From: Raghvendra Rohit <iraghvendrarohit@gmail.com>
Sent: Tuesday, April 23, 2019 1:57 PM
To: lightweight-crypto
Cc: zhangwentao@iie.ac.cn; lwc-forum@list.nist.gov
Subject: OFFICIAL COMMENT : KNOT
Attachments: knot-preimage-security.pdf

Dear all,

I think that the preimage security level of KNOT Hash instances is lower than the claimed ones. Based on my understanding of the specification, the security level are as follows:

****

Hash instance	**: Claimed security (bit)**	**: Actual security (bit)**
KNOT-Hash(256, 256, 32, 128)	: 128	: **125.28**
KNOT-Hash(384, 384, 48, 192)	: 192	: **187.92**
KNOT-Hash(512, 512, 64, 256)	: 256	: **250.57**

****
*****

Attached is a short note on the attack details.
Is the attack's interpretation correct?

Thanks,
Raghav

Preimage Security of KNOT-Hash

Raghvendra Rohit

Department of Electrical and Computer Engineering, University of Waterloo.
rsrohit@uwaterloo.ca

Abstract. KNOT is a Round 1 submission of the ongoing NIST lightweight cryptography project. In this short note, we show that the preimage security of KNOT-Hash instances with squeezing rate half the state size is lower than the claimed security. Our attack exploits the non-randomness properties of the KNOT Sbox which reduce the preimage complexities.
In particular, if $2n$ is the squeezing rate then the preimage security is approximately $(\log_2(\frac{3}{4}))^{-n} \times 2^{\frac{3n}{4}} \times (\log_2(3))^{\frac{n}{2}}$. For $n = 64$, 96 and 128, the former bound translates to $2^{125.28}$, $2^{187.92}$ and $2^{250.57}$, respectively.

Keywords: KNOT · NIST lightweight cryptography project · Preimage

1 The KNOT Permutation

KNOT is an SPN based iterative permutation [ZDY+]. The b-bit state can be viewed as

$$\begin{bmatrix} a_{0,\frac{b}{4}-1} & a_{0,\frac{b}{4}-2} & \cdots & a_{0,1} & a_{0,0} \\ a_{1,\frac{b}{4}-1} & a_{1,\frac{b}{4}-2} & \cdots & a_{1,1} & a_{1,0} \\ a_{2,\frac{b}{4}-1} & a_{2,\frac{b}{4}-2} & \cdots & a_{2,1} & a_{2,0} \\ a_{3,\frac{b}{4}-1} & a_{3,\frac{b}{4}-2} & \cdots & a_{3,1} & a_{3,0} \end{bmatrix}$$

A round consists of the following transformations:

1. Addition of round constant to row 0.
2. Application of 4 bit Sbox column wise.
3. Left cyclic shift row 1, row 2 and row 3 by some constants.

The round constants, rotation constants and number of rounds of KNOT permutation does not affect our analysis. We refer the reader to [ZDY+] for their respective details.

2 Observations on the KNOT Sbox

The 4-bit KNOT Sbox and the corresponding truth table values are given in Table 1 and 2.

Table 1: KNOT Sbox

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(x)$	4	0	A	7	B	E	1	D	9	F	6	8	5	2	C	3

Note the following observations from Table 2.

Observation 1. $\mathsf{Pr}(x_0 = y_2 + y_3 + 1) = \frac{3}{4}$ (gray rows in Table 2).

Observation 2. For rows satisfying observation 1, the number of sbox input/output corresponding to $(y_1, y_0) = (0,0), (1,0), (0,1)$ and $(1,1)$ are 2, 4, 3 and 3, respectively.

Table 2: KNOT Sbox truth table

x_3	x_2	x_1	x_0	y_3	y_2	y_1	y_0
0	0	0	0	0	1	0	0
0	0	0	1	0	0	0	0
0	0	1	0	1	0	1	0
0	0	1	1	0	1	1	1
0	1	0	0	1	0	1	1
0	1	0	1	1	1	1	0
0	1	1	0	0	0	0	1
0	1	1	1	1	1	0	1
1	0	0	0	1	0	0	1
1	0	0	1	1	1	1	1
1	0	1	0	0	1	1	0
1	0	1	1	1	0	0	0
1	1	0	0	0	1	0	1
1	1	0	1	0	0	1	0
1	1	1	0	1	1	0	0
1	1	1	1	0	0	1	1

3 Preimage attack on KNOT-Hash

The KNOT permutation adopts the sponge mode to provide hashing functionality. We denote an instance of hash by KNOT-Hash($b, 4n, r, 2n$) where b, $4n$, r and $2n$ denote the size of state, message digest, input rate and squeezing rate in bits, respectively. Our attack works on the following instances: KNOT-Hash($256, 256, 32, 128$), KNOT-Hash($384, 384, 48, 192$) and KNOT-Hash($512, 512, 64, 256$), the first one being the primary recommendation by designers. The designers claim preimage security level equals the squeezing rate in all three instances.

