From:	MEGE, Alexandre <alexandre.mege@airbus.com>
Sent:	Monday, July 22, 2019 12:24 PM
To:	lightweight-crypto
Cc:	lwc-forum@list.nist.gov
Subject:	OFFICIAL COMMENT: REMUS [AD-INT]

Dear All,

It seems REMUSN3V1 is vulnerable to Key recovery attacks.

This vulnerability comes from the very simple Key Derivation function (L <= (Key Xor (N||0^{32})) .

This KDF produces an output of size 128 bit. A birthday attack will find a collision with an offline precomputed value of L with 2^64 complexity. Once a collision happens, Key recovery is trivial.

This is in contradiction with the security claim of table 3.2 claiming 128 bit security for key recovery for REMUSN3V1.

Best regards,

Alexandre Mège

From: Ashwin Jha <letterstoashwin@gmail.com>
Sent: Wednesday, July 24, 2019 7:14 AM
To: MEGE, Alexandre; lightweight-crypto; lwc-forum@list.nist.gov
Cc: Nilanjan Datta; Mridul Nandi
Subject: Re: [lwc-forum] OFFICIAL COMMENT: REMUS [AD-INT]

Dear all,

We have a follow-up on Alexandre's observation. By viewing the key recovery attack as a state recovery attack one can actually construct forgery attacks on REMUS-N1 (primary version), REMUS-N3, and REMUS-M1.
On a closer inspection, we found that the same attack also works on
TGIF-N1 (primary version) and TGIF-M1 as well. Basically, the attack recovers the nonce-based key L, which can be used to construct valid forgeries.

The forgery attack works as follows:

Suppose the key derivation function KDF_K takes a nonce value N as input and outputs a nonce-based key L.

1. For $i = 1$ to 2^{a}:
i. Sample L^i in without replacement manner from $\{0,1\}^{128}$.
ii. Simulate the encryption of (A,M) using L^i as the nonce-based key, where $|A| = n$ and $M = M_1||M_2||M_3||M_4$, where each $|M_k| = n$ for $k \in \{1,2,3,4\}$. Response: (C^i,T^i) where $C^i=C^i_1||C^i_2||C^i_3||C^i_4$. Store (L^i,C^i,T^i) in a list H.

2. Sort H on (C,T).

3. For $j = 1$ to 2^{128-a}:
i. Query (N^j,A,M) to the encryption oracle of AE. Response:
(C^j,T^j). Search (C^j,T^j) in H (binary search would suffice).
ii. Suppose there exist index $i \in H$ such that $(C^j,T^j) = (C^i,T^i)$ then it would mean that $L^j = L^i$ with very high probability (matching on all 5 blocks of ciphertext and tag helps in avoiding false positives).

4. Now, the adversary can easily construct forgeries given the nonce-based key L^j for nonce value N^j.

The query complexity is 2^{128-a} and the total time complexity is $2^{a} + a2^{a} + a2^{128-a}$, where the first, second, and third terms correspond to time complexity of step 1, 2, and 3, respectively. We ignore the time complexity of step 4, as it can be made negligible.
Clearly, for good choice of a, say a=90, we get query complexity approx. 2^{38} and time complexity $< 2^{100}$ (using $a < 128$).
This is well within the data and time limit set by NIST. Note that, the forgery attack is in nonce-respecting model.

We remark that, here we count the direct block cipher calls within time complexity as is the norm for ideal cipher model (REMUS and TGIF use ICM). This is also plausible in real scenario where the adversary can actually make block cipher evaluations on its own by devoting sufficient time. In this regard, we also note that according to NIST's requirement, the adversary can not be restricted to at most 2^{64} and 2^{60} offline queries (as is directed on page 22 of REMUS
specification) in REMUS-N1/M1/TGIF-N1/M1 and REMUS-N3, respectively.
This is especially required from REMUS-N1 and TGIF-N1, which are the primary variants in their respective submissions.

In summary, it seems that REMUS-N1/N3/M1 and TGIF-N1/M1 do not satisfy NIST requirements.

The N1/N3/M1 variants of REMUS (and TGIF) have an inherent weakness:

insufficient randomness in the initial state (key,input). Although the key is derived using nonce for each encryption query, the adversary can easily fix a constant value as the initial input. So, to create an initial state collision the adversary just needs to collide the initial key.

