From: Moody, Dustin (Fed)
To: Alperin-Sheriff, Jacob (Fed)
Subject: RE: Response on Library Usage
Date: Thursday, August 3, 2017 10:38:00 AM

Do you want to send new draft text?

(and that's what we'd decided at our previous meeting – but we're now changing our minds!)

From: Alperin-Sheriff, Jacob (Fed)
Sent: Thursday, August 03, 2017 10:37 AM
To: Bassham, Lawrence E (Fed) <lawrence.bassham@nist.gov>; Perlner, Ray (Fed) <ray.perlner@nist.gov>; Moody, Dustin (Fed) <dustin.moody@nist.gov>
Cc: internal-pqc <internal-pqc@nist.gov>
Subject: Re: Response on Library Usage

I would agree with allowing the reference implementation to call the third party libraries, and only wrote it as I did because I thought someone else wanted it that way …

From: "Bassham, Lawrence E (Fed)" <lawrence.bassham@nist.gov>
Date: Thursday, August 3, 2017 at 7:31 AM
To: "Perlner, Ray (Fed)" <ray.perlner@nist.gov>, "Alperin-Sheriff, Jacob (Fed)" <jacob.alperin-sheriff@nist.gov>, "Moody, Dustin (Fed)" <dustin.moody@nist.gov>
Cc: internal-pqc <internal-pqc@nist.gov>
Subject: Re: Response on Library Usage

In particular, to allow the reference implementation to call OpenSSL seems like it could be a good idea. I don't know enough about potential submissions to know how NTL and GMP would contribute or detract from the readability and understandability of the reference implementation.

On: 02 August 2017 16:00, "Perlner, Ray (Fed)" <ray.perlner@nist.gov> wrote:

Is it really a good idea to prevent the reference implementation from calling third party C++ and assembly libraries? I don't think it would improve readability if every submitter had to implement their own C versions of AES, SHA, and the like. It seems like we should either loosen this requirement, or we should provide API calls for common symmetric primitives. (Did we say we were going to do that anyway?)

From: Alperin-Sheriff, Jacob (Fed)

Sent: Wednesday, August 02, 2017 3:29 PM
To: Moody, Dustin (Fed) <dustin.moody@nist.gov>
Cc: internal-pqc <internal-pqc@nist.gov>
Subject: Re: Response on Library Usage

Maybe rewrite Q3 as

Q3: What exceptions, if any, are there to the requirement for ANSI C source code? In particular, may C++ code or assembly optimizations be used?
A3: The mandatory reference implementation should be written in ANSI C alone, with no C++ and no assembly usage of any kind, including inline assembly.

For the mandatory optimized implementations, all original and new code written for the submission should be written in as ANSI C-like a manner as possible and may not contain any assembly (including inline assembly), subject to some caveats.

In particular, implementations that use NTL (see Question and Answer 16 for details on the use of third-party open source libraries) are necessarily allowed to be written in C++. However, the original and new code in this submission must still be as ANSI C-like as possible, and should only use C++ functionality where absolutely required in order to use NTL. In particular, as with code using any other third-party open source code, the submission must contain build scripts for both Windows and Linux that compile properly on the Intel x64 reference platform using version 6.4.0 of the GNU Compiler Collection (GCC).

Note that we are allowing the use of third party open-source libraries in the mandatory optimized implementations that themselves rely on assembly optimizations; see Question and Answer 16 for details.

Any optional additional implementations that submitters wish to include are subject to no constraints at all regarding the language and platform.

From: "Moody, Dustin (Fed)" <dustin.moody@nist.gov>
Date: Wednesday, August 2, 2017 at 3:22 PM
To: "Alperin-Sheriff, Jacob (Fed)" <jacob.alperin-sheriff@nist.gov>
Subject: RE: Response on Library Usage

I don't quite understand this sentence:

Furthermore, while no assembly (including inline assembly) can be used as part the original and new code in the mandatory optimized implementation, we are allowing the use of third party open-source code that themselves rely on assembly optimizations, subject to the constraints described in Question and Answer 16.