We now show the attack procedure for KNOT-Hash($256, 256, 32, 128$). The details for the other instances are similar and hence omitted.

Attack procedure for KNOT-Hash($256, 256, 32, 128$). Our attack is independent of the number of rounds of KNOT permutation as we exploit the following properties.

1. Sbox observations as given in Section 2.
2. No linear layer after the Sbox layer.

Note that message digest bits are squeezed from the first two rows. Thus, we can do the inverse shift operation and obtain the state (only 128 bts are known) before Shiftrows operation (this is actually the output of last round Sbox layer). The attack then work as follows:

1. Collect $q = (\log_2(\frac{3}{4}))^{-64} \approx 2^{26.56}$ random message digests (apply observation 1 on 64 sboxes). Since, $2^{26.56} < 2^{32}$, the preimage consists of only 1 message block.
2. For each message digest, the expected number of states is $2^{16} \times 4^{16} \times (\log_2(3))^{16} \times (\log_2(3))^{16} \approx 2^{98.72}$. This is because we have 64 sboxes and each known (y_1, y_0) tuple occurs on average $\frac{1}{4}$ times. Now apply inverse of KNOT permutation and denote the output state by $S_r'\|S_c'$. Also, let $S_r||S_c$ be the state after the initialization. If $S_c = S_c'$, the preimage is then $S_r \oplus S_r'$.
3. Repeat step 2 for q message digests.
   The overall time complexity is $q \times 2^{98.72} \approx 2^{125.28}$. Note that by Step 1, on average we will get the preimage of one message digest.

4 Concluding Remarks

In this note, we have shown that the preimage security of KNOT-Hash$(256, 256, 32, 128)$, KNOT-Hash$(384, 384, 48, 192)$ and KNOT-Hash$(512, 512, 64, 256)$ are $2^{125.28}$, $2^{187.92}$ and $2^{250.57}$, compared to 2^{128}, 2^{192} and 2^{256}, respectively.

Note that similar attack is applicable to KNOT-AEAD instances where tag size is half the state size. However, the time complexities are greater than the claimed security but requires low data and are better than the average exhaustive key search.

References

[ZDY+] Wentao Zhang, Tianyou Ding, Bohan Yang, Zhenzhen Bao, Zejun Xiang, Fulei Ji, and Xuefeng Zhao. KNOT: Round 1 Submission to NIST-LWC. 2019. https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/round-1/spec-doc/KNOT-spec.pdf.

From: 包珍珍 <baozhenzhen10@gmail.com>
Sent: Monday, April 29, 2019 11:11 AM
To: Raghvendra Rohit
Cc: lightweight-crypto; 张文涛; lwc-forum@list.nist.gov
Subject: Re: [lwc-forum] OFFICIAL COMMENT : KNOT

Dear Raghvendra,

Thank you for your interest in KNOT.

We can understand the Observations on the KNOT Sbox in your note.

However, we think the preimage attack (described in Sect. 3 Preimage attack on KNOT-Hash) has some problems:
1. Step1: For the randomly collected 2^{26.56} message digests, how the attacker ensures that the preimages (or one of the preimages) consist of only 1 message block?
The size of the message digests is 256 bits. The message digests corresponding to messages of only one block have at most 2^{32} possible values.
So, we think, for the given 2^{26.56} message digests, or the randomly collected 2^{26.56} message digests, the probability for one of them be the message digests of a preimage consisting of 1 message block is quite small.
Accordingly, we think the attacker cannot ensure that at least one of the 2^{26.56} message digests comes from an internal state of which all 64 S-boxes fulfill the equation in Observation 1 and at the same time it's preimage consists of only 1 message block.