Thanks and Regards,
Nilanjan Datta, Ashwin Jha, Mridul Nandi

>

From: Thomas Peyrin (Assoc Prof) <thomas.peyrin@ntu.edu.sg>
Sent: Friday, July 26, 2019 12:50 PM
To: Ashwin Jha; MEGE, Alexandre; lightweight-crypto; lwc-forum@list.nist.gov
Cc: Nilanjan Datta; Mridul Nandi
Subject: RE: [lwc-forum] OFFICIAL COMMENT: REMUS [AD-INT]

Dear Alexandre, Nilanjan, Ashwin, Mridul,

Thanks a lot for analysing our candidate REMUS. We agree with the analysis you described (except that the complexity of Alexandre's key recovery method on REMUS-N3 should be 80-bit instead of 64-bit *), but we don't agree with the fact that it is a weakness. Actually, these results don't contradict our claims. Nilanjan-Ashwin-Mridul's attack is within our security proof and claims. Regarding Alexandre's method, we do agree that our table 3.2 should have been (much much) clearer, but it is to be understood with negligible online queries: "For key recovery, the adversary needs to find the k=128-bit key used in KDF with 2^k offline queries (computations)".

More generally, the question about REMUS-M1/N1/N3 fulfilling the NIST requirements also stems from the fact that the NIST document is a bit vague regarding the allowable security requirements ("security against 2^112 computations"). The claimed security of REMUS-M1/N1 or REMUS-N3 is indeed highly dependent on the amount of online queries, but these modes should not be used close to their online complexity limits. Yet, we believe such candidates with extremely efficient performance figures would be interesting in some LWC scenarios.

In any case, we emphasize that our security claim for REMUS-M2/N2 is full 128-bit and thus is surely within NIST requirements. All in all, we do not plan to change our submission REMUS. However, if the NIST judges that our claims for M1/N1/N3 variants do not follow the requirements, of course we will only keep M2/N2 versions (which has only slightly worse performance than M1/N1/N3).

TGIF using the N1/N2 modes as well (not N3), our comments equally apply to this NIST candidate.

Regards,

The REMUS team.

* One can't do a birthday on the full L value in REMUS-N3 as part of it (32-bit to be precise) is not randomized with N (since N is only 96 bits). Thus, you would have to build/maintain a table of 2^80 offline and make 2^48 online queries.

-----Original Message-----
From: Ashwin Jha [mailto:letterstoashwin@gmail.com]
Sent: Wednesday, 24 July, 2019 13:14
To: MEGE, Alexandre <alexandre.mege@airbus.com>; lightweight-crypto@nist.gov; lwc-forum@list.nist.gov
Cc: Nilanjan Datta <nilanjan_isi_jrf@yahoo.com>; Mridul Nandi <mridul.nandi@gmail.com>
Subject: Re: [lwc-forum] OFFICIAL COMMENT: REMUS [AD-INT]

Dear all,

We have a follow-up on Alexandre's observation. By viewing the key recovery attack as a state recovery attack one can actually construct forgery attacks on REMUS-N1 (primary version), REMUS-N3, and REMUS-M1.
On a closer inspection, we found that the same attack also works on
TGIF-N1 (primary version) and TGIF-M1 as well. Basically, the attack recovers the nonce-based key L, which can be used to construct valid forgeries.

From: MEGE, Alexandre <alexandre.mege@airbus.com>
Sent: Friday, July 26, 2019 1:21 PM
To: Thomas Peyrin (Assoc Prof); Ashwin Jha; lightweight-crypto; lwc-forum@list.nist.gov
Cc: Nilanjan Datta; Mridul Nandi
Subject: RE: [lwc-forum] OFFICIAL COMMENT: REMUS [AD-INT]

[AIRBUS DEFENCE AND SPACE INTERNAL]
Dear Thomas,

I fully agree with you that NIST security requirements "security against 2^112 computations" is too vague, especially coupled with the minimal input size "2^50-1 Bytes".

In fact, those requirements have been understood very differently across LWC submissions:

* Some submissions have 128 bits internal state with a birthday attacks in 2^64 ciphered block complexity, and around 88 bits complexity within the limit of 2^50 Bytes per key.
* Some submission have 160 bits internal state, claiming 112 bits security within the limit of 2^50 Bytes per key.
* Some submissions, mostly sponge-based ones, have at least 256 bits internal state with at least 224 bit capacity. They can claim unconditional 112 bits security.

