From: Miguel Montes <miguel.montes@gmail.com>
Sent: Monday, April 22, 2019 9:43 PM
To: lightweight-crypto; martin_zhangbin@hotmail.com
Cc: lwc-forum@list.nist.gov
Subject: OFFICIAL COMMENT: Quartet
Attachments: quartet_collisions.c

Dear all:
We have found nonce, message and ad collisions in the reference implementation of Quartet.
For instance, with key 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF, all nonces with the first 8 bytes set as zero produce the same result.

We believe this is an implementation problem. The χ function, as implemented, is not invertible, and leads to a loss of entropy.

Here are some examples (some of them using values of the KATs).

```
AD collision
Key = 000102030405060708090A0B0C0D0E0F
Nonce = 000102030405060708090A0B
PT = 0001020304050607
AD = 0001020304050607
CT = 7045497B883E2E232A7F2D0BD017E3F51230822DCDF78614

Key = 000102030405060708090A0B0C0D0E0F
Nonce = 000102030405060708090A0B
PT = 0001020304050607
AD = 4a910b5b20050607
CT = 7045497B883E2E232A7F2D0BD017E3F51230822DCDF78614

Key = 000102030405060708090A0B0C0D0E0F
Nonce = 000102030405060708090A0B
PT = 0001020304050607
AD = a881415308010607
CT = 7045497B883E2E232A7F2D0BD017E3F51230822DCDF78614

Key = 000102030405060708090A0B0C0D0E0F
Nonce = 000102030405060708090A0B
PT = 0001020304050607
AD = 02190349000d0607
CT = 7045497B883E2E232A7F2D0BD017E3F51230822DCDF78614

Key = 000102030405060708090A0B0C0D0E0F
Nonce = 000102030405060708090A0B
PT = 0001020304050607
AD = 60090941280d0607
CT = 7045497B883E2E232A7F2D0BD017E3F51230822DCDF78614

Key = 000102030405060708090A0B0C0D0E0F
Nonce = 000102030405060708090A0B
PT = 0001020304050607
AD = aa994a1920090607
CT = 7045497B883E2E232A7F2D0BD017E3F51230822DCDF78614

Message collision
```

```
Key = 000102030405060708090A0B0C0D0E0F
Nonce = 000102030405060708090A0B
PT = 0001020304050607
AD = 000102030405060708090A0B0C0D0E0F
CT = 07C4EFDFB83063F6CB624D5709DDF01C917FF792AA9E8964

Key = 000102030405060708090A0B0C0D0E0F
Nonce = 000102030405060708090A0B
PT = 0001020300050607
AD = 000102030405060708090A0B0C0D0E0F
CT = 07C4EFDFBC3063F6CB624D5709DDF01C917FF792AA9E8964

Key = 000102030405060708090A0B0C0D0E0F
Nonce = 000102030405060708090A0B
PT = 0001000300050607
AD = 000102030405060708090A0B0C0D0E0F
CT = 07C4EDDFBC3063F6CB624D5709DDF01C917FF792AA9E8964

Nonce collision
Key = FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
Nonce = 000000000000000001010101
PT = 0001020304050607
AD = 0001020304050607
CT = 8538B2ABFA3AEFE4EB63F595F2C8CC8FE535BEF78C697C3E

Key = FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
Nonce = 0000000000000000FFFFFFF
PT = 0001020304050607
AD = 0001020304050607
CT = 8538B2ABFA3AEFE4EB63F595F2C8CC8FE535BEF78C697C3E

Key = FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
Nonce = 000000000000000000A0B0C0
PT = 0001020304050607
AD = 0001020304050607
CT = 8538B2ABFA3AEFE4EB63F595F2C8CC8FE535BEF78C697C3E

Key = FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
Nonce = 000000000000000099999999
PT = 0001020304050607
AD = 0001020304050607
CT = 8538B2ABFA3AEFE4EB63F595F2C8CC8FE535BEF78C697C3E

Key = FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
Nonce = 000000000000000055555555
PT = 0001020304050607
AD = 0001020304050607
CT = 8538B2ABFA3AEFE4EB63F595F2C8CC8FE535BEF78C697C3E
```

Attached is a small program that produces the results shown

Best regards
Miguel Montes and Daniel Penazzi

From: Rotella Yann <yann.rotella@gmail.com>
Sent: Tuesday, April 23, 2019 4:27 AM
To: lwc-forum
Cc: lightweight-crypto
Subject: Re: OFFICIAL COMMENT: Quartet

Dear all,

I don't think it is only the implementation.

Indeed, it's a Chi mapping on four bits and hence it's not invertible by design: it seems there is for instance no output of Hamming weight 3 after the nonlinear mapping.

