsen, and S. F. Mjølsnes: "Cryptographic hash function BLUE MIDNIGHT WISH. Submission to NIST (Round 2)". Available: http://people.item.ntnu.no/~danilog/Hash/BMW-SecondRound/Supporting_Documentation/BlueMidnightWishDocumentation.pdf (2010/08/27), September 2009.

[8] B. Preneel, "Analysis and Design of Cryptographic Hash Functions," PhD thesis, Katholieke Universiteit Leuven, January 1993.

From: hash-forum@nist.gov on behalf of Svein Johan Knapskog [knapskog@q2s.ntnu.no]
Sent: Tuesday, October 19, 2010 3:44 AM
To: Multiple recipients of list
Subject: OFFICIAL COMMENT: New implementations of Blue Midnight Wish in hardware

Dear all,

We would like to inform you that we have implemented the core functionality of the Blue Midnight Wish hash function with finalization but without the padding stage on Xilinx Virtex-5 FPGA. The implementations require 51 slices for BMW-256 and 105 slices for BMW-512. Both BMW versions require two blocks of memory: one memory block to store the intermediate values and hash constants and the other to store the instruction controls. The proposed implementation achieves a throughput of 68.71 Mpbs for BMW-256 and 112.18 Mpbs for BMW-512.

The complete report in PDF and complete implementation packages and workbenches can be downloaded from:
http://www.q2s.ntnu.no/sha3_nist_competition/start

On behalf of the Blue Midnight Wish team,
Svein Johan Knapskog

From:	hash-forum@nist.gov on behalf of Danilo Gligoroski [danilo.gligoroski@gmail.com]
Sent:	Wednesday, November 17, 2010 10:02 AM
To:	Multiple recipients of list
Subject:	OFFICIAL COMMENT:Blue Midnight Wish (Round 2)
Attachments:	CommentNov2010.pdf

Hi,

We comment on latest work by Leurent and Thomsen: ``New Distinguisher on BMW compression function''.

1. We think that Laurent-Thomsen work is a great result in the study of the compression function of Blue Midnight Wish.

2. However, we also think that the correct title of their work that is compliant with the widely accepted cryptographic terminology should be ``Practical Partial-Pseudo-Collisions on the Compression Function of BMW''.

3. This attack fits perfectly in the established framework for analyzing Blue Midnight Wish which we have posted on the SHA-3 forum list on 27/08/2010.

4. Further on, we disagree with Laurent and Thomsen allegation that their work in finding partial pseudo-collisions in Blue Midnight Wish is analogous with the work of den Boer and Bosselaers on MD5 because den Boer and Bosselaers found *COMPLETE* pseudo-collisions on the narrow-pipe compression function of MD5, while Leurent and Thomsen found a *PARTIAL* pseudo-collision on the double-pipe compression function of Blue Midnight Wish with 212 bits in the output left out of reach of their controlling technique and because of two essentially different design principles in Blue Midnight Wish that are not present in MD5:
a) Blue Midnight Wish is a double-pipe hash design and
b) Blue Midnight Wish is similar with the highly respected cryptographic primitive HMAC.
These two design principles renders out all pseudo-attacks (as the one of Leurent and Thomsen) on Blue Midnight Wish as attacks without a potential and a perspective neither to harm nor to break the algorithm.

Please see the attached document with an extended explanation.

Best regards,
The Blue Midnight Wish team

A Comment on Leurent and Thomsen work - New Distinguisher on BMW compression function

The Blue Midnight Wish team

November 16, 2010

Abstract

We give a comment on latest work by Leurent and Thomsen: "New Distinguisher on BMW compression function" [1]. We think that Laurent-Thomsen work is a great result in the study of the compression function of Blue Midnight Wish. However, we also think that the correct title of their work that is compliant with the widely accepted cryptographic terminology should be "Practical Partial-Pseudo-Collisions on the Compression Function of Blue Midnight Wish". This attack fits perfectly in the established framework for analyzing Blue Midnight Wish which we have posted on the SHA-3 forum list on 27/08/2010. Further on, we disagree with Laurent and Thomsen allegation that their work in finding partial pseudo-collisions in Blue Midnight Wish is analogous with the work of den Boer and Bosselaers on MD5 in [2] because den Boer and Bosselaers found **complete** pseudo-collisions on the compression function of MD5, while Leurent and Thomsen found a **partial** pseudo-collision in the compression function of Blue Midnight Wish with 212 bits in the output left out of reach of their controlling technique and because of two essentially different design principles in Blue Midnight Wish that are not present in MD5: Blue Midnight Wish is a double-pipe hash design and is similar with the highly respected cryptographic primitive HMAC. These two design principles renders out all pseudo-attacks (as the one of Leurent and Thomsen) on Blue Midnight Wish as attacks without a potential and a perspective neither to harm nor to break the algorithm.

