

**Subject:** OFFICIAL COMMENT: LUX

**From:** Watanabe Dai <dai.watanabe.td@hitachi.com>

**Date:** Thu, 09 Apr 2009 11:59:04 +0900

**To:** hash-function@nist.gov

**CC:** hash-forum@nist.gov

Dear all,

I looked at Wu et. al's report [1] on LUX and I believe that their observation can obviously be extended to a collision attack and a second preimage attack which would be labeled orange in the SHA-3 zoo according to their computational complexities.

The following is the brief sketch of the attacks.

== Wu et. al's observation

Let  $H=(h_0, h_1, \dots, h_7)$  be the hash value and  $h_i=(a_i, b_i, c_i, d_i)$  be its byte expression. Then the following relation holds for  $0 < i < 8$ .  
 $0xf7*a_i + 0x4c*b_i + 0xf4*c_i + d_i = 0x4e * S(a_{(i-1)}) \dots \dots (1)$   
 See [1] Section 3.1 for more detail.

== What does it mean?

Eq.(1) means that 4 bytes of  $h_i$  of the output determine 1 byte of  $h_{(i-1)}$ . In other words, LUX-256 has undesirable property that the  $56(=8*7)$  bits of the output are determined by the remaining  $256-56=200$  bits without extra computational cost. It obviously reduces the complexity of the birthday attack and the second preimage attack.

== Collision attack and second preimage attack

Find a partial collision such that  $b_i=b_i', c_i=c_i', d_i=d_i'$  for  $0 \leq i < 8$  and  $a_7=a_7'$ . With help of Eq.(1) and the fact that  $S$  is a permutation, we have  $a_i=a_i'$  for  $0 \leq i < 7$ .

=== Complexity of collision attack

\* Hash function call:  $2^{\{(3*8+1)*8/2\}} = 2^{100}$ .

\* Memory:  $2^{100}$ .

=== Complexity of second preimage attack

\* Hash function call:  $2^{200}$ ,

\* Memory: none.

== LUX-512

LUX-512 has the same weakness as LUX-256 so that the 56(=8\*7) bits of the output are determined by the remaining 512-56=456 bits. In the similar manner to the attacks on LUX-256, the collision attack and the second preimage attack require  $2^{\{456/2\}}=2^{228}$  and  $2^{456}$  complexity respectively.

[1] Shuang Wu, Dengguo Feng, Wenling Wu  
Cryptanalysis of the Hash Function LUX-256  
[http://ehash.iaik.tugraz.at/uploads/3/36/Analysis\\_LUX\\_1.pdf](http://ehash.iaik.tugraz.at/uploads/3/36/Analysis_LUX_1.pdf)

Regards,  
Dai Watanabe

**Subject:** RE: OFFICIAL COMMENT: LUX  
**From:** Niels Ferguson <niels@microsoft.com>  
**Date:** Thu, 9 Apr 2009 11:33:52 -0400  
**To:** Multiple recipients of list <hash-forum@nist.gov>

I have not validated this attack, but if I understand the results correctly, LUX-256 can be seen as a 200-bit hash function with a post-processing function that stretches the 200 bits into a 256-bit result. That means it can't be used in SP 800-90 Hash-DRBG, it generates weak keys if used in a normal hash-based KDF, has reduced security for HMAC-LUX, etc.

A minor point: We can drop the memory requirements for collision finding from  $2^{100}$  ( $2^{228}$  for LUX-512) to a constant size by using Floyd's or Brent's cycle finding algorithm.

Regards,

Niels

---

From: [hash-forum@nist.gov](mailto:hash-forum@nist.gov) [[hash-forum@nist.gov](mailto:hash-forum@nist.gov)] On Behalf Of Watanabe Dai [[dai.watanabe.td@hitachi.com](mailto:dai.watanabe.td@hitachi.com)]  
 Sent: Wednesday, April 08, 2009 8:02 PM  
 To: Multiple recipients of list  
 Subject: OFFICIAL COMMENT: LUX

Dear all,

I looked at Wu et. al's report [1] on LUX and I believe that their observation can obviously be extended to a collision attack and a second preimage attack which would be labeled orange in the SHA-3 zoo according to their computational complexities.

The following is the brief sketch of the attacks.

== Wu et. al's observation

Let  $H=(h_0, h_1, \dots, h_7)$  be the hash value and  $h_i=(a_i, b_i, c_i, d_i)$  be its byte expression. Then the following relation holds for  $0 < i < 8$ .  
 $0xf7*a_i + 0x4c*b_i + 0xf4*c_i + d_i = 0x4e * S(a_{i-1}) \dots (1)$   
 See [1] Section 3.1 for more detail.