A figure showing the claimed security versus online the number of online ciphered Byte would show very different results for all LWC candidates.
Anyone interesting to collect this data?

Best regards,
Alexandre Mège

THIS DOCUMENT IS NOT SUBJECT TO EXPORT CONTROL.

-----Original Message-----
From: Thomas Peyrin (Assoc Prof) [mailto:thomas.peyrin@ntu.edu.sg]
Sent: Friday, July 26, 2019 6:50 PM
To: Ashwin Jha; MEGE, Alexandre; lightweight-crypto@nist.gov; lwc-forum@list.nist.gov
Cc: Nilanjan Datta; Mridul Nandi
Subject: RE: [lwc-forum] OFFICIAL COMMENT: REMUS [AD-INT]

Dear Alexandre, Nilanjan, Ashwin, Mridul,

Thanks a lot for analysing our candidate REMUS. We agree with the analysis you described (except that the complexity of Alexandre's key recovery method on REMUS-N3 should be 80-bit instead of 64-bit *), but we don't agree with the fact that it is a weakness. Actually, these results don't contradict our claims. Nilanjan-Ashwin-Mridul's attack is within our security proof and claims. Regarding Alexandre's method, we do agree that our table 3.2 should have been (much much) clearer, but it is to be understood with negligible online queries: "For key recovery, the adversary needs to find the k=128-bit key used in KDF with 2^k offline queries (computations)".

From: Ashwin Jha <letterstoashwin@gmail.com>
Sent: Saturday, July 27, 2019 5:15 AM
To: Thomas Peyrin (Assoc Prof)
Cc: MEGE, Alexandre; lightweight-crypto; lwc-forum@list.nist.gov; Nilanjan Datta; Mridul Nandi
Subject: Re: [lwc-forum] OFFICIAL COMMENT: REMUS [AD-INT]

Dear Thomas and all,

The authenticity security bounds of REMUS-N1/M1 and TGIF-N1/M1 (N1/M1 in short) has a term O(q_p\sigma/2^n) where q_p, \sigma, and n denote the number of offline block cipher evaluations, total number of effective blocks in the encryption and decryption queries, and the block size, respectively. We agree that our attacks validate the tightness of authenticity security bounds of N1/M1. However, our point is that N1/M1 have inadequate security as per the NIST requirements.

The NIST call clearly states that any attack should require at least 2^{112} computations or 2^{50}-1 bytes of data. While we agree that "2^{112} computations" is quite vague, yet following the prevalent sense one can conclude that this is a bound on the total time complexity (both online and offline) of the attack.

Coming back to our attacks on N1/M1, as mentioned in the previous email, they are well within the NIST limits as the number of computations is less than 2^{100} (the number of offline block cipher evaluations is about 2^{96}) and the amount of data is about 2^{38} messages of length 4 (i.e. around 2^{44.33} bytes). Actually, the attack will work with any message of length 2 or more.

Further, as Thomas emphasized
"The claimed security of REMUS-M1/N1 or REMUS-N3 is indeed highly dependent on the amount of online queries, but these modes should not be used close to their online complexity limits.", we remark that the amount of online queries of our attack is actually well below 2^{64} bytes. As a side note, it's actually not clear from the specifications whether the online limit is in bytes or block size (i.e. 16 bytes).

After a closer inspection, we see that on page 22 of REMUS's specification, it is mentioned that "an integer x in the table means an attack possibly breaks the scheme with online query complexity Q_{online} and offline query complexity Q_{offline} if \max\{Q_{online},Q_{offline}\} = 2^x.".

For N1/M1, x is bounded by 64, which is not adequate, as the offline queries are actually offline evaluations of the block cipher, which should really be considered in the time complexity of the attacker.
Indeed the attacker can make the necessary offline block cipher evaluations on its own time. Clearly, the attacker has a much higher quota of time, i.e. 2^{112}.

We think that this line of argument (bounding the "offline queries" to
2^{64}) is flawed. As a matter of fact, by this logic the state-of-the-art security bounds on Sponge duplex guarantee that a construction with 128-bit capacity and 64-bit rate satisfies the NIST criteria, as here again one can bound the number of offline permutation evaluations to 2^{64}.

