From: Perlner, Ray (Fed)
To: (b) (6)
Subject: RE: Memory
Date: Wednesday, November 15, 2017 1:56:00 PM

I have several additional comments on the project-then-minrank section:

1. The argument that the projection to r+a+v dimensions cannot reduce the rank of the HFE component applies also to the minrank-then-project strategy. It should probably be introduced in the earlier section instead.
2. It is unnecessary to repeat the minors-modeling step as you describe here. “Once a vinegar variable is found, this process is repeated until the vinegar subspace is eliminated.” If you’ve successfully done minors modeling once, as with the minrank-then-project strategy, you already know what linear combination of the public maps is of interest. You can therefore apply the same process to eliminate the vinegar variables at cost only (r + a + v)^\omega q^{r+a+1}.
3. Trying to remove only one vinegar variable with the projection may not be optimal. Removing more variables lets you project further (see next comment), and it reduces the cost of the minrank attack. We may want to generalize to projections that reduce the effective number of vinegar variables by c.
4. When projecting before running minrank, we need to make sure that the resulting instance of the minrank problem (n-a bilinear forms, dimension n+v-k, rank r+a+v - c) is still fully determined
   This requires k \geq n + c -r -a - \sqrt{n-a}. (Equivalently, the dimension of the image of the projection must follow n+v-k \leq r+a+v -c + \sqrt{n-a}).
5. Also, note that in characteristic 2, the symmetric bilinear maps always have even rank. Thus, if we start with an even number of vinegar variables, removing an odd number, c, of vinegar variables will reduce the rank by c+1.

From: Daniel Smith (b) (6)
Sent: Tuesday, November 14, 2017 10:54 AM
To: Perlner, Ray (Fed) <ray.perlner@nist.gov>
Subject: Re: Memory

The language is a bit sloppy here. I think that you mean to represent pi:F^(n+v) -> F^(n+v) as the product phi \circ pi_1 \circ \phi^-1 and pi_2 where pi_1:E -> E and pi_2:F^v -> F_v. That seems right to me. There is always a basis in which a linear map with a kernel of dimension n-r-a has degree q^(n-r-a). More than that, it is always possible to find a low degree central map f' and this low degree projection that compose to the same function. Thanks. I'll add your argument in. That definitely helps.

Cheers!

On Mon, Nov 13, 2017 at 3:45 PM, Perlner, Ray (Fed) <ray.perlner@nist.gov> wrote:

> Ok. Great.

By the way, you say: “We may choose $k=n-\lceil log_q(D)\rceil - a - v$, and expect that the rank of $P\circ\pi$ is still $\lceil log_q(D)\rceil + a + v$, since the HFE component is still likely full rank.”

I don’t think projection from n+v to r+a+v dimensions can decrease the rank of the HFE-component. My reason is as follows:

The unprojected HFE- map can be viewed as a bilinear form acting on the Frobenius powers of the plaintext: (x, x^q, … x^(q^(r+a-1))). The projected HFE- map can be viewed as the same bilinear form acting on (pi(x), pi(x)^q, … pi(x)^(q^(r+a-1))). As long as (pi(x), pi(x)^q, … pi(x)^(q^(r+a-1))) are all linearly independent, the rank should be the same.

Note that, if the projected space within the full n + v dimensional space is of dimension r + a + v, the intersection of the projected space with the HFE subspace is of dimension at least r + a. We can therefore represent pi(x) as a linear combination of the Frobenius powers ranging from (x, … x^(q^(n-r-a))). If the highest Frobenius power in pi(x) is x^(q^(n-r-a)), then the highest Frobenius power in pi(x)^(q^(r+a-1)) is x^(q^(n-1)), the highest Frobenius power in in pi(x)^(q^(r+a-2)) is x^(q^(n-2)), and so on. Since each polynomial contains a Frobenius power that is not in the subsequent polynomials, they are all clearly linearly independent. If the highest Frobenius power in pi(x) is something smaller, the argument still works. E.g. if the highest Frobenius power in pi(x) is x^(q^(n-r-a-3)), then the highest Frobenius power in pi(x)^(q^(r+a-1)) is is x^(q^(n-4)), the highest Frobenius power in in pi(x)^(q^(r+a-2)) is x^(q^(n-5)), and so on. Likewise, everything is still clearly linearly independent.

From: Daniel Smith (b) (6)
Sent: Monday, November 13, 2017 12:20 PM

To: Perlner, Ray (Fed) <ray.perlner@nist.gov>
Subject: Re: Memory

Hi, Ray,

I came about the same estimate by different means. What you said makes perfect sense. I think that I'm right about the low rank output of the minrank attack having a kernel orthogonal to the vinegar subspace. The analysis gives me the same estimate.

