From: Danilo Gligoroski <danilog@ntnu.no>
Sent: Thursday, December 28, 2017 10:40 PM
To: pqc-comments
Cc: pqc-forum@list.nist.gov
Subject: [pqc-forum] OFFICIAL COMMENT: BIKE

Dear BIKE-2 designers,

The KAT files for BIKE-2 have wrong or somewhat misleading format.

While in the documentation you state that the public key is r bits long, KAT files has public keys with n bits long where the systematic part is unnecessarily printed out.

The same situation is with the ciphertexts that instead of being r bits long, they are n bits long, with half of the bits being zeros.

I could not find any remark in the documentation or in the README files that clarifies your decision to produce KAT files with unnecessary and redundant outputs.

I see this as a technical programming mistake, and should be fixed.

Regards,
Danilo!

From: Misoczki, Rafael <rafael.misoczki@intel.com>
Sent: Friday, December 29, 2017 4:13 PM
To: Danilo Gligoroski; pqc-comments
Cc: pqc-forum@list.nist.gov; Misoczki, Rafael
Subject: RE: [pqc-forum] OFFICIAL COMMENT: BIKE

Dear Danilo,

Thank you for your comment.

BIKE-2 has public keys in systematic form, thus only r bits need to be communicated. BIKE-1 and BIKE-3 do not have public keys in systematic form, thus require n=2r bits.

The reference code implements all BIKE-1, BIKE-2 and BIKE-3 variants, and we wanted to keep the same format for the KAT files, to simplify things. Thus, we made BIKE-2 to be n bits long as well, although the first r bits can obviously be omitted, if/when the protocol is used.

Please note that the additional implementation of BIKE-2 does output only r bits.

Best Regards,
BIKE team

-----Original Message-----
From: Danilo Gligoroski [mailto:danilog@ntnu.no]
Sent: Friday, December 29, 2017 1:40 AM
To: pqc-comments@nist.gov
Cc: pqc-forum@list.nist.gov
Subject: [pqc-forum] OFFICIAL COMMENT: BIKE

Dear BIKE-2 designers,

The KAT files for BIKE-2 have wrong or somewhat misleading format.

While in the documentation you state that the public key is r bits long, KAT files has public keys with n bits long where the systematic part is unnecessarily printed out.

The same situation is with the ciphertexts that instead of being r bits long, they are n bits long, with half of the bits being zeros.

I could not find any remark in the documentation or in the README files that clarifies your decision to produce KAT files with unnecessary and redundant outputs.

I see this as a technical programming mistake, and should be fixed.

Regards,
Danilo!

From: Danilo Gligoroski <danilog@ntnu.no>
Sent: Friday, January 05, 2018 2:23 AM
To: pqc-comments
Cc: pqc-forum@list.nist.gov
Subject: OFFICIAL COMMENT: BIKE

Topic: Simple ciphertext distinguisher for BIKE-1 and BIKE-2

Dear BIKE-1 and BIKE-2 designers,

If we look at your supplied KAT files for BIKE-2 Level 1 and Level 3 we can notice the following pattern:
For BIKE-2, Level 1, the Hamming weights for the first 16 ciphertexts are:
{5112, 5062, 5082, 5088, 5152, 5062, 5028, 5046, 5022, 5056, 5076, 5096, 5100, 5114, 5064, 5196}.
As we can see, all weights are even.

For BIKE-2, Level 3, the Hamming weights for the first 16 ciphertexts are:
{9975, 9989, 9903, 9915, 9869, 9891, 9927, 10075, 9923, 9895, 10031, 9989, 9843, 10059, 9921}.
As we can see, all weights are odd.

I transformed this observation into the following simple ciphertext distinguisher for BIKE-1 and BIKE-2.

For BIKE-1, the public key is (f0, f1). Let the received ciphertext is c = (c0, c1).
Compute: c2 = c0 f0^{-1} + c1 f1^{-1}

For BIKE-1 Level 1 and Level 5:
If HammingWeight(c2) is odd then
  claim that c was not produced by the public key (f0, f1) else
  make a guess with probability 1/2 whether the c was produced by the public key (f0, f1)

Success probability of winning the guess game is 0.75

For BIKE-1 Level 3
If HammingWeight(c2) is even then
  claim that c was not produced by the public key (f0, f1) else
  make a guess with probability 1/2 whether the c was produced by the public key (f0, f1)

Success probability of winning the guess game is 0.75

For BIKE-2, the public key is h. Let the received ciphertext is c.

