From: Ashwin Jha <letterstoashwin@gmail.com>
Sent: Friday, June 14, 2019 5:45 AM
To: lightweight-crypto
Cc: lwc-forum@list.nist.gov; bhattacharjeearghya29@gmail.com; bishu.math.ynwa@gmail.com; Nilanjan Datta; Mridul Nandi; Ashwin Jha
Subject: OFFICIAL COMMENT: Qameleon

Dear all,

We have found a trivial forgery against all the general purpose variants of Qameleon, namely qameleon12812864gpv1, qameleon12812896gpv1 (primary candidate), and qameleon128128128tcgpv1.

The attack (demonstrated below) exploits the improper tweak setting for tag generation block cipher call.

Forgery Attack:
1. Query (N,A,M_1||M_1) to the encryption oracle. Let (C_1||C_2,T) be the ciphertext and tag pair.
2. Forge with (N,A,\epsilon,T), where \epsilon denotes empty ciphertext.

First, the checksum of M := M_1||M_1 matches with the checksum for empty message, i.e. 0; second, the tweak value for tag generation block cipher call is same in both the cases, i.e. , 4||v||0 (since nonce is same and |M|/128 < 2^28); and lastly, AD is same in both the cases. Thus, the forgery succeeds with probability 1.

In fact, the attack works for any message M = M_1||...||M_m such that
M_1 \xor ... \xor pad(M_m) = 0 and m < 2^28. We have verified our attack using the reference implementation of Qameleon.

--
Regards,
Arghya Bhattacharjee, Bishwajit Chakraborty, Nilanjan Datta, Ashwin Jha, Mridul Nandi

From: Roberto Avanzi <roberto.avanzi@gmail.com>
Sent: Friday, June 14, 2019 8:43 AM
To: Ashwin Jha
Cc: lightweight-crypto; lwc-forum; bhattacharjeearghya29@gmail.com; bishu.math.ynwa@gmail.com; Nilanjan Datta; Mridul Nandi; Ashwin Jha
Subject: Re: [lwc-forum] OFFICIAL COMMENT: Qameleon
Attachments: signature.asc

Thank you very much. This seems tone a silly oversight on our side, if correct.
Sunday I will meet with another designer and we will fix it. Also, the web site is going up soon.

Roberto

> On 14 Jun 2019, at 11:44, Ashwin Jha <letterstoashwin@gmail.com> wrote:
>
> Dear all,
>
> We have found a trivial forgery against all the general purpose
> variants of Qameleon, namely qameleon12812864gpv1,
> qameleon12812896gpv1 (primary candidate), and qameleon128128128tcgpv1.
>
> The attack (demonstrated below) exploits the improper tweak setting
> for tag generation block cipher call.
>
> Forgery Attack:
> 1. Query (N,A,M_1||M_1) to the encryption oracle. Let (C_1||C_2,T) be
> the ciphertext and tag pair.
> 2. Forge with (N,A,\epsilon,T), where \epsilon denotes empty ciphertext.
>
> First, the checksum of M := M_1||M_1 matches with the checksum for
> empty message, i.e. 0; second, the tweak value for tag generation
> block cipher call is same in both the cases, i.e. , 4||v||0 (since
> nonce is same and |M|/128 < 2^28); and lastly, AD is same in both the
> cases. Thus, the forgery succeeds with probability 1.
>
> In fact, the attack works for any message M = M_1||...||M_m such that
> M_1 \xor ... \xor pad(M_m) = 0 and m < 2^28. We have verified our
> attack using the reference implementation of Qameleon.
>
> --
> Regards,
> Arghya Bhattacharjee, Bishwajit Chakraborty, Nilanjan Datta, Ashwin
> Jha, Mridul Nandi
>
> --
> To unsubscribe from this group, send email to
> lwc-forum+unsubscribe@list.nist.gov

From: Roberto Avanzi <roberto.avanzi@gmail.com>
Sent: Tuesday, June 18, 2019 2:02 PM
To: Ashwin Jha
Cc: lightweight-crypto; lwc-forum; bhattacharjeearghya29@gmail.com; bishu.math.ynwa@gmail.com; Nilanjan Datta; Mridul Nandi
Subject: Re: [lwc-forum] OFFICIAL COMMENT: Qameleon
Attachments: signature.asc

Thank you again for your remark. As I said before it was a silly oversight on our side. But there also a bit more to it.

Take a message M = M_1 || M_2 || M_3 where M_1 has full block length (no fractional part), the checksum of M_2 is zero, the length of M_2 is a multiple of 2^24 blocks, and M_3 is at least one byte. Take any AD A.

Now ask for the encryption of message M with AD A. You get a nonce N and tag, say, T. At this point you send (N, A, M_1 || M_3, T) instead. The checksum is the same, the tag computation is the same.

Note that 2^24 is because we are putting the truncated byte length of the message in the tweak for the last block. The last 4 or 3 bits (depending on whether the message blocks are 16 or 8 bytes, respectively) are sufficient, because of tweak domain separation in the upper bits. The attack now becomes even more flexible.

Why did this happen?

It appears that we have made a very unfortunate slip in instantiating \ThetaCB. In this mode, the tweak in the tag computation should contain the length -- indeed all OCB / \ThetaCB security proofs depend on this and it does not make sense otherwise. In a revision of the spec, the length will be included. It is really not much more than a typo that stuck. Similarly for the use of \nu instead of N in the tag computation.

Finally, in the tweak formation for the tag computation, the current rotated nonce N should be used and not \nu. Also a typo, and then copy-and-paste.

