From: Serge Vaudenay <serge.vaudenay@epfl.ch> via pqc-forum <pqc-forum@list.nist.gov>
To: pqc-forum@list.nist.gov
Subject: [pqc-forum] IND-1CCA transform
Date: Friday, May 27, 2022 05:24:40 PM ET

Dear PQC Forum,

Our paper called "A note on IND-qCCA security in the ROM and its applications: CPA security is sufficient for TLS 1.3" will be published in this year Eurocrypt proceedings (eprint version: https://gcc02.safelinks.protection.outlook.com/?url=https%3A%2F%2Feprint.iacr.org%2F2021%2F844&amp;data=05%7C01%7Candrew.regenscheid%40nist.gov%7Ca3981d61406746720a0508da40274dec%7C2ab5d82fd8fa4797a93e054655c61dec%7C1%7C0%7C637892834804643502%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&amp;sdata=1aEASovrEEheDWJ72Ltj2P7304KuoJ1r6e4X4i4EDUQ%3D&amp;reserved=0 <https://gcc02.safelinks.protection.outlook.com/?url=https%3A%2F%2Feprint.iacr.org%2F2021%2F844&amp;data=05%7C01%7Candrew.regenscheid%40nist.gov%7Ca3981d61406746720a0508da40274dec%7C2ab5d82fd8fa4797a93e054655c61dec%7C1%7C0%7C637892834804643502%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&amp;sdata=1aEASovrEEheDWJ72Ltj2P7304KuoJ1r6e4X4i4EDUQ%3D&amp;reserved=0>).

It is known that IND-1CCA (IND-CCA restricted to *1* decapsulation query) is enough for applications such as TLS but it wasn't known how to obtain IND-1CCA at a cheaper cost than IND-CCA.

In this work, we show that IND-1CCA KEMs can very easily be constructed from any CPA-secure PKE in the (Q)ROM. In particular, compared to Fujisaki-Okamoto-like transforms, our transforms do not use derandomization and re-encryption when decapsulating.

IND-1CCA KEMs can be used in several popular protocols, such as (PQ) TLS 1.3 or the recently proposed KEMTLS (https://gcc02.safelinks.protection.outlook.com/?url=https%3A%2F%2Feprint.iacr.org%2F2020%2F534&amp;data=05%7C01%7Candrew.regenscheid%40nist.gov%7Ca3981d61406746720a0508da40274dec%7C2ab5d82fd8fa4797a93e054655c61dec%7C1%7C0%7C637892834804643502%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&amp;sdata=l2sE6azmgkLnEO13IVC1PLyLv4eVHYq89A0zPqRd4hk%3D&amp;reserved=0

<https://gcc02.safelinks.protection.outlook.com/?url=https%3A%2F%2Feprint.iacr.org%2F2020%2F534&amp;data=05%7C01%7Candrew.regenscheid%40nist.gov%7Ca3981d61406746720a0508da40274dec%7C2ab5d82fd8fa4797a93e054655c61dec%7C1%7C0%7C637892834804643502%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&amp;sdata=l2sE6azmgkLnEO13IVC1PLyLv4eVHYq89A0zPqRd4hk%3D&amp;reserved=0>).

Compared to existing instantiations of these protocols with IND-CCA KEMs, using IND-1CCA KEMs derived with our transforms would (at least) halve the decapsulation time and would simplify the implementation of the primitives (i.e. no re-encryption when decapsulating).

This could be especially interesting for isogeny-based schemes, which suffer from slow computation compared to other PQ KEMs. It is also possible to use the second transform in the paper to obtain a IND-1CCA version of SIDH which preserves the symmetry of the scheme (i.e. the ciphertext does not depend on the public-key).

Best regards

Loïs Huguenin-Dumittan and Serge Vaudenay

From: D. J. Bernstein <djb@cr.yp.to> via pqc-forum@list.nist.gov
To: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] IND-1CCA transform
Date: Friday, May 27, 2022 08:00:47 PM ET
Attachments: smime.p7m

'Serge Vaudenay' via pqc-forum writes:

> Compared to existing instantiations of these protocols with IND-CCA KEMs, using IND-1CCA KEMs derived with our transforms would (at least) halve the decapsulation time and would simplify the implementation of the primitives (i.e. no re-encryption when decapsulating).

Can you elaborate on this simplicity evaluation? Previous evaluations (with IND-CPA and hashing rather than IND-1CCA, but this distinction doesn't seem to matter here) indicate that the key-exchange ranking from simplest to most complicated is

(1) pure IND-CCA KEM, as in https://cr.yp.to/talks.html#2016.02.24;
(2) about 10% faster overall: mix IND-CCA KEM with IND-1CCA KEM;
(3) SIGMA-style as in TLS: signature system + IND-1CCA KEM.