For BIKE-2 Level 1 and Level 5:
If HammingWeight(c) is odd then

claim that c was not produced by the public key h else
make a guess with probability 1/2 whether the c was produced by the public key h

Success probability of winning the guess game is 0.75

For BIKE-2 Level 3
If HammingWeight(c) is even then
claim that c was not produced by the public key h else
make a guess with probability 1/2 whether the c was produced by the public key h

Success probability of winning the guess game is 0.75

Best regards,
Danilo!

From: Mike Hamburg <mike@shiftleft.org>
Sent: Friday, January 05, 2018 3:16 AM
To: Danilo Gligoroski
Cc: pqc-comments; pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] OFFICIAL COMMENT: BIKE

Hi Danilo,

Is “ciphertexts indistinguishable from random” one of the NIST desiderata? If so, I must have missed it.

Or do you mean that this property suggests some other weakness?

Thanks,
— Mike

> On Jan 4, 2018, at 11:22 PM, Danilo Gligoroski <danilog@ntnu.no> wrote:
>
> Topic: Simple ciphertext distinguisher for BIKE-1 and BIKE-2
>
>
> Dear BIKE-1 and BIKE-2 designers,
>
>
> If we look at your supplied KAT files for BIKE-2 Level 1 and Level 3 we can notice the following pattern:
> For BIKE-2, Level 1, the Hamming weights for the first 16 ciphertexts are:
> {5112, 5062, 5082, 5088, 5152, 5062, 5028, 5046, 5022, 5056, 5076, 5096, 5100, 5114, 5064, 5196}.
> As we can see, all weights are even.
>
> For BIKE-2, Level 3, the Hamming weights for the first 16 ciphertexts are:
> {9975, 9989, 9903, 9915, 9869, 9891, 9927, 10075, 9923, 9895, 10031, 9989, 9843, 10059, 9921}.
> As we can see, all weights are odd.
>
>
> I transformed this observation into the following simple ciphertext distinguisher for BIKE-1 and BIKE-2.
>
>
> For BIKE-1, the public key is (f0, f1). Let the received ciphertext is c = (c0, c1).
> Compute: c2 = c0 f0^{-1} + c1 f1^{-1}
>
> For BIKE-1 Level 1 and Level 5:
> If HammingWeight(c2) is odd then
> claim that c was not produced by the public key (f0, f1)
> else
> make a guess with probability 1/2 whether the c was produced by the public key (f0, f1)
>
> Success probability of winning the guess game is 0.75
>
> For BIKE-1 Level 3

From: Danilo Gligoroski <danilog@ntnu.no>
Sent: Friday, January 05, 2018 6:30 AM
To: Mike Hamburg
Cc: pqc-comments; pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] OFFICIAL COMMENT: BIKE

Hi Mike,

When I sent my observation about the ciphertext distinguisher for BIKE-1 and BIKE-2 I was motivated by this part of the NIST call:
"NIST will perform a thorough analysis of the submitted algorithms in a manner that is open and transparent to the public, as well as encourage the cryptographic community to also conduct analyses and evaluation. This combined analysis will inform NIST's decision on the subsequent development of post-quantum standards."

I understand that sentence from the NIST call as "if you notice something, say something" (about some proposed cipher :-)). For the concrete ciphers BIKE-1 and BIKE-2, I don't know will it lead to some other weakness, but our crypto community will know it as one of the properties of BIKE-1 and BIKE-2.

Best regards,
Danilo!

On 05/01/2018 09:15, Mike Hamburg wrote:
> Hi Danilo,
>
> Is "ciphertexts indistinguishable from random" one of the NIST desiderata? If so, I must have missed it.
>
> Or do you mean that this property suggests some other weakness?
>
> Thanks,
> — Mike
>
>> On Jan 4, 2018, at 11:22 PM, Danilo Gligoroski <danilog@ntnu.no> wrote:
>>
>> Topic: Simple ciphertext distinguisher for BIKE-1 and BIKE-2
>>
>>
>> Dear BIKE-1 and BIKE-2 designers,
>>
>>
>> If we look at your supplied KAT files for BIKE-2 Level 1 and Level 3 we can notice the following pattern:
>> For BIKE-2, Level 1, the Hamming weights for the first 16 ciphertexts are:
>> {5112, 5062, 5082, 5088, 5152, 5062, 5028, 5046, 5022, 5056, 5076, 5096, 5100, 5114, 5064, 5196}.
>> As we can see, all weights are even.
>>
>> For BIKE-2, Level 3, the Hamming weights for the first 16 ciphertexts are:
>> {9975, 9989, 9903, 9915, 9869, 9891, 9927, 10075, 9923, 9895, 10031, 9989, 9843, 10059, 9921}.