== What does it mean?

Eq.(1) means that 4 bytes of  $h_i$  of the output determine 1 byte of  $h_{i-1}$ .

In other words, LUX-256 has undesirable property that the  $56(=8*7)$  bits of the output are determined by the remaining  $256-56=200$  bits without extra computational cost.

It obviously reduces the complexity of the birthday attack and the second preimage attack.

== Collision attack and second preimage attack

Find a partial collision such that  $b_i=b_i', c_i=c_i', d_i=d_i'$  for  $0 \leq i < 8$  and  $a_7=a_7'$ . With help of Eq.(1) and the fact that  $S$  is a permutation, we have  $a_i=a_i'$  for  $0 \leq i < 7$ .

=== Complexity of collision attack

\* Hash function call:  $2^{\{(3*8+1)*8/2\}} = 2^{100}$ .  
 \* Memory:  $2^{100}$ .

=== Complexity of second preimage attack

\* Hash function call:  $2^{200}$ ,

\* Memory: none.

== LUX-512

LUX-512 has the same weakness as LUX-256 so that the 56(=8\*7) bits of the output are determined by the remaining 512-56=456 bits.

In the similar manner to the attacks on LUX-256, the collision

attack and the second preimage attack require  $2^{\{456/2\}}=2^{228}$

and  $2^{456}$  complexity respectively.

[1] Shuang Wu, Dengguo Feng, Wenling Wu

Cryptanalysis of the Hash Function LUX-256

[http://ehash.iaik.tugraz.at/uploads/3/36/Analysis\\_LUX\\_1.pdf](http://ehash.iaik.tugraz.at/uploads/3/36/Analysis_LUX_1.pdf)

Regards,

Dai Watanabe

**Subject:** Re: OFFICIAL COMMENT: LUX

**From:** Nicky Mouha <Nicky.Mouha@esat.kuleuven.be>

**Date:** Thu, 9 Apr 2009 17:46:51 -0400

**To:** Multiple recipients of list <hash-forum@nist.gov>

Hi,

The attack strategy is completely valid, but some hexadecimal values in the formula are incorrect. To demonstrate this, I've uploaded a distinguisher for all digest sizes of LUX at <http://www.nickymouha.be/software-en.html>

All credit for this distinguisher should go to Shuang Wu, Dengguo Feng and Wenling Wu. The only thing I did is correct some calculation errors and trivially extend the results to LUX-384/512.

Kind regards,  
Nicky

Niels Ferguson wrote:

I have not validated this attack, but if I understand the results correctly, LUX-256 can be seen as a 200-bit hash function with a post-processing function that stretches the 200 bits into a 256-bit result. That means it can't be used in SP 800-90 Hash-DRBG, it generates weak keys if used in a normal hash-based KDF, has reduced security for HMAC-LUX, etc.

A minor point: We can drop the memory requirements for collision finding from  $2^{100}$  ( $2^{228}$  for LUX-512) to a constant size by using Floyd's or Brent's cycle finding algorithm.  
Regards,

Niels

---

From: [hash-forum@nist.gov](mailto:hash-forum@nist.gov) [[hash-forum@nist.gov](mailto:hash-forum@nist.gov)] On Behalf Of Watanabe Dai  
[[dai.watanabe.td@hitachi.com](mailto:dai.watanabe.td@hitachi.com)]  
Sent: Wednesday, April 08, 2009 8:02 PM  
To: Multiple recipients of list  
Subject: OFFICIAL COMMENT: LUX

Dear all,

I looked at Wu et. al's report [1] on LUX and I believe that their observation can obviously be extended to a collision attack and a second preimage attack which would be labeled orange in the SHA-3 zoo according to their computational complexities.  
The following is the brief sketch of the attacks.

== Wu et. al's observation

Let  $H=(h_0, h_1, \dots, h_7)$  be the hash value and  $h_i=(a_i, b_i, c_i, d_i)$  be its byte expression.  
Then the following relation holds for  $0 < i < 8$ .  
 $0xf7*a_i + 0x4c*b_i + 0xf4*c_i + d_i = 0x4e * S(a_{i-1}) \dots (1)$   
See [1] Section 3.1 for more detail.

== What does it mean?

Eq.(1) means that 4 bytes of  $h_i$  of the output determine 1 byte of  $h_{i-1}$ .  
In other words, LUX-256 has undesirable property that the  $56(=8*7)$  bits of the output are determined by the remaining  $256-56=200$  bits without extra computational cost.  
It obviously reduces the complexity of the birthday attack

and the second preimage attack.