1 Credits to the work of Leurent and Thomsen

We commend Leurent and Thomsen for their research efforts in connection with their cryptanalysis of the Blue Midnight Wish algorithm. We think that the Leurent-Thomsen paper [1] presents a significant result in the study of the compression function of the Blue Midnight Wish.

We appreciate this new advantage in the non-trivial study of differential properties of the compression function of Blue Midnight Wish because this is important and very difficult to achieve. Despite of incorrect conclusions in the paper, it is a great contribution to the study of the differential properties of Blue Midnight Wish compression function. The paper shows new and very nice way how to manipulate differentials inside the two thirds of the compression function. It also shows how difficult it is to bypass entangling bijections, used in the compression function, what is one of the basic building principle of Blue Midnight Wish. We thank Leurent and Thomsen for this great work.

2 Critique of some of the claims and alleged implications of their work to the security of the Blue Midnight Wish hash function

We organize our critical remarks in 5 points.

1. The latest attack on the compression function of Blue Midnight Wish hash function by Leurent and Thomsen is again a pseudo-attack since they control both the message and the chaining value. Thus, the correct title that is compliant with the widely accepted cryptographic terminology should be "*Practical Partial-Pseudo-Collisions on the Compression Function of* Blue Midnight Wish".

2. This attack fits perfectly in the established framework for analyzing Blue Midnight Wish which we have posted on the SHA-3 forum list on 27/08/2010 [3]. The partial pseudo-collision that they find has three fully collided values in the first part of the chaining value. Specifically, their pseudo-attack achieves complete collision on the first 3 variables of the chaining value and partial collision on 7 additional variables, leaving 6 variables of the output beyond collision control. The updated framework which includes the latest pseudo-attack of Leurent and Thomsen is already included in our web page: `http://www.q2s.ntnu.no/sha3_nist_competition/start` and the corresponding pdf: `http://people.item.ntnu.no/~danilog/Hash/BMW-SecondRound/FrameworkHowToEvaluateSecurityInBMW-Nov-2010.pdf` may be downloaded from there.

3. Leurent and Thomsen seem to be trying to increase the value of their analysis by giving their work a perspective and potential impact similar to that of den Boer and Bosselaers' work on their MD5 analysis. From the Leurent and Thomsen paper, we quote: "*To put such attacks into perspective, one might look at the attacks on MD5. The first attack on the compression function was found in 1993 by den Boer and Bosselaers [5], using a very simple differential path. This attack did not threaten the iterated hash function, but the path used in the attack is a core element of the successful attack of Wang et al. in 2005 [10].*"

   There are at least two evident mismatches in using the analogy between the work of den Boer and Bosselaers [2] and the history of the MD5 analysis and the work of Leurent and Thomsen on Blue Midnight Wish:

   a) The collisions for the compression function of MD5 which were found by den Boer and Bosselaers were described by a precise terminology as pseudo-collisions by Robshaw in [4] and by Dobbertin in [5]. Leurent and Thomsen should also strive to use such precise terminology

   b) den Boer and Bosselaers found **COMPLETE** pseudo-collisions on the narrow-pipe compression function of MD5, while Leurent and Thomsen found a **PARTIAL** pseudo-collision in the double-pipe compression function of Blue Midnight Wish, with 212 bits in the output left out of reach of their controlling technique.