From: Misoczki, Rafael <rafael.misoczki@intel.com>
Sent: Monday, January 08, 2018 12:22 PM
To: Danilo Gligoroski; Mike Hamburg
Cc: pqc-comments; pqc-forum@list.nist.gov; Misoczki, Rafael
Subject: RE: [pqc-forum] OFFICIAL COMMENT: BIKE

Dear Danilo, dear all,

Thanks for your comments.
We already knew that the ciphertext parity is defined by the public parameters (please see below the simple analysis on why). We do not see how this would represent any sort of practical weakness to the proposed scheme.

Consider BIKE-1. The private key is composed by two sparse binary polynomials (h_0, h_1) of odd weight w/2 each. The public key is computed by multiplying the pair (h_0, h_1) by a dense polynomial g of odd weight as well (g has odd weight to ensure the public code is not a sub-code of the private one). Thus, pk has even weight (as it is composed by two parts of odd weight). Moreover, codewords in the public code (mgh_0, mgh_1) have even weight regardless of the message m. Finally, the ciphertext is a noisy codeword that encompasses an error vector of weight t. For parameter sets 1 and 5, t is odd, while for parameter set 3, t is even. While adding the errors, the parity of the original codeword switch from even to odd, from odd to even, a fixed number of times (exactly t times). Therefore, what you ended up observing is the fact that the parity of the ciphertext depends on t (a public parameter).

Similar observation applies to BIKE-2 and 3. We do not see this as a unexpected property, but instead as a straightforward consequence of multiplying and adding polynomials of odd/even weight.
One thing we discussed however is that we would like to explicitly mention this property in our security proof, in order to clarify any potential question in this regard. We plan to publish an updated proof in our site (https://na01.safelinks.protection.outlook.com/?url=www.bikesuite.org&data=02%7C01%7Csara.kerman%40nist.gov%7C4e8cf3b9350f4bd2041708d556bc9738%7C2ab5d82fd8fa4797a93e054655c61dec%7C1%7C0%7C636510290356375474&sdata=Suj2fFhocoJ72MKxw1IIRYvxJ3pg8QGfeap3kUfsiMw%3D&reserved=0) shortly. The site is currently under construction and should be up in the coming week.

Best Regards,
BIKE Team

-----Original Message-----
From: Danilo Gligoroski [mailto:danilog@ntnu.no]
Sent: Friday, January 5, 2018 9:30 AM
To: Mike Hamburg <mike@shiftleft.org>
Cc: pqc-comments@nist.gov; pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] OFFICIAL COMMENT: BIKE

Hi Mike,

When I sent my observation about the ciphertext distinguisher for BIKE-1 and BIKE-2 I was motivated by this part of the NIST call:
"NIST will perform a thorough analysis of the submitted algorithms in a manner that is open and transparent to the public, as well as encourage the cryptographic community to also conduct analyses and evaluation. This combined analysis will inform NIST's decision on the subsequent development of post-quantum standards."

From: Danilo Gligoroski <danilog@ntnu.no>
Sent: Monday, January 08, 2018 2:46 PM
To: Misoczki, Rafael; Mike Hamburg
Cc: pqc-comments; pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] OFFICIAL COMMENT: BIKE

Hi Rafael,

After my answer to Mike I went back to your documentation and saw this:
<quote>
Game G2: In this game, the simulator picks uniformly at random the public key, specifically (f0,f1) for BIKE-1 and BIKE-3, and h for BIKE-2. The rest of the game then proceeds honestly.
An adversary distinguishing between these two games is therefore able to distinguish between a well-formed public key and a randomly-generated one.
</quote>

So obviously something is not well said or explained. But I did not want to make a conversation with myself about BIKE, although it was obvious that the correctness of the proof has to be addressed. Good that you have answered and that you plan to clarify the issue in the proof.

Considering BIKE-3, I just skipped it since it is a patented one, and is out of my interest.

