From: Matthieu Lequesne <matthieu.lequesne@inria.fr>
Sent: Friday, January 05, 2018 1:40 PM
To: pqc-comments
Cc: pqc-forum@list.nist.gov
Subject: OFFICIAL COMMENT: Edon-K

Dear Designers of EdonK, Dear all,

We looked at the EdonK KEM and found an attack.
Given the public key and a cipher text, we manage to recover the shared secret.

The ciphertext is a vector C such that C = M . PubMat + error.
The error is of rank 6 by construction.
The parity check matrix PrivMat has coefficients in the sub-vector space of F_q generated by elements a and b.
Hence, the code whose parity-check matrix is PrivMat is an LRPC code of rank 2 as defined in [1, Definition 3].
This code is a super-code of the code generated by PubMat used in the scheme because of Corollary 1 (page 19).

We manage to reconstruct this LRPC code from the public information.
This LRPC code can correct more than 6 errors and we can therefore use it to decode the ciphertext and recover the shared secret.

Here is how we proceed.

The attacker does not have access to the value of a and b but to the
value of a/b=c/d as mentioned in paragraph 7.2.2 of the documentation.
Let us denote \alpha = a/b.
We know that b^{-1}.PrivMat is also a parity-check matrix of the LRPC
code and has its coefficients in the two dimensional sub-vector space
generated by 1 and \alpha.
Hence we know that this code admits a parity-check matrix with
coefficients in <1,\alpha>.
We use this information to reconstruct such a parity-check matrix of the
LRPC code by solving a linear system as in [2, section IV B].

Now we have a parity-check matrix whose entries are all in a
2-dimentional vector space.
We can use the general decoding of LRPC codes.
According to theorem 1 of [1]:
Let H be a (n-k)*n dual matrix of a LRPC code with low rank d>=2 over F_q^m,
then the algorithm 1 decodes a random error e of low rank r such that
r*d <= n-k
with failure probability q^{-(n-k+1-rd)} and complexity r^2*(4d^2*m+n^2).

Here, with the parameters of edonk128ref, we have r=6, d=2, q=2, n=144,
k=104, m=128.
So r*d=12 <= 40=n-k, so we can decode with error probability 2^-29 and
complexity < 2^20.

Then we recover the error. The vector span directly gives a list of candidates.
We recover the shared secret as in step 6 of the decapsulation process.

We intend to write a more detailed document in a few days.

Please tell us if we missed something.
Best regards,

Matthieu Lequesne, Nicolas Sendrier and Jean-Pierre Tillich.

--

[1] Gaborit, P., Murat, G., Ruatta, O., & Zémor, G. (2013, April). Low rank parity check codes and their application to cryptography. In Proc. WCC (pp. 168-180).
[2] Gaborit, Philippe, Olivier Ruatta, and Julien Schrek. "On the complexity of the rank syndrome decoding problem." IEEE Transactions on Information Theory 62.2 (2016): 1006-1019.

From: Danilo Gligoroski <danilog@ntnu.no>
Sent: Friday, January 05, 2018 4:45 PM
To: Matthieu Lequesne; pqc-comments
Cc: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] OFFICIAL COMMENT: Edon-K

Dear Matthieu, Nicolas and Jean-Pierre,

Thank you very much for your interest in Edon-K.

If I understand your attack, basically, what you claim is that you have found a second preimage attack on SHA256 with a complexity 2^129 which is highly unlikely.

Explanation:
Your attack is trying to recover one of many parity check matrices, but with two elements: 1 and a/b. I think that part is ok. Then you refer to the decapsulation procedure to recover the shared secret and hope that Step 6 will be satisfied. Note that the values in Step 6 in the documentation are obtained from Step 4 where the knowledge of both a and b is necessary. Your attack does not recover neither a nor b. What your attack is doing is a production of an alternate vector span, with values multiplied by a/b. Then, by searching trough pairs of values (s_\mu, s_\nu) of your vector span (of up to 2^40 elements), with a probability 2^-29 and complexity <2^20, you hope that you will hit the supplied SHA256 hash value h. So a total cost of your attack is (2^40 x 2^40) x 2^20 / 2^-29 = 2^129 and you have found a second preimage for SHA256. I think that is highly unlikely. Here lies the mistake in your attack: The Step 6 from your alternate vector span will produce the same hash value h with a very low probability <2^-256.