Best regards,

Y. Rotella

Le mardi 23 avril 2019 03:42:58 UTC+2, Miguel Montes a écrit :

> Dear all:
> We have found nonce, message and ad collisions in the reference implementation of Quartet.
> For instance, with key 0xFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF, all nonces with the first 8 bytes set as zero produce the same result.
>
> We believe this is an implementation problem. The χ function, as implemented, is not invertible, and leads to a loss of entropy.
>
> Here are some examples (some of them using values of the KATs).

```
AD collision
Key = 000102030405060708090A0B0C0D0E0F
Nonce = 000102030405060708090A0B
PT = 0001020304050607
AD = 0001020304050607
CT = 7045497B883E2E232A7F2D0BD017E3F51230822DCDF78614

Key = 000102030405060708090A0B0C0D0E0F
Nonce = 000102030405060708090A0B
PT = 0001020304050607
AD = 4a910b5b20050607
CT = 7045497B883E2E232A7F2D0BD017E3F51230822DCDF78614

Key = 000102030405060708090A0B0C0D0E0F
Nonce = 000102030405060708090A0B
PT = 0001020304050607
AD = a881415308010607
CT = 7045497B883E2E232A7F2D0BD017E3F51230822DCDF78614

Key = 000102030405060708090A0B0C0D0E0F
Nonce = 000102030405060708090A0B
PT = 0001020304050607
AD = 02190349000d0607
CT = 7045497B883E2E232A7F2D0BD017E3F51230822DCDF78614

Key = 000102030405060708090A0B0C0D0E0F
```

From: Zhang Bin <martin_zhangbin@hotmail.com>
Sent: Tuesday, April 23, 2019 6:11 AM
To: Rotella Yann; lwc-forum
Cc: lightweight-crypto
Subject: 回复: [lwc-forum] Re: OFFICIAL COMMENT: Quartet

Dear Yann,

Thanks for the interest on Quartet. I just saw the two posts on Quartet. Please note that each \chi sub-mapping in Quartet is defined on 3 lanes, not on 4 lanes; and there are two linear transformations after each \chi sub-function (section 1.5.7 and 1.5.6), thus it is invertible on itself by the design. If we apply the same \chi sub-function twice on the state, we will recover the original state. It is easy to find the state going into a hamming weight 3 state after each sub-function.

Dear Miguel and Daniel,

Thanks for pointing out the implementation bug in Quartet's reference code, actually this problem is also identified and will be fixed soon. There is no change of Quartet's specification.

Best Regards,

Bin Zhang

发件人: Rotella Yann <yann.rotella@gmail.com>
发送时间: 2019 年 4 月 23 日 16:27
收件人: lwc-forum
抄送: lightweight-crypto@nist.gov
主题: [lwc-forum] Re: OFFICIAL COMMENT: Quartet

Dear all,

I don't think it is only the implementation.

Indeed, it's a Chi mapping on four bits and hence it's not invertible by design: it seems there is for instance no output of Hamming weight 3 after the nonlinear mapping.

Best regards,

Y. Rotella

Le mardi 23 avril 2019 03:42:58 UTC+2, Miguel Montes a écrit :

> Dear all:

From: MEGE, Alexandre <alexandre.mege@airbus.com>
Sent: Monday, June 10, 2019 12:33 PM
To: lightweight-crypto
Cc: lwc-forum@list.nist.gov
Subject: [lwc-forum] OFFICIAL COMMENT: Quartet

Dear All,

It seems the latest version of Quartet is vulnerable to forgery, with collisions between incomplete padded PT and PT with ending with data similar to the padding.

Exemple:

Key = 000102030405060708090A0B0C0D0E0F

Nonce = 2A2B2C2D2E2F303132333435

PT = **00000101020203030404050506060707000000000000000000000000000000**

AD =

CT = **E2C93E7C9A3CDBEAA1B45AAF536BCF87EB6D9DE52A5D6A0BA8B1D126561D89FA39D85B8061514487BCD37BF1D4F916**

Key = 000102030405060708090A0B0C0D0E0F

Nonce = 2A2B2C2D2E2F303132333435

PT = **00000101020203030404050506060707000000000000000000000000000000**01

AD =

CT = **E2C93E7C9A3CDBEAA1B45AAF536BCF87EB6D9DE52A5D6A0BA8B1D126561D89**0A**FA39D85B8061514487BCD37BF1D4F916**

- Same with non empty associated Data

Key = 000102030405060708090A0B0C0D0E0F

Nonce = 2A2B2C2D2E2F303132333435

PT = **00000101020203030404050506060707000000000000000000000000000000**

AD = 00000101

CT = **12CA42E90A3A39124BE1F48D933E349411A389773D61C071F9788CB5B4A84DA0AFBF57A59EF1CA056EC9AAC22464CB**

Key = 000102030405060708090A0B0C0D0E0F

Nonce = 2A2B2C2D2E2F303132333435

PT = **00000101020203030404050506060707000000000000000000000000000000**01

AD = 00000101

CT = **12CA42E90A3A39124BE1F48D933E349411A389773D61C071F9788CB5B4A84D**95**A0AFBF57A59EF1CA056EC9AAC22464CB**

Best regards,

Alexandre Mège

--

From: Zhang Bin <martin_zhangbin@hotmail.com>
Sent: Thursday, June 13, 2019 10:28 PM
To: MEGE, Alexandre; lightweight-crypto
Cc: lwc-forum@list.nist.gov
Subject: 回复: OFFICIAL COMMENT: Quartet