2. Step 2: Indeed, by randomly collecting 2^{26.56} message digests, we can expect one of the 2^{26.56} message-digests squeezed from an internal state of which all 64 S-boxes fulfill the equation in Observation 1 (i.e., all 64 S-boxes of the state has input/output pairs falling into the 12 gray rows in Table 2).
However, for such an internal state, we think one cannot expect that:
"there are 1/4 of its 64 S-boxes whose (y1, y0) takes value (0,0), 1/4 of its 64 S-boxes whose (y1, y0) takes value (1,0), 1/4 of its 64 S-boxes whose (y1, y0) takes value (0,1), and 1/4 of its 64 S-boxes whose (y1, y0) takes value (1,1)".
Instead, for such an internal state, we think we should expect that:
"there are 2/12 of its 64 S-boxes whose (y1, y0) takes value (0,0), 4/12 of its 64 S-boxes whose (y1, y0) takes value (1,0), 3/12 of its 64 S-boxes whose (y1, y0) takes value (0,1), and 3/12 of its 64 S-boxes whose (y1, y0) takes value (1,1)".
Accordingly, for a message digest squeezed from an internal state of which all 64 S-boxes fulfill the equation in Observation 1, the expected number of candidate values for the internal state should be: 2^{(2/12) * 64} \times 4^{(4/12) * 64} \times 3^{(3/12) * 64} \times 3^{(3/12) * 64} = 2^104.

For these problems, we think the attack cannot work in its current form.

By the way, there are two small points in your note we think you meant as follows:
1. Sect. 3 Attack Step 1: $q = (log_2(3/4))^{-64}$ => $q = (3/4)^{-64}$
2. Sect. 3 Attack Step 2: $2^{16} \times 4^{16} \times {log_2(3)}^{16} \times {log_2(3)}^{16}$ => $2^{16} \times 4^{16} \times 3^{16} \times 3^{16}$

Thanks and regards,
KNOT team

On Wed, Apr 24, 2019 at 1:57 AM Raghvendra Rohit <iraghvendrarohit@gmail.com> wrote:

> Dear all,

From: Raghvendra Rohit <iraghvendrarohit@gmail.com>
Sent: Monday, April 29, 2019 6:31 PM
To: 包珍珍
Cc: lightweight-crypto; 张文涛; lwc-forum@list.nist.gov
Subject: Re: [lwc-forum] OFFICIAL COMMENT : KNOT

Dear KNOT Team,

Thanks a lot for looking into the note and pointing out the inconsistencies in Step 1 and Step 2. I agree that computations
were wrong which lead to overall complexities more than 2^{n/2} where n is the digest size.

However, I still believe that the absence of diffusion layer within the rows after the Sbox layer could somehow be exploited,
and needs further investigation.

Thanks and Regards,
Raghav

On Mon, Apr 29, 2019 at 11:11 AM 包珍珍 <baozhenzhen10@gmail.com> wrote:

> Dear Raghvendra,
>
> Thank you for your interest in KNOT.
>
> We can understand the Observations on the KNOT Sbox in your note.
>
> However, we think the preimage attack (described in Sect. 3 Preimage attack on KNOT-Hash) has some problems:
> 1. Step1: For the randomly collected 2^{26.56} message digests, how the attacker ensures that the preimages (or one of the preimages) consist of only 1 message block?
> The size of the message digests is 256 bits. The message digests corresponding to messages of only one block have at most 2^{32} possible values.
> So, we think, for the given 2^{26.56} message digests, or the randomly collected 2^{26.56} message digests, the probability for one of them be the message digests of a preimage consisting of 1 message block is quite small.
> Accordingly, we think the attacker cannot ensure that at least one of the 2^{26.56} message digests comes from an internal state of which all 64 S-boxes fulfill the equation in Observation 1 and at the same time it's preimage consists of only 1 message block.
>
> 2. Step 2: Indeed, by randomly collecting 2^{26.56} message digests, we can expect one of the 2^{26.56} message-digests squeezed from an internal state of which all 64 S-boxes fulfill the equation in Observation 1 (i.e., all 64 S-boxes of the state has input/output pairs falling into the 12 gray rows in Table 2).
> However, for such an internal state, we think one cannot expect that:
> "there are 1/4 of its 64 S-boxes whose (y1, y0) takes value (0,0), 1/4 of its 64 S-boxes whose (y1, y0) takes value (1,0), 1/4 of its 64 S-boxes whose (y1, y0) takes value (0,1), and 1/4 of its 64 S-boxes whose (y1, y0) takes value (1,1)".
> Instead, for such an internal state, we think we should expect that:
> "there are 2/12 of its 64 S-boxes whose (y1, y0) takes value (0,0), 4/12 of its 64 S-boxes whose (y1, y0) takes value (1,0), 3/12 of its 64 S-boxes whose (y1, y0) takes value (0,1), and 3/12 of its 64 S-boxes whose (y1, y0) takes value (1,1)".