Best regards,
Danilo!

On 05/01/2018 19:40, Matthieu Lequesne wrote:
> Dear Designers of EdonK, Dear all,
>
> We looked at the EdonK KEM and found an attack.
> Given the public key and a cipher text, we manage to recover the shared secret.
>
> The ciphertext is a vector C such that C = M . PubMat + error.
> The error is of rank 6 by construction.
> The parity check matrix PrivMat has coefficients in the sub-vector space of F_q generated by elements a and b.
> Hence, the code whose parity-check matrix is PrivMat is an LRPC code of rank 2 as defined in [1, Definition 3].
> This code is a super-code of the code generated by PubMat used in the scheme because of Corollary 1 (page 19).
>
> We manage to reconstruct this LRPC code from the public information.
> This LRPC code can correct more than 6 errors and we can therefore use it to decode the ciphertext and recover the shared secret.

From: Matthieu Lequesne <matthieu.lequesne@inria.fr>
Sent: Tuesday, February 20, 2018 10:30 AM
To: pqc-comments
Cc: pqc-forum@list.nist.gov; tillich
Subject: OFFICIAL COMMENT: Edon-K
Attachments: edonk-attack.sage; PQCkemKAT.rsp

Dear all,

We mentioned the existence of an attack on the Edon-K KEM on January 5th.
A detailed description of our attack is now available on arxiv:
https://arxiv.org/abs/1802.06157

We also implemented the attack. Our script reads the public key and ciphertext from the KAT file and successfully recovers the secret within a minute.
You will find the Sage script attached. It is designed to work on the reference version (named "edonk128ref"). It reads its input from the file "PQCkemKAT.rst" (placed in the same directory) and successfully recovers the shared secret for all examples. Just run "sage edonk-attack.sage" to try it on the first example of the KAT file.

We would like to thank Danilo Gligoroski who answered all our questions about his scheme.

Best regards,

Matthieu Lequesne and Jean-Pierre Tillich

From: Danilo Gligoroski <danilog@ntnu.no>
Sent: Tuesday, February 20, 2018 12:33 PM
To: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] OFFICIAL COMMENT: Edon-K

Hi,

I want to congratulate Matthieu and Jean-Pierre for their excellent job.

Although there are several other ways how to mitigate the attack on Edono-K that are not discussed in the paper (such as increasing the rank of the matrix H), I think that in this moment it would be better to focus our attention to other good submissions that have not yet received much attention.

I appreciate also NIST offer to describe the current scheme at the upcoming workshop, even it has been broken. NIST, so far is running the PQC standardization process with a lot of authority, credibility, and a fair treatment to all submitters.

I withdraw Edon-K from the standardization process.

Best regards,
Danilo!

On 20/02/2018 16:30, Matthieu Lequesne wrote:
> Dear all,
>
> We mentioned the existence of an attack on the Edon-K KEM on January 5th.
> A detailed description of our attack is now available on arxiv:
https://arxiv.org/abs/1802.06157
>

> We also implemented the attack. Our script reads the public key and ciphertext from the KAT file and successfully recovers the secret within a minute.
> You will find the Sage script attached. It is designed to work on the reference version (named "edonk128ref"). It reads its input from the file "PQCkemKAT.rst" (placed in the same directory) and successfully recovers the shared secret for all examples. Just run "sage edonk-attack.sage" to try it on the first example of the KAT file.
>
> We would like to thank Danilo Gligoroski who answered all our questions about his scheme.
>
> Best regards,
>
> Matthieu Lequesne and Jean-Pierre Tillich
>