Dear all:
There is a small error in the reference implementation of Lotus.
When the nonce is mixed with the key, only CRYPTO_ABIES of the nonce are used. As a result, the cipher behaves as one with a 64 bit-nonce, instead of the specified 128.

Best regards
Miguel Montes
Dear Miguel,

Thanks for pointing out the bug in the reference implementation of LOTUS-AEAD.

Dear all,

Specifically, the bug was at line 96 of encrypt.c file of LOTUS-AEAD implementation.

Incorrect version: "xor_bytes(nonced_key, nonce, CRYPTO_ABYTES);"

Correct version: "xor_bytes(nonced_key, nonce, CRYPTO_NPUBBYTES);"

We have fixed the bug in the reference implementation (also attached here).

NOTE: The bug pertains to the reference implementation and does not require any change in the specification of LOTUS-AEAD.

Regards,
LOTUS-AEAD and LOCUS-AEAD Team

On Sun, Apr 28, 2019 at 1:43 AM Miguel Montes <miguel.montes@gmail.com> wrote:
> Dear all:
> There is a small error in the reference implementation of Lotus.
> When the nonce is mixed with the key, only CRYPTO_ABYTES of the nonce are used. As a result, the cipher behaves as one with a 64 bit-nonce, instead of the specified 128.
> Best regards
> Miguel Montes
> --
> To unsubscribe from this group, send email to
> lwc-forum+unsubscribe@list.nist.gov
> Visit this group at
> https://groups.google.com/a/list.nist.gov/d/forum/lwc-forum
> ---
> You received this message because you are subscribed to the Google Groups "lwc-forum" group.
> To unsubscribe from this group and stop receiving emails from it, send an email to lwc-forum+unsubscribe@list.nist.gov.
From: MEGE, Alexandre <alexandre.mege@airbus.com>
Sent: Monday, June 3, 2019 12:28 PM
To: lightweight-crypto
Cc: lwc-forum@list.nist.gov
Subject: OFFICIAL COMMENT: LOTUS-AEAD and LOCUS-AEAD

Dear All,

It seems locus and lotus are vulnerable against forgery attack. I have found collisions between a message with empty Associated Data and a message with AD = PT || PT. I was also able to find collision between messages with empty PT by adding zeros at the end of AD.

Ex for twegift64locusaeadv1:

- First example

Key = 000102030405060708090A0B0C0D0E0F
Nonce = 000102030405060708090A0B0C0D0E0F
PT = 000001010202030405060606060707
AD = 0000010102020303040505050505050606060707
CT = 6994E43F3496F6821EC1DE1A5EE1C34423FC0961F413508F

Key = 000102030405060708090A0B0C0D0E0F
Nonce = 000102030405060708090A0B0C0D0E0F
PT = 000001010202030304050506060707
AD =
CT = 6994E43F3496F6821EC1DE1A5EE1C34423FC0961F413508F

- Second example

Key = 000102030405060708090A0B0C0D0E0F
Nonce = 000102030405060708090A0B0C0D0E0F
PT = 000001010202030304050506060707
AD = 00000101020203030000010102020303
CT = 1AC5DA1E5AE5C740705DA2B38E8E616B

Key = 000102030405060708090A0B0C0D0E0F
Nonce = 000102030405060708090A0B0C0D0E0F
PT = 0000010102020303
AD =
CT = 1AC5DA1E5AE5C740705DA2B38E8E616B

- Collisions with zero padding and empty PT:

Key = 000102030405060708090A0B0C0D0E0F
Nonce = 000102030405060708090A0B0C0D0E0F
PT =
AD = 0000000000010000
CT = BAFA57086BEB963D

Key = 000102030405060708090A0B0C0D0E0F
Nonce = 000102030405060708090A0B0C0D0E0F
PT =
AD = 000000000001000000000000000000000000000000000000
CT = BAFA57086BEB963D

Best regards,
Alexandre Mège
Hi all,

The observation by Alexandre holds true only when key = Nonce. The reason is in whenever K = N, K_N = K + N = 0^n => L = 0. (Line 12, Line 14 of Algorithm 1 in specs. document). Thus, the output v_xor after after processing the associated data is same (L = 0 => all keys are zero in proc_ad function). Hence, the tags are same.