The server needs some sort of long-term identity key, and all of the solutions use either a signature key or an IND-CCA key for this, so it's not as if the implementor can stop with just IND-1CCA (or IND-CPA).

One could try arguing that, well, a signature system is needed for other reasons, and then #3 avoids the complication of an IND-CCA KEM. But it's just as easy to argue that an IND-CCA KEM is needed for other reasons, and then adding an IND-1CCA KEM is an unnecessary complication. See also https://www.imperialviolet.org/2018/12/12/cecpq2.html: "CPA vs CCA security is a subtle and dangerous distinction, and if we're going to invest in a post-quantum primitive, better it not be fragile."

---D. J. Bernstein

--

From: Serge Vaudenay <serge.vaudenay@epfl.ch> via pqc-forum <pqc-forum@list.nist.gov>
To: pqc-forum@list.nist.gov
CC: Loïs Huguenin-Dumittan <lois.huguenin-dumittan@epfl.ch>
Subject: Re: [pqc-forum] IND-1CCA transform
Date: Sunday, May 29, 2022 05:55:20 AM ET

Dear Dan,

We meant that decapsulation is simpler to implement than its FO-like counterparts and roughly shows a 2x speedup. There are more details in our paper.

We hope as well that this result will not suggest to degrade security where IND-CCA is really needed. Of course, it does not replace signature.

Loïs (in cc) and I will be at Eurocrypt and will be happy to discuss in person.

Best regards

Serge

On 28.05.22 01:59, D. J. Bernstein wrote:
> 'Serge Vaudenay' via pqc-forum writes:
>> Compared to existing instantiations of these protocols with IND-CCA KEMs,
>> using IND-1CCA KEMs derived with our transforms would (at least) halve the
>> decapsulation time and would simplify the implementation of the primitives
>> (i.e. no re-encryption when decapsulating).
>
> Can you elaborate on this simplicity evaluation? Previous evaluations
> (with IND-CPA and hashing rather than IND-1CCA, but this distinction
> doesn't seem to matter here) indicate that the key-exchange ranking from
> simplest to most complicated is
>

>   (1) pure IND-CCA KEM, as in https://gcc02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fcr.yp.to%2Ftalks.html%232016.02.24&amp;data=05%7C01%7Candrew.regenscheid%40nist.gov%7C7f96c8e316e54410248508da415953aa%7C2ab5d82fd8fa4797a93e054655c61dec%7C1%7C0%7C637894149202900335%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&amp;sdata=z8Q0QTHaZMTGPB%2BC0lEkANGOEixB9Syvqyu2hnoJgRY%3D&amp;reserved=0;
>   (2) about 10% faster overall: mix IND-CCA KEM with IND-1CCA KEM;
>   (3) SIGMA-style as in TLS: signature system + IND-1CCA KEM.
>
> The server needs some sort of long-term identity key, and all of the
> solutions use either a signature key or an IND-CCA key for this, so it's
> not as if the implementor can stop with just IND-1CCA (or IND-CPA).
>
> One could try arguing that, well, a signature system is needed for other
> reasons, and then #3 avoids the complication of an IND-CCA KEM. But it's
> just as easy to argue that an IND-CCA KEM is needed for other reasons,
> and then adding an IND-1CCA KEM is an unnecessary complication. See also
> https://gcc02.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwww.imperialviolet.org%2F2018%2F12%2F12%2Fcecpq2.html&amp;data=05%7C01%7Candrew.regenscheid%40nist.gov%7C7f96c8e316e54410248508da415953aa%7C2ab5d82fd8fa4797a93e054655c61dec%7C1%7C0%7C637894149202900335%7CUnknown%7CTWFpbGZsb3d8eyJWIjoiMC4wLjAwMDAiLCJQIjoiV2luMzIiLCJBTiI6Ik1haWwiLCJXVCI6Mn0%3D%7C3000%7C%7C%7C&amp;sdata=StqDSJVyyHQjEdfqaRHCHMHDi6%2Bd%2BK2CNKV43aJ0m5g%3D&amp;reserved=0: "CPA vs CCA
> security is a subtle and dangerous distinction, and if we're going to
> invest in a post-quantum primitive, better it not be fragile."
>
> ---D. J. Bernstein
>

--

You received this message because you are subscribed to the Google Groups "pqc-forum" group.
To unsubscribe from this group and stop receiving emails from it, send an email to pqc-forum+unsubscribe@list.nist.gov.
To view this discussion on the web visit https://groups.google.com/a/list.nist.gov/d/msgid/pqc-forum/b4d286e2-7255-a997-1368-8fef12fab2db%40epfl.ch.