Particularly the first part “while no assembly (including inline assembly) can be used as part the original and new code in the mandatory optimized implementation”.

From: Alperin-Sheriff, Jacob (Fed)
Sent: Wednesday, August 02, 2017 3:19 PM
To: internal-pqc <internal-pqc@nist.gov>
Subject: Re: Response on Library Usage

It was actually overly complete.

Revised Draft

Q3: What exceptions, if any, are there to the requirement for ANSI C source code? In particular, may C++ code or assembly optimizations be used?
A3: The mandatory reference implementation should be written in ANSI C alone, with no C++ and no assembly usage of any kind, including inline assembly.

For the mandatory optimized implementations, all original and new code written for the submission should be written in as ANSI C-like a manner as possible, subject to some caveats.

In particular, implementations that use NTL (see Question and Answer 16 for details on the use of third-party open source libraries) are necessarily allowed to be written in C++. However, the original and new code in this submission must still be as ANSI C-like as possible, and should only use C++ functionality where absolutely required in order to use NTL. In particular, as with code using any other third-party open source code, the submission must contain build scripts for both Windows and Linux that compile properly on the Intel x64 reference platform using version 6.4.0 of the GNU Compiler Collection (GCC).

Furthermore, while no assembly (including inline assembly) can be used as part the original and new code in the mandatory optimized implementation, we are allowing the use of third party open-source code that themselves rely on assembly optimizations, subject to the constraints described in Question and Answer 16.

Any optional additional implementations that submitters wish to include are subject to no constraints at all regarding the language and platform.

Q16: Can third party open-source code be used in submissions?
A16:
Third-party code cannot be used in the mandatory reference implementation.

In the mandatory optimized implementation, submissions may use NTL Version 10.5.0 (http://www.shoup.net/ntl/download.html), GMP Version 6.1.2 (https://gmplib.org), and OpenSSL Version 1.10f (https://www.openssl.org/source). Submitters may assume that these libraries are

installed on the reference platform and do not need to provide them along with their submissions.

If a submitter wishes to use a third-party open source library other than the ones specified above, they must send a request to NIST at pqc-comments@nist.gov by September 1st, 2017, with the name of the library and a link to the primary website hosting it from which it may be downloaded. NIST will either approve or deny this request within 2 weeks of receiving it. Should a request be approved, it will be added to the above list of acceptable third-party open source libraries provided in this FAQ.

All submission packages using third-party open source code should contain build scripts which will allow for seamless "one-stop" building of the submissions.
For example, on a Linux platform, it should require no more work to build the than running the standard
> ./configure [--options]
> make
> make install
succession of commands. In particular, the build process should be able to find the versions of these libraries specified above that will be pre-installed on the reference platform.

Separate build scripts should be included for the reference Windows platform and reference Linux platform that work using the GNU Compiler Collection version 6.4.0 and related tools as well as any platform-specific commands required.

In addition, as part of the written submission, the submitter shall describe in their own words the functionalities provided by any algorithms from third-party open-source libraries that are used in the implementations.

From: "Moody, Dustin (Fed)" <dustin.moody@nist.gov>
Date: Wednesday, August 2, 2017 at 3:17 PM
To: "Alperin-Sheriff, Jacob (Fed)" <jacob.alperin-sheriff@nist.gov>
Subject: RE: Response on Library Usage

Jacob,
Is your new proposed text for A3 complete? The line seems to be missing some text at the end….

Thanks

From: Alperin-Sheriff, Jacob (Fed)
Sent: Wednesday, August 02, 2017 3:07 PM
To: internal-pqc <internal-pqc@nist.gov>
Subject: Response on Library Usage

First I should note (we didn't clearly state this), that we should be fully replacing BOTH Question and Answer 3 and Question and Answer 16 with this new response in the FAQ, as it is modifying/clarifying them both.

Q3: Does the requirement for ANSI C source code preclude the use of assembly language optimizations?

A3: The optimized code required as part of the submission package should be ANSI C with no assembly (this includes inline assembly). This code is meant to be portable. If significant optimizations can be made with assembly, then it can be included as an additional implementation and discussed in the performance analysis.

