**From:** Smith-Tone, Daniel (Fed)
**Sent:** Tuesday, January 09, 2018 12:20 PM
**To:** pqc-comments
**Cc:** pqc-forum@list.nist.gov
**Subject:** DAGS-5 Entropy

Dear community,

I was asked to comment on DAGS since there seem to be no comments on this issue yet.

The parameters and performance data in the submission files for DAGS-5 are invalid because it is incapable of establishing shared keys of length at least 256-bits as required by our CFP--- at least not keys with 256-bits of entropy. The issue is that the shared seed for generating the keys has only 192 significant bits.

We communicated with the submitters on this issue before our acceptance decision and we agree that it is an easy issue to correct. One would assume that they have updated this aspect of the scheme on their project's website.

Please note--- along with my apologies--- this was not the only scheme I noticed that did not meet the implicit entropy requirements for KEMs, but I can't remember what other schemes suffered from this same oversight. I will try to check as my schedule allows, but as a warning, we may find similar (and hopefully easily correctable) errors in other submissions. The good news is that this is something that should be easy to fix in general; the bad news is that it invalidates data that we could really use.

Cheers,
Daniel Smith-Tone

**From:** Barelli Elise <elise.barelli@inria.fr>
**Sent:** Wednesday, May 16, 2018 5:26 AM
**To:** pqc-comments
**Cc:** pqc-forum@list.nist.gov
**Subject:** OFFICIAL COMMENT: DAGS

Dear PQC forum members,

We looked at DAGS submission and found a structural attack on this scheme.

There are two versions of the attack. The first one is based on a partial brute force search and permits to recover the secret elements of DAGS_1,3 and 5 in about $2^{70}$, $2^{80}$ and $2^{58}$ operations.

The second version is based on the resolution of a bilinear system with Grobner bases. We do not have an estimate of the complexity but the attack has been implemented in MAGMA and breaks DAGS_5, claimed 256 bits security key, in about 35 seconds and DAGS_1, claimed 128 bits security key, in 20 minutes.

The details of this attack can be found in an article posted on Arxiv:
https://arxiv.org/abs/1805.05429

We are also available if you have any question about our attack.

Élise Barelli and Alain Couvreur

**From:** Edoardo Persichetti <epersichetti@fau.edu>
**Sent:** Wednesday, May 16, 2018 1:11 PM
**To:** Barelli Elise; Alain Couvreur; pqc-comments; pqc-forum@list.nist.gov
**Cc:** team@dags-project.org
**Subject:** Re: [pqc-forum] OFFICIAL COMMENT: DAGS
**Attachments:** Parameter Change.pdf

Dear all

First of all, I'd like to thank Alain and Elise for the really interesting work on DAGS and the "norm trace code". They were also very kind to notify the DAGS team of the attack ahead of time.

When we chose DAGS parameters, one of our goals was to obtain a more detailed understanding of the complexity of algebraic attacks. All known attacks, at the time, presented in fact only a vague complexity analysis which defined a rather loose environment to set parameters. Thus, we deliberately chose our parameters aggressively, with the aim of stimulating cryptanalysis. It seems we have accomplished exactly that.

The attack mentioned in this work is unrelated to previous known attacks (i.e. it is new). The first version of the attack includes a very detailed complexity analysis. We are thankful for said analysis, as it allows us to adjust our parameters and make sure they are indisputably secure.

We found that, in most cases, it is trivial to defeat the attack: the simplest and most immediate fix consists of changing the underlying base field. In particular, for DAGS_1 it is enough to switch from GF(2^5) to GF(2^6) and for DAGS_3 from GF(2^6) to GF(2^8), and keep every other parameter (code length, dimension, etc.) exactly as is. This brings the attack complexity to the desired security levels and beyond.
For DAGS_5, the parameters were chosen even more aggressively than for the other sets, and in particular it looks like the dyadic order s (DAGS notation) = 2^gamma (attack notation) is too large to provide security - hence the low complexity. Thus, in addition to enlarging the base field, we also need to modify the code parameters - this is still very easily accomplished.

The effect of these changes is an increase in data size which is non-trivial but not at all dramatic. I have attached a simple summary of the parameter changes which hopefully clarifies the picture. The change in the field arithmetic is rather painless too, as the arithmetic for GF(2^6) is already implemented, and it is possible to implement the one for GF(2^8) rather easily. In fact, using GF(2^8) instead of GF(2^6) allows us to make better use of the current space allocation for field elements (a byte) and is a much more natural fit for implementation.