PS: Attached is the locus code for verification.

--------
Thanks,
Raghav

On Monday, June 3, 2019 at 12:28:30 PM UTC-4, alexandre.mege wrote:

Dear All,

It seems locus and lotus are vulnerable against forgery attack.

I have found collisions between a message with empty Associated Data and a message with AD = PT || PT.

I was also able to find collision between messages with empty PT by adding zeros at the end of AD.

Ex for twegift64locusaeadv1:

- First example

Key = 000102030405060708090A0B0C0D0E0F
Nonce = 000102030405060708090A0B0C0D0E0F
PT = 00000101020304050607060707
Dear Alexandre,

Thanks for showing an interest in LOTUS-AEAD and LOCUS-AEAD.

As Raghav rightly pointed out, the attack works only when some nonce collides with the master key.

Since the 128-bit master key is chosen uniformly at random, the probability that it equals a fixed nonce value is $1/2^{128}$.

One can make at most $2^{64}$ queries to the AE scheme, say each with distinct nonce value. Then, the attack succeeds with at most $1/2^{64}$ probability.

Consequently, this does not disprove the security claims of LOTUS-AEAD and LOCUS-AEAD.

--
 Regards,
LOTUS-AEAD and LOCUS-AEAD Team

On Tue, 4 Jun 2019, 1:53 am Raghvendra Rohit, <iraghvendrarohit@gmail.com> wrote:

Hi all,

The observation by Alexandre holds true only when key = Nonce.
The reason is in whenever K = N, K_N = K + N = 0^n => L = 0. (Line 12, Line 14 of Algorithm 1 in specs. document). Thus, the output v_xor after after processing the associated data is same (L = 0 => all keys are zero in proc_ad function).
Hence, the tags are same.

PS: Attached is the locus code for verification.

--------
Thanks,
Raghav

On Monday, June 3, 2019 at 12:28:30 PM UTC-4, alexandre.mege wrote:

Dear All,

It seems locus and lotus are vulnerable against forgery attack.

I have found collisions between a message with empty Associated Data and a message with AD = PT || PT.
Dear LOTUS-AEAD and LOCUS-AEAD Team

Thank you for the quick feedback.
I confirm that this collision only happens if there is a collision between Key and nonce.
As noted by Ashwin, it does not impact the security claims of LOTUS-AEAD and LOCUS-AEAD.

Regards,
Alexandre Mège

From: Ashwin Jha [mailto:letterstoashwin@gmail.com]
Sent: Tuesday, June 04, 2019 3:35 AM
To: MEGE, Alexandre
Cc: lwc-forum@list.nist.gov; lightweight-crypto@nist.gov; iraghvendrarohit@gmail.com; avik chakraborti; Nilanjan Datta; cuauhtemoc.mancillas83@gmail.com; sasaki.yu@lab.ntt.co.jp; Ashwin Jha
Subject: Re: [lwc-forum] OFFICIAL COMMENT: LOTUS-AEAD and LOCUS-AEAD

Dear Alexandre,

Thanks for showing an interest in LOTUS-AEAD and LOCUS-AEAD.

As Raghav rightly pointed out, the attack works only when some nonce collides with the master key.

Since the 128-bit master key is chosen uniformly at random, the probability that it equals a fixed nonce value is $1/2^{128}$.

One can make at most $2^{64}$ queries to the AE scheme, say each with distinct nonce value. Then, the attack succeeds with at most $1/2^{64}$ probability.

Consequently, this does not disprove the security claims of LOTUS-AEAD and LOCUS-AEAD.

--
Regards,
LOTUS-AEAD and LOCUS-AEAD Team

On Tue, 4 Jun 2019, 1:53 am Raghvendra Rohit, <iraghvendrarohit@gmail.com> wrote:

Hi all,