An important remark:

We note that our previous (original) specification may still be considered secure if we assume that the length (which has to be transmitted separately) is transmitted in a secure way. Protecting the length is delegated to the protocol. Note that we do transmit the length additionally in the reference implementation - without protection because it is just a reference implementation of the cipher.

Finally, initially empty messages were not supported, but then the NIST committee contacted us saying that the implementation did not pass the tests for empty messages. Instead of excluding them we made a last minute change to support them. Never do last minute changes! We should just have excluded them from the vector test generation. With hindsight this was good since it allowed you to find that forgery and then us later to generalise it.

The cleanest way to support everything while keeping the mode as simple as possible and with less conditional parts as possible is then

- Include the \ell-1 in the tweak computation - fixing our slip above. Similar for \nu -> N

- Always pad AD and message. This makes the mapping of messages to padded messages prefix-free, removing the need to conditionally domain separate the last tweak or not. This also simplifies the code.

BONUS: Because of the prefix free mapping, the plaintext length can now be recovered upon decryption without having to send the length with it.

We will soon post the update at the web site https://qameleon.site

Arghya, Bishwajit, Nilanjan, Ashwin, and Mridul, thank you again for spotting this early!

Roberto

> On 14 Jun 2019, at 12:44, Ashwin Jha <letterstoashwin@gmail.com> wrote:
>
> Dear all,
>
> We have found a trivial forgery against all the general purpose
> variants of Qameleon, namely qameleon12812864gpv1,
> qameleon12812896gpv1 (primary candidate), and qameleon128128128tcgpv1.
>
> The attack (demonstrated below) exploits the improper tweak setting
> for tag generation block cipher call.
>
> Forgery Attack:
> 1. Query (N,A,M_1||M_1) to the encryption oracle. Let (C_1||C_2,T) be
> the ciphertext and tag pair.
> 2. Forge with (N,A,\epsilon,T), where \epsilon denotes empty ciphertext.
>
> First, the checksum of M := M_1||M_1 matches with the checksum for
> empty message, i.e. 0; second, the tweak value for tag generation
> block cipher call is same in both the cases, i.e. , 4||v||0 (since
> nonce is same and |M|/128 < 2^28); and lastly, AD is same in both the
> cases. Thus, the forgery succeeds with probability 1.
>
> In fact, the attack works for any message M = M_1||...||M_m such that
> M_1 \xor ... \xor pad(M_m) = 0 and m < 2^28. We have verified our
> attack using the reference implementation of Qameleon.
>
> --
> Regards,
> Arghya Bhattacharjee, Bishwajit Chakraborty, Nilanjan Datta, Ashwin
> Jha, Mridul Nandi
>
> --
> 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
> ---
> To unsubscribe from this group and stop receiving emails from it, send an email to lwc-forum+unsubscribe@list.nist.gov.
>

From: Roberto Avanzi <roberto.avanzi@gmail.com>
Sent: Wednesday, June 19, 2019 4:56 AM
To: lightweight-crypto
Cc: lwc-forum@list.nist.gov
Subject: OFFICIAL COMMENT: Qameleon
Attachments: signature.asc

Regarding the forgery found by Arghya Bhattacharjee, Bishwajit Chakraborty, Nilanjan Datta, Ashwin Jha, and Mridul Nandi: Thank you again for your remark. As I said before it was a silly oversight on our side. But there also a bit more to it.

Take a message M = M_1 || M_2 || M_3 where M_1 has full block length (no fractional part), the checksum of M_2 is zero, the length of M_2 is a multiple of 2^24 blocks, and M_3 is at least one byte. Take any AD A.

Now ask for the encryption of message M with AD A. You get a nonce N and tag, say, T. At this point you send (N, A, M_1 || M_3, T) instead. The checksum is the same, the tag computation is the same.

Note that 2^24 is because we are putting the truncated byte length of the message in the tweak for the last block. The last 4 or 3 bits (depending on whether the message blocks are 16 or 8 bytes, respectively) are sufficient, because of tweak domain separation in the upper bits. The attack now becomes even more flexible.

How could this happen?

It appears that we have made a very unfortunate slip in instantiating \ThetaCB. In this mode, the tweak in the tag computation should contain the length -- indeed all OCB / \ThetaCB security proofs depend on this and it does not make sense otherwise. In a revision of the spec, the length will be included. It is really not much more than a typo that stuck.

Similarly, in the tweak formation for the tag computation, the current rotated nonce N should be used and not \nu. This value depends on the length, it should be there. This is also a slip, and then it was copy-and-paste.

An important remark:

We note that our previous (original) specification may still be considered secure if we assume that the length (which has to be transmitted separately) is transmitted in a secure way. Protecting the length is delegated to the protocol. Note that we do transmit the length additionally in the reference implementation - without protection because it is just a reference implementation of the cipher.

Finally, initially empty messages were not supported, but then the NIST committee contacted us saying that the implementation did not pass the tests for empty messages. Instead of excluding them we made a last minute change to support them. Never do last minute changes! We should just have excluded them from the vector test generation. With hindsight this was good since it allowed you to find that forgery and then us later to generalise it.

The cleanest way to support everything while keeping the mode as simple as possible and with less conditional parts as possible is then

- Include the \ell-1 in the tweak computation - fixing our slip above. Similar for \nu -> N

- Always pad AD and message. This makes the mapping of messages to padded messages prefix-free, removing the need to conditionally domain separate the last tweak or not. This also simplifies the code.

- BONUS: because of the prefix free mapping, the plaintext length can now be recovered upon decryption without having to send the length with it.

We will discuss this internally a bit more and then post the update at the web site https://qameleon.site