Also, I don't think that it is true that we need to apply the HFE- attack after filtering out the subspace. Once we find the vinegar subspace we apply a projection to the low rank map on to the orthogonal complement and then apply the algorithm for recovering U from my HFE- paper. There is no need to do the minrank again. It doesn't change the complexity any, but it is better.

Cheers!

On Mon, Nov 13, 2017 at 12:08 PM, Perlner, Ray (Fed) <ray.perlner@nist.gov> wrote:

> Here's how I think of it:
>
> The input space of the unprojected map is an n+v dimensional space, which is the direct product of the n dimensional HFE input space and the v dimensional space generated by the standard-basis vinegar variables. Our projection defines an r+a+v dimensional subspace of that space.
>
> Now we can consider a basis for the projected space consisting of r+a+v vectors, each with n+v components. This should look like a random r+a+v by n+v matrix. A vinegar variable is projected out iff the r+a+v by v submatrix, consisting of only those columns corresponding to a vinegar variable, has less than full rank. This happens with probability approximately q^-(r+a+1).
>
> Does this help?
>
> **From:** Daniel Smith (b) (6)
> **Sent:** Monday, November 13, 2017 11:10 AM
> **To:** Perlner, Ray (Fed) <ray.perlner@nist.gov>
> **Subject:** Re: Memory
>
> Okay... Maybe this is it.
>
> In the normal basis the quadratic form looks like figure 2 in the paper. The kernel of this is orthogonal to both the vinegar subspace. Maybe we should project away from this subspace (well, maybe all but a one dimensional subspace of it). Then we will have a rank r+a+v map that is (r+a+v+1) x (r+a+v+1) as a matrix. If we look a codim one projections here, the ones that intersect the vinegar space reduce the rank, whereas the HFE ones do not. This is much better. I'm so tired. Am I right?
>
> Cheers!
>
> On Mon, Nov 13, 2017 at 11:03 AM, Daniel Smith (b) (6) wrote:
>
> > I meant advantage of q^(r+a+v).
> >
> > On Mon, Nov 13, 2017 at 11:00 AM, Daniel Smith (b) (6) wrote:
> >
> > > Thanks, Ray.
> > >
> > > I figured it out last night. I have a different probability than you. I think that when we obtain the low rank quadratic form that the kernel is orthogonal to the vinegar subspace. I think that this is necessary for the MinRank to work. If I'm right, then we get an advantage of q^(r+a). But even after that I don't have the same figure. Projecting down to r+a+v requires selecting n-r-a-v linear forms to vanish. The span of these linear forms is of size

q^(n-a-r-v). If each one has a probability of q^-n of begin orthogonal to the HFE subspace, then isn't the probability that you project out a vinegar variable q^-(r+a+v) instead of q^-(r+a+1)? Maybe I'm making a mistake.

Cheers,
Daniel

On Mon, Nov 13, 2017 at 10:35 AM, Perlner, Ray (Fed) <ray.perlner@nist.gov> wrote:

> I assume this is for the HFEv- paper, not the SRHFE paper. Right? I believe it works as follows:
>
> Do minors modeling to find a rank r+a+v linear combination of the public equations (over the extension field). Note that this is not yet a full key recovery. To get a full key recovery, take the rank r+a+v bilinear form and randomly project down to r+a+v variables. If the rank after projection less than r+a+v, you know you've projected out a vinegar variable (Note, this happens with probability q^-(r+a+1).) To figure out which linear combination of the projection equations was required to remove the vinegar variable, just apply a random linear mixing to the projection equations and see if you can take away a projection equation without increasing the rank. Repeat the process until you're left with the single projection equation that removes the vinegar variable. Do this v times, and you remove all the vinegar variables. Once the vinegar variables are gone, you can do the HFE- attack from your previous paper to get a full key recovery.
>
> **From:** Daniel Smith (b) (6)
> **Sent:** Sunday, November 12, 2017 11:10 AM
> **To:** Perlner, Ray (Fed) <ray.perlner@nist.gov>
> **Subject:** Memory
>
> Hi,
>
> I've forgotten what the MinRank then Projection idea was. Is that to do MinRank with twice as many variables as HFE and look for structure in the solution? It seems like a kind of stupid idea to me, though it makes sense to address it, I guess. I don't really remember this, though. If we were to call the central map a bivariate map f(X,V) over the extension E, then it makes sense to do MinRank here, and then relative to the vector [X,X^q,...,X^q^(n-1),V,V^q,...,V^q(n-1)], the MinRank solution would look like a block matrix with the upper left block of HFE shape and the others random subject to the degree bound on X. Is that the idea we had for MinRank first?
>
> I should have done this stuff when it was in my head.
>
> Cheers,
> Daniel