== Collision attack and second preimage attack

Find a partial collision such that  
 $b_i = b'_i, c_i = c'_i, d_i = d'_i$  for  $0 \leq i < 8$  and  $a_7 = a'_7$ .  
With help of Eq.(1) and the fact that  $S$  is a permutation,  
we have  $a_i = a'_i$  for  $0 \leq i < 7$ .

=== Complexity of collision attack

\* Hash function call:  $2^{\{(3*8+1)*8/2\}} = 2^{100}$ .  
\* Memory:  $2^{100}$ .

=== Complexity of second preimage attack

\* Hash function call:  $2^{200}$ ,  
\* Memory: none.

== LUX-512

LUX-512 has the same weakness as LUX-256 so that the 56(=8\*7) bits  
of the output are determined by the remaining 512-56=456 bits.  
In the similar manner to the attacks on LUX-256, the collision  
attack and the second preimage attack require  $2^{\{456/2\}} = 2^{228}$   
and  $2^{456}$  complexity respectively.

[1] Shuang Wu, Dengguo Feng, Wenling Wu  
Cryptanalysis of the Hash Function LUX-256  
[http://ehash.iaik.tugraz.at/uploads/3/36/Analysis\\_LUX\\_1.pdf](http://ehash.iaik.tugraz.at/uploads/3/36/Analysis_LUX_1.pdf)

Regards,  
Dai Watanabe

**Subject:** Re: OFFICIAL COMMENT: LUX  
**From:** Ivica Nikolic <cube444@gmail.com>  
**Date:** Sat, 11 Apr 2009 05:40:43 -0400  
**To:** Multiple recipients of list <hash-forum@nist.gov>

Dear all,

We knew about these observations since they are a straightforward consequence of the weaknesses in the output filter only. We already decided to drop the filter due to the observations of Wu et al. We find the recent work on the number of blank rounds of LUX done by Schmidt-Nielsen of more importance.

Best regards,

Ivica, Alex, Dmitry

2009/4/9 Watanabe Dai <[dai.watanabe.td@hitachi.com](mailto:dai.watanabe.td@hitachi.com)>

Dear all,

I looked at Wu et. al's report [1] on LUX and I believe that their observation can obviously be extended to a collision attack and a second preimage attack which would be labeled orange in the SHA-3 zoo according to their computational complexities.

The following is the brief sketch of the attacks.

== Wu et. al's observation

Let  $H=(h_0, h_1, \dots, h_7)$  be the hash value and  $h_i=(a_i, b_i, c_i, d_i)$  be its byte expression.

Then the following relation holds for  $0 < i < 8$ .

$$0xf7*a_i + 0x4c*b_i + 0xf4*c_i + d_i = 0x4e * S(a_{i-1}) \dots (1)$$

See [1] Section 3.1 for more detail.

== What does it mean?

Eq.(1) means that 4 bytes of  $h_i$  of the output determine 1 byte of  $h_{i-1}$ .

In other words, LUX-256 has undesirable property that the  $56 (= 8 * 7)$  bits of the output are determined by the remaining  $256 - 56 = 200$  bits without extra computational cost.

It obviously reduces the complexity of the birthday attack and the second preimage attack.

== Collision attack and second preimage attack

Find a partial collision such that

$b_i = b'_i, c_i = c'_i, d_i = d'_i$  for  $0 \leq i < 8$  and  $a_7 = a'_7$ .

With help of Eq.(1) and the fact that  $S$  is a permutation, we have  $a_i = a'_i$  for  $0 \leq i < 7$ .

=== Complexity of collision attack

\* Hash function call:  $2^{\{(3 \cdot 8 + 1) \cdot 8 / 2\}} = 2^{100}$ .

\* Memory:  $2^{100}$ .

=== Complexity of second preimage attack

\* Hash function call:  $2^{200}$ ,

\* Memory: none.

== LUX-512

LUX-512 has the same weakness as LUX-256 so that the 56(=8\*7) bits of the output are determined by the remaining 512-56=456 bits.

In the similar manner to the attacks on LUX-256, the collision attack and the second preimage attack require  $2^{\{456/2\}} = 2^{228}$  and  $2^{456}$  complexity respectively.

[1] Shuang Wu, Dengguo Feng, Wenling Wu

Cryptanalysis of the Hash Function LUX-256

[http://ehash.iaik.tugraz.at/uploads/3/36/Analysis\\_LUX\\_1.pdf](http://ehash.iaik.tugraz.at/uploads/3/36/Analysis_LUX_1.pdf)

Regards,

Dai Watanabe