Dear Alexandre,

Thanks for the interest on Quartet. There is only one version of Quartet so far. Since the state update of Quartet is invertible, it seems that the posted observation is possible only when the IV/nonce is re-used with the same key so that the same intermediate internal states just before absorbing the message will be reached. However, Quartet works in the single key nonce-respecting setting, as already stated in the introduction and Chapter 2 of the document, i.e., it is written that "The cipher does not promise any integrity or confidentiality if the legitimate key holder uses the same nonce (IV) to encrypt two different (plaintext, associated data) pairs under the same key".

Besides, in the next paragraph of the same chapter 2, it is written that "If some padding scheme is needed, Quartet adopts the same padding scheme as that in [13]. Precisely, if the plaintext block size is mr bits, the padding scheme just appends a single 1 and the smallest number of 0s to the plaintext M such that the length of the padded plaintext is a multiple of mr bits. Thus the resulting padded plaintext is split into blocks of mr bits,

pad_mr(M) = M|| 1|| 0^(mr−1−(|M| mod mr)) if |M| > 0. "

That is, if the length of M is larger than 0, there will always be the padding whatever the input has a length of multiple of the unit mr or not.

Thus, due to the fact that there is a parameter mr here that cannot be determined in advance (which may be dependent on some concrete application), the reference code is free of padding and is supposed to process the already padded messages, not the incomplete padded message**.** In the posted example, the length of M is 31 bytes and M' is 32 bytes, if we choose mr to be 64, then after padding, the first message is of length 32=4*8 bytes, while the second one has a length of 40=5*8 bytes ; if mr is choose to be 32, then after padding, the first message is of length 32=4*8 bytes, and the second one is of length 36=4*9 bytes, which will have different tags accordingly in each case. There may be other possible values of mr in practical applications as well.

Thus, there is no change of Quartet's specification and the revised reference code.

Best Regards,

Bin Zhang

发件人: MEGE, Alexandre <alexandre.mege@airbus.com>
发送时间: 2019年6月11日 0:32
收件人: lightweight-crypto@nist.gov
抄送: lwc-forum@list.nist.gov
主题: [lwc-forum] OFFICIAL COMMENT: Quartet

From: Leo Perrin <perrin.leo@gmail.com>
Sent: Friday, July 5, 2019 10:25 AM
To: lwc-forum
Subject: [lwc-forum] OFFICIAL COMMENT: Quartet
Attachments: Quartet_forgery.zip

Dear All,

We have found a practical differential forgery attack against Quartet. It is a consequence of the existence of many high probability differential trails linking the rate before and after the call to the permutation R. For instance, if there is a difference of 2^6 on the first 64-bit block of the message and a difference of 2^26 + 2^25 + 2^20 on the second block of the message, then the difference in the internal state will cancel out with probability 2^{-6} and will yield a forgery with probability 2^{-9}.

We have experimentally verified our findings (the code is attached).

We reached out to the author before posting our result here, his answer follows.

"""Thanks a lot again for the analysis. Quartet is a design exploration of lightweight authenticated cipher based on a stream cipher that works with non-binary data unit. Unlike the bit-based authenticated cipher, which have several linear, non-linear feedbacks and non-linear output functions, it is tried to absorb and encrypt the message in a natural fast way and with a relatively small number of initialization rounds.

However, the distinction between a stream cipher and an authenticated cipher is overlooked here. That is, it is originally supposed to have the immunity against similar tag forgery attacks by the nonce respecting restriction so that the same internal state just before absorbing the message could not be reached. However, this restriction could be bypassed by only modifying the ciphertext and leaving the tag unchanged in the authenticated cipher scenario, which is an overlooked and different cryptanalysis setting from a stream cipher. The attacker just analyzes the differential properties of the involved permutation and then submits a number of such modified (ciphertext', tag) to the decryption and verification oracle, with some probability he will succeed.

I think this attack is interesting and is valid. If I am not wrong, this overlook could be remedied by injecting the same message multiple times into different and carefully selected parts of the internal state and assure a differential propagation probability below the security bound. Besides, limit the number of tag verification failures seems also be helpful in practice to frustrate such attacks. Maybe in a revision of the design, these two directions will be involved.
"""

Best regards,

André, Anne, Baptiste, Daniel, Ferdinand, Gaëtan, Henri, Léo, Marine, Patrick, Paul, Victor, and Virginie
--