From: D. J. Bernstein <djb@cr.yp.to> via pqc-forum@list.nist.gov
To: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] IND-1CCA transform
Date: Tuesday, May 31, 2022 12:58:17 PM ET
Attachments: smime.p7m

'Serge Vaudenay' via pqc-forum writes:

> We meant that decapsulation is simpler to implement than its FO-like
> counterparts and roughly shows a 2x speedup.

Certainly skipping reencryption makes a KEM simpler (and faster). But I don't see how you get to the conclusion that this "would simplify the implementation of the primitives" used by "these protocols".

Previous evaluations say that the simplest way to achieve the security goals for these key-exchange protocols---note that having a long-term server identity is part of the goals---is to use just an IND-CCA KEM.

For example, key exchange based purely on IND-CCA NewHope is simpler than combining signatures with the pre-NISTPQC version of NewHope. The complication of reencryption inside IND-CCA NewHope is outweighed by the complication of adding a separate signature system.

The simplest (and fastest) baseline in your comparison is KEMTLS. I don't see how your paper provides a simpler option. You're making KEM-based key exchange more complicated to implement, asking for an IND-CCA KEM _and_ a faster IND-1CCA KEM. Even if all components are shared (starting with the reencryption-encapsulation overlap), there are more API functions to implement, test, document, verify, etc., and there isn't anything removed from the code. Am I missing something?

I agree that there's a speedup (about 10% in the overall protocol), but my question is about the simplicity claim.

---D. J. Bernstein

--

From: Blumenthal, Uri - 0553 - MITLL <uri@ll.mit.edu> via pqc-forum@list.nist.gov
To: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] IND-1CCA transform
Date: Tuesday, May 31, 2022 01:12:59 PM ET
Attachments: smime.p7m

On 5/31/22, 12:58, "D. J. Bernstein" <pqc-forum@list.nist.gov on behalf of djb@cr.yp.to> wrote:

> 'Serge Vaudenay' via pqc-forum writes:
> > We meant that decapsulation is simpler to implement than its FO-like
> > counterparts and roughly shows a 2x speedup.
>
> Certainly skipping reencryption makes a KEM simpler (and faster). But I
> don't see how you get to the conclusion that this "would simplify the
> implementation of the primitives" used by "these protocols".
>
> . . .
>
> The simplest (and fastest) baseline in your comparison is KEMTLS. I
> don't see how your paper provides a simpler option. You're making
> KEM-based key exchange more complicated to implement, asking for an
> IND-CCA KEM _and_ a faster IND-1CCA KEM.

Would _both_ IND-CCA _and_ IND_1CCA KEMs be necessary for KEMTLS? Can they not both be IND-1CCA? What would be the consequences…?

> I agree that there's a speedup (about 10% in the overall protocol), but
> my question is about the simplicity claim.

;-)

--
You received this message because you are subscribed to the Google Groups "pqc-forum" group.

From: Douglas Stebila <dstebila@gmail.com> via pqc-forum@list.nist.gov
To: Blumenthal, Uri - 0553 - MITLL <uri@ll.mit.edu>
CC: pqc-forum@list.nist.gov
Subject: Re: [pqc-forum] IND-1CCA transform
Date: Tuesday, May 31, 2022 03:09:29 PM ET

On May 31, 2022, at 13:11, Blumenthal, Uri - 0553 - MITLL <uri@ll.mit.edu> wrote:

>
>> The simplest (and fastest) baseline in your comparison is KEMTLS. I
>> don't see how your paper provides a simpler option. You're making
>> KEM-based key exchange more complicated to implement, asking for an
>> IND-CCA KEM _and_ a faster IND-1CCA KEM.
>
> Would _both_ IND-CCA _and_ IND_1CCA KEMs be necessary for KEMTLS? Can they not both be IND-1CCA? What would be the consequences…?

The KEM used for long-term authentication in KEMTLS does need to have IND-CCA security, not just IND-1CCA: since the user will use the same long-term key in many sessions, it needs to be able to handle many chosen ciphertext decapsulations.

It is only the KEM used for the ephemeral key exchange for which it suffices to have IND-1CCA security. And even this assumes that such ephemeral keys are really used in only a single session, which has not always been the case for "ephemeral" Diffie-Hellman in some TLS implementations.

Douglas