Regards,
Nilanjan, Ashwin, Mridul

On Fri, Jul 26, 2019 at 10:51 PM MEGE, Alexandre <alexandre.mege@airbus.com> wrote:
>
> [AIRBUS DEFENCE AND SPACE INTERNAL]
> Dear Thomas,
>

From: Thomas Peyrin (Assoc Prof) <thomas.peyrin@ntu.edu.sg>
Sent: Sunday, July 28, 2019 3:52 AM
To: Ashwin Jha
Cc: MEGE, Alexandre; lightweight-crypto; lwc-forum@list.nist.gov; Nilanjan Datta; Mridul Nandi
Subject: RE: [lwc-forum] OFFICIAL COMMENT: REMUS [AD-INT]

Hi,

Yes, we agree that your attack validates the tightness of our security proof. You agree with us that NIST's "2^{112} computations" is quite vague, and you interpret it as "this is a bound on the total time complexity (both online and offline) of the attack". This is surely a possible interpretation, but we don't necessarily see it the same way. In any case, regardless of the interpretation, we recall that M1/N1/N3 are the aggressive versions of REMUS, and the N2/M2 versions are undoubtedly within the NIST requirements.

Regarding your paragraph about "flawed argument", please note that Table 3.1 shows the representative and conservative case from the curve suggested by the security bounds. They are security bounds, and it of course does not limit the offline query to 2^64 when the online query is limited. This is significantly different from the case of Sponge with 128-bit capacity and 64-bit rate, and we think the security of REMUS is very different from your Sponge example. We would like to make clear that we did not mention at any moment that your Sponge example is secure.

We believe that N1/N3 versions are interesting, because of their extremely good performances. Of course, there are compromise that need to be done to get such performances, but other designs also do make security compromise and it is quite a multi-dimensional problem. Take for example ASCON (or any of the many sponge-based candidates which don't use the hermetic-sponge principle): it is a sponge based design for which very efficient distinguishers exist for the internal permutation used. This means that the assumptions for the security proof don't hold. We understand that this compromise is done to obtain an efficient design, but then one cannot directly compare with other designs (like REMUS) for which security assumptions are not disproven and therefore provide a very important safety net. Basically, REMUS-N1/N3 do compromise on the actual bounds, but no compromise on the security assumptions; while non-hermetic sponge candidates don't do compromise on the actual bounds, but they do on the security assumptions. We are not sure which one is the best approach, but probably both are directions worth to be tried.

Regards,

The REMUS team.

-----Original Message-----
From: Ashwin Jha [mailto:letterstoashwin@gmail.com]
Sent: Saturday, 27 July, 2019 11:15
To: Thomas Peyrin (Assoc Prof) <thomas.peyrin@ntu.edu.sg>
Cc: MEGE, Alexandre <alexandre.mege@airbus.com>; lightweight-crypto@nist.gov; lwc-forum@list.nist.gov; Nilanjan Datta <nilanjan_isi_jrf@yahoo.com>; Mridul Nandi <mridul.nandi@gmail.com>
Subject: Re: [lwc-forum] OFFICIAL COMMENT: REMUS [AD-INT]

Dear Thomas and all,

The authenticity security bounds of REMUS-N1/M1 and TGIF-N1/M1 (N1/M1 in short) has a term O(q_p\sigma/2^n) where q_p, \sigma, and n denote the number of offline block cipher evaluations, total number of effective blocks in the encryption and decryption queries, and the block size, respectively. We agree that our attacks validate the tightness of authenticity security bounds of N1/M1. However, our point is that N1/M1 have inadequate security as per the NIST requirements.

From: MEGE, Alexandre <alexandre.mege@airbus.com>
Sent: Monday, July 29, 2019 4:37 AM
To: Thomas Peyrin (Assoc Prof); Ashwin Jha
Cc: lightweight-crypto; lwc-forum@list.nist.gov; Nilanjan Datta; Mridul Nandi
Subject: RE: [lwc-forum] OFFICIAL COMMENT: REMUS [AD-INT]

Dear Thomas,
Thank you for your answer and very interesting discussion.

I come back on the cost of the key recovery attack on REMUS-N3, you estimated it at 2^80 offline queries with following argument:
>> * One can't do a birthday on the full L value in REMUS-N3 as part of
>> it (32-bit to be precise) is not randomized with N (since N is only 96 bits). Thus, you would have to build/maintain a table of 2^80 offline and make 2^48 online queries.