Q16: Can third party open-source code be used in submissions?

A16: In short, they may be used, with the following caveats.

1. The library source code should be integrated into the submission package in a self-contained manner. This means that the submission package should contain build scripts which will allow for seamless "one-stop" building of the submitter's original code and all dependencies.

For example, on a Linux platform, it should require no more work to build the than running the standard

> ./configure [--options]
> make
> make install

succession of commands. The build process should not require the installation of any new libraries that are not contained in the submission package.

Separate build scripts should be included for the reference Windows platform and reference Linux platform that work using the GCC Compiler Collection (or ports thereof) and related tools as well as any platform-specific commands required.

2. As part of the written submission, the submitter shall describe in their own words the functionalities provided by any algorithms from third-party open-source libraries that are used in the implementations.
3. The submitter is responsible for ensuring that they abide by all requirements of the license (if any) under which said library has been released.

I also think we should specify a specific version of the GNU Compiler Collection (GCC) that they should target to ensure things compile. (I chose 6.4.0 instead of the latest 7.1 because I figure 6.4.0 is more mature and stable; if anyone disagrees I'm open to different version suggestions).

Incidentally, I also talked to Mike Cooper of the Cryptographic Module Validation Program (about whether he thought there was any good reason to avoid the non-FIPS validated more extensive OpenSSL package; he said no) and he raised some good points to think about later on in the process about algorithm testability

Response drafts (for the FAQ):

Q3: What exceptions, if any, are there to the requirement for ANSI C source code? In particular, may C++ code or assembly optimizations be used?
A3: The mandatory reference implementation should be written in ANSI C alone, with no C++ and no assembly usage of any kind, including inline assembly. While they may use

For the mandatory optimized implementations, all original and new code written for the submission should be written in as ANSI C-like a manner as possible, subject to some caveats.

In particular, implementations that use NTL (see Question 16 for details on the use of third-party open source libraries) are necessarily allowed to be written in C++. However, the original and new code in this submission must still be as ANSI C-like as possible, and should only use C++ functionality where absolutely required in order to use NTL. In particular, as with code using any other third-party open source code, the submission must contain build scripts for both Windows and Linux that compile properly on the Intel x64 reference platform using version 6.4.0 of the GNU Compiler Collection (GCC).

Furthermore, while no assembly (including inline assembly) can be used as part the original and new code in the mandatory optimized implementation, we are allowing the use of third party open-source code that themselves rely on assembly optimizations, subject to the constraints described in Question 16.

Any optional additional implementations that submitters wish to include are subject to no constraints at all regarding the language and platform.

Q16: Can third party open-source code be used in submissions?
A16:
Third-party code cannot be used in the mandatory reference implementation.

In the mandatory optimized implementation, submissions may use NTL Version 10.5.0 (http://www.shoup.net/ntl/download.html), GMP Version 6.1.2 (https://gmplib.org), and OpenSSL Version 1.10f (https://www.openssl.org/source). Submitters may assume that these libraries are installed on the reference platform and do not need to provide them along with their submissions.

If a submitter wishes to use a third-party open source library other than the ones specified above, they must send a request to NIST at pqc-comments@nist.gov by September 1st, 2017, with the name of the library and a link to the primary website hosting it from which it may be downloaded. NIST will either approve or deny this request within 2 weeks of receiving it. Should a request be approved, it will be added to the above list of acceptable third-party open source libraries provided in this FAQ.

All submission packages using third-party open source code should contain build scripts which will allow for seamless “one-stop” building of the submissions.
For example, on a Linux platform, it should require no more work to build the than running the standard
> ./configure [--options]
> make
> make install
succession of commands. In particular, the build process should be able to find the versions of these libraries specified above that will be pre-installed on the reference platform.

Separate build scripts should be included for the reference Windows platform and reference Linux platform that work using the GNU Compiler Collection version 6.4.0 and related tools as well as any platform-specific commands required.

In addition, as part of the written submission, the submitter shall describe in their own words the functionalities provided by any algorithms from third-party open-source libraries that are used in the implementations.

—Jacob Alperin-Sheriff