Regarding the second attack strategy, it is evident that it provides a speedup on the first, to the point that it becomes feasible to run the attack on DAGS_1 and DAGS_5 in practice. Once again, we point out that the set of parameters chosen for DAGS_5 was particularly aggressive, which is the reason for such an impressive running time of the attack. However, in the paper, the authors note that the attack fails when applied in practice to DAGS_3, even with the original parameters. We are confident that the same will hold for DAGS_1 and DAGS_5 once we switch to our updated sets of parameters. We have asked Alain and Elise to kindly give us a confirmation of this by attempting to run the attack on all our updated parameters.

To conclude, we would like to thank Alain and Elise again for pointing out an interesting algebraic attack and especially for the great job on the complexity analysis. We believe the attack can be thwarted by a simple parameter change (as described above), and with only minor consequences in terms of data size and speed. Therefore, we would like to reassure NIST and everyone else in this forum that DAGS is still absolutely safe and sound.

We will integrate parameter changes in our "live" version of the spec document at https://www.dags-project.org/#files.

Best Regards,
Edoardo and teh DAGS team

# Change of Parameters after Attack

Find below an overview of the parameter change and its consequences. We write in red the parameters that are different between the two versions. We indicate with N/A the case where the attack is not applicable (fails to run in practice).

**Table 1:** Original DAGS Parameters.

| Parameter Set | $q$ | $m$ | $n$ | $k$ | $s$ | $t$ | $w$ | ISD | Attack 1 | Attack 2 |
|---|---|---|---|---|---|---|---|---|---|---|
| DAGS_1 | $2^5$ | 2 | 832 | 416 | $2^4$ | 13 | 104 | 128 | $\approx 70$ | 15 min |
| DAGS_3 | $2^6$ | 2 | 1216 | 512 | $2^5$ | 11 | 176 | 192 | $\approx 80$ | N/A |
| DAGS_5 | $2^6$ | 2 | 2112 | 704 | $2^6$ | 11 | 352 | 256 | $\approx 58$ | 30 sec |

**Table 2:** New DAGS Parameters.

| Parameter Set | $q$ | $m$ | $n$ | $k$ | $s$ | $t$ | $w$ | ISD | Attack 1 | Attack 2 |
|---|---|---|---|---|---|---|---|---|---|---|
| DAGS_1 | $2^6$ | 2 | 832 | 416 | $2^4$ | 13 | 104 | 128 | $\approx 126$ | ? |
| DAGS_3 | $2^8$ | 2 | 1216 | 512 | $2^5$ | 11 | 176 | 192 | $\approx 288$ | N/A |
| DAGS_5 | $2^8$ | 2 | 1600 | 896 | $2^5$ | 11 | 176 | 256 | $\approx 289$ | ? |

**Table 3:** Change in Sizes.

| Parameter Set | Public Key | | Private Key | | Ciphertext | |
|---|---|---|---|---|---|---|
| DAGS_1 | 6760 | 8112 | 2080 | 2496 | 552 | 656 |
| DAGS_3 | 8448 | 11264 | 3648 | 4864 | 944 | 1248 |
| DAGS_5 | 11616 | 19712 | 6336 | 6400 | 1616 | 1632 |

**From:** Edoardo Persichetti <epersichetti@fau.edu>
**Sent:** Tuesday, December 18, 2018 8:22 PM
**To:** pqc-comments; pqc-forum@list.nist.gov
**Subject:** OFFICIAL COMMENT: DAGS

Dear all

This is to notify the community of the latest progress on DAGS. We investigated a variety of aspects, such as:

- a variant called SimpleDAGS which follows the Niederreiter paradigm featuring a simpler description and a tighter security reduction
- a new set of binary parameters which can be seen as a tradeoff between efficiency and key size
- improved implementation techniques for the scheme

In particular, the improved implementation refers to the updated parameters that have been communicated some time ago in an email on this forum in response to the Barelli-Couvreur paper. Despite the small increase in data size which is necessary to avoid the attack, our implementation is much faster than that provided as reference code in the first round of submissions. We plan to submit our new implementation to SUPERCOP for benchmarking soon.

We would also like to remark that both the updated parameters and the new binary parameters are out of scope of algebraic attacks, and in particular the Barelli-Couvreur attack. Therefore, both q-ary and binary parameters are a perfectly viable choice for DAGS, and the usage of one set or other should mostly be at the implementor's discretion (i.e. a tradeoff as mentioned above). An implementation for the binary case is currently underway and will be distributed upon completion.

A preprint containing all the updates is available at https://eprint.iacr.org/2018/1203. The updated q-ary parameters are already featured in the last version of the DAGS specification document https://dags-project.org/pdf/DAGS_spec_v2.pdf and we plan to incorporate the rest of the material in a successive version of the document

Sincerely,
Edoardo and the DAGS Team