I think the attack complexity remains at 2^64 with following proof:

This attack looks for collision after KDF on the value of 'L' : 'L' <= (Key Xor (N||0^32) One can find with probability 1 a collision in complexity 2^64 offline cipher calls and 2^64 online cipher calls with those sets:
* Online cipher Calls : Call the cipher for all the nonces from 0 to 2^64-1 (ie try all the possibilities for the 64 Lsb)
    => The 64 LSB of 'L' will cover all the 2^64 possible values, including all '0'. The 64 MSBs are the 64 MSBs of the Key.
* Offline cipher calls: Call the ciphers with 'L' with the 64 LSB fixed to 0 and the 64 MSB covering all the 2^64 possible values.
    => The two sets will collide for 'L' with the 64 MSB of the Key and 64 LSB at '0'.

A computation complexity of 2^81 offline calls holds if there is a limitation on the maximum number of online calls of 2^50 Bytes (=>2^47 maximum online calls).

Best regards,
Alexandre Mège

THIS DOCUMENT IS NOT SUBJECT TO EXPORT CONTROL.

-----Original Message-----
From: Thomas Peyrin (Assoc Prof) [mailto:thomas.peyrin@ntu.edu.sg]
Sent: Sunday, July 28, 2019 9:52 AM
To: Ashwin Jha
Cc: MEGE, Alexandre; lightweight-crypto@nist.gov; lwc-forum@list.nist.gov; Nilanjan Datta; Mridul Nandi
Subject: RE: [lwc-forum] OFFICIAL COMMENT: REMUS [AD-INT]

Hi,

Yes, we agree that your attack validates the tightness of our security proof. You agree with us that NIST's "2^{112} computations" is quite vague, and you interpret it as "this is a bound on the total time complexity (both online and offline) of the attack". This is surely a possible interpretation, but we don't necessarily see it the same way. In any case, regardless of the

From: Thomas Peyrin (Assoc Prof) <thomas.peyrin@ntu.edu.sg>
Sent: Tuesday, July 30, 2019 8:27 PM
To: MEGE, Alexandre; Ashwin Jha
Cc: lightweight-crypto; lwc-forum@list.nist.gov; Nilanjan Datta; Mridul Nandi
Subject: RE: [lwc-forum] OFFICIAL COMMENT: REMUS [AD-INT]

Dear Alexandre,

Thanks for your clarification, we may have misunderstood your previous message. We agree with the new analysis you described and we think what you are proposing is a birthday attack within our bound.
We would like to note that your new analysis does not change our previous response.

Regards,

The REMUS Team

-----Original Message-----
From: MEGE, Alexandre [mailto:alexandre.mege@airbus.com]
Sent: Monday, 29 July, 2019 10:37
To: Thomas Peyrin (Assoc Prof) <thomas.peyrin@ntu.edu.sg>; Ashwin Jha <letterstoashwin@gmail.com>
Cc: lightweight-crypto@nist.gov; lwc-forum@list.nist.gov; Nilanjan Datta <nilanjan_isi_jrf@yahoo.com>; Mridul Nandi <mridul.nandi@gmail.com>
Subject: RE: [lwc-forum] OFFICIAL COMMENT: REMUS [AD-INT]

Dear Thomas,
Thank you for your answer and very interesting discussion.

I come back on the cost of the key recovery attack on REMUS-N3, you estimated it at 2^80 offline queries with following argument:
>> * One can't do a birthday on the full L value in REMUS-N3 as part of
>> it (32-bit to be precise) is not randomized with N (since N is only 96 bits). Thus, you would have to build/maintain a table of 2^80 offline and make 2^48 online queries.

I think the attack complexity remains at 2^64 with following proof:

This attack looks for collision after KDF on the value of 'L' : 'L' <= (Key Xor (N||0^32) One can find with probability 1 a collision in complexity 2^64 offline cipher calls and 2^64 online cipher calls with those sets:
* Online cipher Calls : Call the cipher for all the nonces from 0 to 2^64-1 (ie try all the possibilities for the 64 Lsb)
  => The 64 LSB of 'L' will cover all the 2^64 possible values, including all ‘0’. The 64 MSBs are the 64 MSBs of the Key.
* Offline cipher calls: Call the ciphers with 'L' with the 64 LSB fixed to 0 and the 64 MSB covering all the 2^64 possible values.
  => The two sets will collide for 'L' with the 64 MSB of the Key and 64 LSB at ‘0’.

A computation complexity of 2^81 offline calls holds if there is a limitation on the maximum number of online calls of 2^50 Bytes (=>2^47 maximum online calls).

Best regards,
Alexandre Mège