Cheers,
Danilo!

On 08/01/2018 18:22, Misoczki, Rafael wrote:
> Dear Danilo, dear all,
>
> Thanks for your comments.
> We already knew that the ciphertext parity is defined by the public parameters (please see below the simple analysis on why). We do not see how this would represent any sort of practical weakness to the proposed scheme.
>
> Consider BIKE-1. The private key is composed by two sparse binary polynomials (h_0, h_1) of odd weight w/2 each. The public key is computed by multiplying the pair (h_0, h_1) by a dense polynomial g of odd weight as well (g has odd weight to ensure the public code is not a sub-code of the private one). Thus, pk has even weight (as it is composed by two parts of odd weight). Moreover, codewords in the public code (mgh_0, mgh_1) have even weight regardless of the message m. Finally, the ciphertext is a noisy codeword that encompasses an error vector of weight t. For parameter sets 1 and 5, t is odd, while for parameter set 3, t is even. While adding the errors, the parity of the original codeword switch from even to odd, from odd to even, a fixed number of times (exactly t times). Therefore, what you ended up observing is the fact that the parity of the ciphertext depends on t (a public parameter).
>
> Similar observation applies to BIKE-2 and 3. We do not see this as a unexpected property, but instead as a straightforward consequence of multiplying and adding polynomials of odd/even weight.
> One thing we discussed however is that we would like to explicitly mention this property in our security proof, in order to clarify any potential question in this regard. We plan to publish an updated proof in our site (https://na01.safelinks.protection.outlook.com/?url=www.bikesuite.org&data=02%7C01%7Csara.kerman%40nist.gov%7

From: Perlner, Ray (Fed)
Sent: Thursday, February 15, 2018 5:20 PM
To: pqc-comments
Cc: pqc-forum@list.nist.gov
Subject: OFFICIAL COMMENT: BIKE

Dear BIKE team.

I have a few questions regarding the security proof given in http://bikesuite.org/files/BIKE.pdf .

Theorems 2 and 3 seem to claim that the security assumptions for BIKE-1 are unlike those of BIKE-2, but rather similar to the security assumptions for BIKE-3. Is this right? If so, that seems surprising, since a BIKE-2 key exchange can be turned by an adversary into a fairly generic BIKE-1 key exchange, with the same shared secret, simply by

1) Switching f0, f1 and multiplying them by a random polynomial (gh0) , resulting in f0' and f1' and,
2) Replacing the ciphertext c with (c0, c1) = (mf0' + c, mf1').

Likewise, as long as f1 is invertible, a BIKE-1 key exchange can be turned by an adversary into a generic BIKE-2 key exchange by

1) Replacing f0 and f1 by f1' = f1^-1 f0 and,
2) Replacing (c0, c1) by c = c1 f1' + c0.

Additionally, I am having trouble following the proof. On page 33 of the linked pdf, it is claimed that the public key of BIKE-1 is an instance of the (2-1) QCSD problem, defined on page 29, but I cannot see the connection. (I can, however, see how to turn it into a QCCF problem, assuming f1 is invertible, I think.) Am I missing something?

Thanks,
Ray Perlner

From: team@bikesuite.org
Sent: Saturday, March 10, 2018 1:52 PM
To: Perlner, Ray (Fed); pqc-comments
Cc: pqc-forum@list.nist.gov; team@bikesuite.org
Subject: RE: OFFICIAL COMMENT: BIKE

Dear Ray, dear all,

Thank you for your comments on BIKE. Indeed the proof had a few typos in sorting out the underlying security problems for BIKE variants.

We have updated the proof to a more formal and detailed format, and it is now available on our website http://bikesuite.org For backtracking the changes, we are still keeping the old version of the document on our website.

We stress that these changes do not impact the practical security of the scheme, the parameters, nor the algorithms/spec of our schemes.

Please let us know if you have additional comments/questions.

Best regards,
BIKE Team

--

From: Perlner, Ray (Fed) [mailto:ray.perlner@nist.gov]
Sent: Thursday, February 15, 2018 2:20 PM
To: pqc-comments <pqc-comments@nist.gov>
Cc: pqc-forum@list.nist.gov
Subject: [pqc-forum] OFFICIAL COMMENT: BIKE

Dear BIKE team.

I have a few questions regarding the security proof given in http://bikesuite.org/files/BIKE.pdf .

