From: (b) (6)
To: Perlner, Ray A. (Fed)
Subject: Re: more hmmms
Date: Thursday, June 4, 2020 7:18:58 PM

I think this issue applied to GeMSS is more interesting, though...

On Thu, Jun 4, 2020 at 6:22 PM Perlner, Ray A. (Fed) <ray.perlner@nist.gov> wrote:

> Oh. I see. That's the attack you had in mind. Yeah I wouldn't be ok with a 250-bit signature that worked that way (I see no reason you can't get smaller with something like Feistel-Patarin though.) I suppose it's also true that you could in principle get a better attack with a 256-bit hash value if the verification query is super cheap, by doing fewer hash queries and more verification queries than 2^128. That said, the rainbow query should cost at least as many bit operations as the number of bits in the uncompressed Rainbow public key, which is more than 2^15. Maybe you can amortize those queries somehow, though.
>
> **From:** Daniel Smith (b) (6)
> **Sent:** Thursday, June 4, 2020 5:02 PM
> **To:** Perlner, Ray A. (Fed) <ray.perlner@nist.gov>
> **Subject:** Re: more hmmms
>
> Well how do hash-and-sign signatures work? (Not necessarily anything we have in our process, but generically. For us we have things like the Fiestel-Patarin construction.) So you take message m, hash it to get H(m), and sign it, s=S(H(m)). Then you have a string with the property that V(s)=H(m). For EUF-CMA, the adversary chooses the message and only needs to demonstrate the ability to construct a single valid signature even with access to signatures for other messages. (Or in the strong notion, the adversary needs to demonstrate the ability to construct a valid signature even of a message for which he has a different valid signature.) So why can't the adversary generate pairs (m',s') until H(m')=V(s')? That is not two valid signatures for the same hash value, but it seems to me that it does violate EUF-CMA. What am I not understanding?
>
> On Thu, Jun 4, 2020 at 4:34 PM Perlner, Ray A. (Fed) <ray.perlner@nist.gov> wrote:
>
> > Don't see any reason why not.
> >
> > **From:** Daniel Smith (b) (6)
> > **Sent:** Thursday, June 4, 2020 4:31 PM
> > **To:** Perlner, Ray A. (Fed) <ray.perlner@nist.gov>
> > **Subject:** Re: more hmmms

Let me ask the same question in a different way. Would you be okay with a scheme with 250 bit signatures?

On Thu, Jun 4, 2020 at 4:02 PM Daniel Smith (b) (6) wrote:

> I agree. I don't think that there is an issue with anything. My concern would be rainbow since if for some reason we were to need resistance to collision attacks to the 143-bit level (beyond birthday barrier anyway), then Rainbow Ia would have too few equations. I don't think that it is reasonable, though, even with unit oracle queries. (Because it is not a property of the function or its structure and the same discussion would apply to AES, don't you think?)
>
> On Thu, Jun 4, 2020 at 3:57 PM Perlner, Ray A. (Fed) <ray.perlner@nist.gov> wrote:
>
> > You mean random oracle queries? Given that we're assuming the random oracle is implemented by something like SHA in the actual standard, shouldn't a random oracle query cost something like 2^18 bit operations? If so, 2^128 oracle queries is comfortably above the threshold for category 1. No?
> >
> > **From:** Daniel Smith (b) (6)
> > **Sent:** Thursday, June 4, 2020 3:52 PM
> > **To:** Perlner, Ray A. (Fed) <ray.perlner@nist.gov>
> > **Subject:** more hmmms
> >
> > Hi, Ray,
> >
> > I have a question on your opinion about collision attacks for hash-and-sign signatures. Should we be interested in hitting the 2^128 level for this for level I, or 2^143? It seems to me that the 2^128 is more reasonable here since you can treat the signing oracle as truly random (so it makes less sense a comparison to inverting AES). I just want your opinion on it.
> >
> > Cheers,
> >
> > Daniel