   However, since they have put their work into this perspective and are making allegations that their work will decrease the confidence in Blue Midnight Wish as den Boer's and

Bosselaer's work did for MD5, we would also like to put into perspective their attack (and all other pseudo-attacks) on Blue Midnight Wish by recalling the similarity of the finalization of the Blue Midnight Wish algorithm with the HMAC (that fact Leurent and Thomsen are mentioning in the introduction to their work and was first mentioned in the analysis of the SHA-3 candidates done by Andreeva, Mennink and Preneel in [6]). So having a hash function for which similar design principles as for HMAC (one of the most trusted designs in the contemporary cryptology) have been used, clearly increases the confidence in Blue Midnight Wish and renders out all pseudo-attacks on it as attacks without a potential and a perspective neither to harm nor to break the algorithm.

4. We do not see as a truthful and as a big achievement their claim in the conclusion "*We also note that if the compression function is truncated like in the final transformation of BMW, we can still build pairs of message which collide in more than 110 bits with complexity* 2^{32}*. This is the first distinguisher on the truncated compression function of BMW.*" As already noted, it is not a distinguisher but a pseudo-distinguisher. With the same computational effort that they are using (2^{32} calls to the compression function) a generic partial collision search can find a real partial collision on approximately 198 bits on the second-half of the chaining value (by "real" we mean without the need to control every input into the compression function i.e. without going into the direction of a pseudo-attack).

   Moreover, the claim: "*... if the compression function is truncated like in the final transformation of* Blue Midnight Wish*, we can still build pairs of messages which collide in more than 110 bits with complexity* 2^{32}*.*" is not correct. In the part where they say that they are able to build "*pairs of messages*", the precise phrasing would be ..."*pairs of new chaining values and pairs of final constants M ...*", because M is no longer a message block in the final transformation of Blue Midnight Wish.

   Additionally, the presented pseudo-distinguisher requires a huge control of the chaining variable H which in the final transformation is a pre-computed value obtained by digesting the message and in Laurent-Thomsen work there are no indications how their complete control over the chaining variable and the message blocks can be transformed into an attack that controls the whole message. If the conclusion is rewritten correctly, we would see the following statement: "*We also note that if the output of the compression function is truncated to its half, we can still build pairs of constants and pairs of chaining-hash values which collide in more than 110 bits with complexity* 2^{32}*.*" Now this is true, but without any value. As we have already stated, a much better partial-collision result is possible to obtain without employing the pseudo-attack by a simple generic search of partial collisions. So the conclusion in their paper should be corrected in some way (and if we were given their draft work in advance as it is a general ethical attitude in academic Cryptologic research) we would have pointed out these incorrect parts.

5. Let us analyze the implication of finding partial pseudo-collisions on the security of the hash function. Recall that the final step of the hash function Blue Midnight Wish is $C(H_{LAST}, CONST)$, where H_{LAST} is the value of the previous iterative hashing of the padded message m and $C()$ is the compression function. As it was correctly noted

in the paper, it is similar to HMAC construction - H_{LAST} is not a message block now, but a "*pre-hash*" value. Moreover, in the case of Blue Midnight Wish, the length of this pre-hash value is twice as long as in the case of HMAC construction!

Let us suppose the attacker succeeded to find a collision or a near-collision on the whole Blue Midnight Wish hash function. How he/she succeeded to do that? There are only two cases. The first one is that the values $H_{LAST} = H'_{LAST}$ in the last step are the same and are coming from two different digested messages $m \neq m'$. The second one is that the values H_{LAST} and H'_{LAST} in the last step are different.

a) In the first case the attacker found a complete collision (not a pseudo-collision) of the compression function (with double length output). So, in this case, the first necessary condition is that the attacker has to find a **COMPLETE** collision of the double-pipe compression function. Note that it is only necessary, not sufficient condition, because there has to be a way how to obtain this value $H_{LAST} = H'_{LAST}$ for two different messages from the beginning of hashing. The important note is that any near-collision even on all bits but one is not useful. It has to be complete collision on all bits of the double-pipe compression function! Usually, finding near-collision of the compression function is a great result. Here it could be even contra productive. Having very near values H_{LAST} and H'_{LAST} i.e. $Hamming(H_{LAST}, H'_{LAST})$ is low, the final operation $C(H_{LAST}, CONST)$ and $C(H'_{LAST}, CONST)$ will diffuse them into two values having Hamming distance around 256 (512 for BMW512). Bijections used in Blue Midnight Wish behave like MDS codes - small changes in the input guarantee big changes in the output. So the first task is different from the traditional hash constructions: we need **COMPLETE** collision of the compression function, *assuming that ANY NEAR-COLLISION IS VERY NEGATIVE RESULT*! Moreover the first task *works on DOUBLE LENGTH* values compared with the traditional narrow-pipe hash designs.