Theorems 2 and 3 seem to claim that the security assumptions for
BIKE-1 are unlike those of BIKE-2, but rather similar to the security assumptions for BIKE-3. Is this right? If so, that seems surprising, since a BIKE-2 key exchange can be turned by an adversary into a fairly generic BIKE-1 key exchange, with the same shared secret, simply by

1) Switching f0, f1 and multiplying them by a random polynomial (gh0)
, resulting in f0' and f1' and,
2) Replacing the ciphertext c with (c0, c1) = (mf0' + c, mf1').

Likewise, as long as f1 is invertible, a BIKE-1 key exchange can be turned by an adversary into a generic BIKE-2 key exchange by

From: Perlner, Ray (Fed)
Sent: Wednesday, July 25, 2018 2:14 PM
To: pqc-comments
Cc: pqc-forum@list.nist.gov
Subject: OFFICIAL COMMENT: BIKE

Dear BIKE team, dear all.

I have some comments regarding reaction attacks and misuse resilience.

The BIKE specification recommends protecting against reaction attacks by changing the public key for every new key exchange. However, since the only reaction attack mentioned is the GJS attack, which requires something on the order of a trillion messages on average, one might worry that implementers may be tempted to relax the key reuse restrictions. It is worth noting that (aside from the proscription against key reuse) there is a better reaction attack against the BIKE scheme as specified, since it will report a decryption failure whenever given a ciphertext created with an error vector whose weight is not equal to t. For example, an attacker can recover the error vector from a recorded BIKE 1 key exchange with approximately n decryption failure queries against the same public key. The attacker does this by using the fact that a ciphertext differing from the target ciphertext at 2 bit positions will only successfully decrypt if exactly one of those positions is the position of a nonzero bit in the target error vector. I believe similar attacks can be mounted against the other BIKE variants.

Obviously this does not contradict the claimed security properties (IND-CPA) of BIKE, but it may be worth leaving as a question to the forum whether it might be a good idea to explicitly prevent this sort of reaction attack in the name of misuse resilience.

Cheers,
Ray Perlner

From: Misoczki, Rafael <rafael.misoczki@intel.com>
Sent: Saturday, August 04, 2018 2:11 PM
To: Perlner, Ray (Fed); pqc-comments
Cc: pqc-forum@list.nist.gov; team@bikesuite.org
Subject: RE: OFFICIAL COMMENT: BIKE

Dear Ray,

Thank you for your comments.

A correct BIKE deployment must use ephemeral keys as clearly stated in our proposal. This has two major benefits:

1-Forward secrecy
2-Completely defeat reaction attacks like GJS and the one you described

Nevertheless, for the sake of completeness, we plan to add to our document a description of the rationale you described. We think this is informative and should indeed discourage users to relax the key-reuse restriction, as you insightfully mentioned.

Best Regards,
BIKE Team

From: Perlner, Ray (Fed) [mailto:ray.perlner@nist.gov]
Sent: Wednesday, July 25, 2018 11:14 AM
To: pqc-comments <pqc-comments@nist.gov>
Cc: pqc-forum@list.nist.gov
Subject: [pqc-forum] OFFICIAL COMMENT: BIKE

Dear BIKE team, dear all.

I have some comments regarding reaction attacks and misuse resilience.

The BIKE specification recommends protecting against reaction attacks by changing the public key for every new key exchange. However, since the only reaction attack mentioned is the GJS attack, which requires something on the order of a trillion messages on average, one might worry that implementers may be tempted to relax the key reuse restrictions. It is worth noting that (aside from the proscription against key reuse) there is a better reaction attack against the BIKE scheme as specified, since it will report a decryption failure whenever given a ciphertext created with an error vector whose weight is not equal to t. For example, an attacker can recover the error vector from a recorded BIKE 1 key exchange with approximately n decryption failure queries against the same public key. The attacker does this by using the fact that a ciphertext differing from the target ciphertext at 2 bit positions will only successfully decrypt if exactly one of those positions is the position of a nonzero bit in the target error vector. I believe similar attacks can be mounted against the other BIKE variants.

Obviously this does not contradict the claimed security properties (IND-CPA) of BIKE, but it may be worth leaving as a question to the forum whether it might be a good idea to explicitly prevent this sort of reaction attack in the name of misuse resilience.

Cheers,
Ray Perlner