How to measure the effectiveness of Leurent-Thomsen near-collision of the compression function? Should we continue to extend their near-collisions from 300 to more bits up to 511? We have just offered arguments that from the breaking point of view for the whole hash function it could be even contra productive. But it has a big sense and big importance for understanding the insights of the hash function and to study its properties. So their paper is great, because it shows some properties of the compression function. This paper shows that even with a total control of every input variable in the compression function of Blue Midnight Wish, at best you can get is a partial collision which has not much use for breaking the real hash function, and that speaks very much in favor of the strength of the hash function. Just compare the situation of having a total control over all inputs of the "compression function" of the sponge designs - you need only 2 calls to the inverse of the bijective function and you have a **COMPLETE pseudo-collision** (not that it has anything to do with the general strength of the sponge-based hash designs).

b) The alternative to the first case is the second case consisting of finding two different pre-hash values $H_{LAST} \neq H'_{LAST}$ such that the $chopC(H_{LAST}, CONST)$ and

$chopC(H'_{LAST}, CONST)$ are equal or near. Note again that this is not sufficient condition for a successful attack, because the attacker in this case have to find a way how to obtain these pre-hash values $H_{LAST} \neq H'_{LAST}$ for two different messages from the beginning of hashing. In this second task the attacker has to explore the function $H_{LAST} \mapsto chopC(H_{LAST}, CONST)$. This function is very different from the compression function $(M, H) \mapsto C(M, H)$ since the roles of M and H are now swapped. Moreover, the partial transformations inside the $chopC$ are very different from $C(M, H)$, when one variable is a constant.

And, of course, the function $H_{LAST} \mapsto chopC(H_{LAST}, CONST)$ has half freedom both in input and output variables, so differential strategies and paths will be very different from the first case. Also, when you look at the nice Fig.1 of the Leurent-Thomsen paper, the variable M is now going into H_{LAST}, the variable Q_a is now (due to the constant $CONST$) a **BIJECTIVE image** of H_{LAST}, and Q_b is some kind of one-way function of H_{LAST}. These three variables are inputs to the function f_2. Now, Q_a behaves like MDS code of H_{LAST} - the smaller changes in H_{LAST}, the bigger changes in $Q_a(H_{LAST})$, so in the couple $(H_{LAST}, Q_a(H_{LAST}))$ there is guaranteed some amount of changes in total. The behavior of the special function $H_{LAST} \mapsto f_2(H_{LAST}, Q_a(H_{LAST}), Q_b(H_{LAST}))$ is crucial. This is the second way how to explore collisions of Blue Midnight Wish, which has not been explored so far. We would like to stimulate any research in this direction.

We can conclude this point that whatever the attacker knows and uses, he/she has to complete either scenario a) or scenario b). In the first scenario it is necessary to obtain the complete collision on the double pipe. Obtaining near-collision has no significance for launching an attack, it has a meaning for the study of the compression function. And this is the case of Leurent-Thomsen paper.

References

[1] G. Leurent and S. S. Thomsen, "Practical Partial-Collisions on the Compression Function of BMW", SHA-3 Hash forum list from 12 Nov 2010.

[2] B. den Boer and A. Bosselaers, "Collisions for the compression function of MD5", Advances in Cryptology Eurocrypt 93, LNCS, vol. 773, Springer-Verlag, 1994, pp. 293-304.

[3] Blue Midnight Wish team, "A framework for Measuring and Evaluating the Progress of the Cryptanalysis of the Hash Function Blue Midnight Wish", SHA-3 Hash forum list from 27 Aug 2010.

[4] M. Robshaw, "On pseudo-collisions in MD5", Technical Report TR-102, ver. 1.1., RSA Laboratories, July 1994.

[5] H. Dobbertin, "Cryptanalysis of MD5 compress", presented at the rump session of EurocrCrypt'96.

[6] E. Andreeva and B. Mennink and B. Preneel, "Security Reductions of the Second Round SHA-3 Candidates", Cryptology ePrint Archive, Report 2010/381.