Comment on Post-Quantum Cryptography Requirements and Evaluation Criteria

Richard Barnes <rbarnes@mozilla.com>

Fri 9/16/2016 9:03 PM

PQC Public Comments

To: pqc-comments <pqc-comments@nist.gov>;

Cc: Chen, Lily (Fed) <lily.chen@nist.gov>; JC Jones <jcj@mozilla.com>;

Dear Ms. Chen,

Thank you for the opportunity to submit comments on the Submission Requirements and Evaluation Criteria to be used in NIST's process for standardizing quantum-resistant public-key cryptographic algorithms.

Mozilla's mission as a non-profit organization is to promote openness, innovation, and opportunity online. Protecting the security of Internet communications is a core part of that mission. Mozilla is a major user of cryptographic standards. Our products engage in billions of HTTPS transactions per day, and we maintain of one the most widely used open-source cryptographic libraries. We are also deeply involved in the standardization of cryptographic protocols in the IETF. Eric Rescorla, a Mozilla fellow, is editor of the TLS specification, Richard Barnes is a former member of the Internet Engineering Steering Group, and several other Mozilla staff are active in cryptography-related IETF working groups. It is from this perspective that we offer our comments.

Our primary concern with the proposed process is that it needs to ensure that the algorithms standardized through it work in the real world. The draft documents provided for comment present problems at both legal and technical levels.

1. Submitted algorithms must be usable without compensation to patent holders (RAND-Z, not only RAND) and implementations must bear an open-source license

The draft Call for Proposals is correct to note that "royalty-free availability of cryptosystems and implementations has facilitated adoption". It is surprising that this observation is followed by allowances for royalty-bearing cryptosystems, e.g., in the Statement by Patent Owners 2.D.2. Allowing royalty-bearing cryptosystems to be submitted will inhibit both the thorough evaluation of proposals and their eventual adoption by industry.

As the draft CFP acknowledges in several places, contributions by the broader research community will be essential in helping NIST make a thorough evaluation of the submitted algorithms. In order to make these contributions, members of the community including researchers in the commercial and academic sectors will need be able to implement the submitted algorithms. A requirement to license patents for such implementations would make it impossible for many researchers to participate in evaluation of algorithms, undermining the completeness and the legitimacy of the NIST process.

In this context, it should be noted that U.S. Courts have all but eliminated the availability of the "experimental use defense" to patent infringement: "[R]egardless of whether a particular institution or entity is engaged in an endeavor for commercial gain, so long as the act is in furtherance of the alleged infringer's legitimate business and is not solely for amusement, to satisfy idle curiosity, or for strictly philosophical inquiry**,** the act does not qualify for the very narrow and strictly limited experimental use defense. Moreover, the profit or non-profit status of the user is not determinative." *Madey v. Duke Univ.*, 307 F.3d 1351, 1362 (Fed. Cir. 2002) (finding university's research projects with no commercial application still "unmistakably further the institution's legitimate business objectives" in education). (See also *Soitec, S.A. v. Silicon Genesis Corp.*, 81 Fed.Appx. 734, 737 (Fed.Cir. 2003), "There is no fair use or research and development exception for infringement of normal commercial processes.")

There is also a need for researchers to be able to use and modify the submitted implementations in order to evaluate the costs and benefits of the algorithm in different contexts. For example, a researcher might adapt the optimized implementation to run on a machine architecture common in mobile devices to see if the algorithm is suitable for use in that environment. In order to allow this usage, it is imperative that the submitted implementations be licensed under an open-source license, and in particular one that allows for the creation of derivative works. We encourage NIST to specify a small set of acceptable open-source licenses. There are several such licenses available: Many of the policies in the US CIO's list of licensing resources recommend the CC0 or CC-BY licenses; we would also find licenses such as the MIT, BSD, Mozilla Public License, or Apache Public License acceptable.

Standardization of a royalty-bearing algorithm would strongly inhibit industry adoption of the algorithm, especially in open standards organizations and open-source projects. The IETF has historically avoided standardization of royalty-bearing algorithms, so it would be difficult to establish the ancillary protocol standards needed to integrate a NIST-standard algorithm into Internet protocols.

Open source projects such as OpenSSL and NSS are critical to the deployment of cryptographic protocols on the Internet. These projects are often unable to use algorithms that are only available through royalty-bearing licenses. In particular, it would be extremely difficult for Mozilla to include a royalty-bearing algorithm in its products (including Firefox) even if it were standardized by NIST. The only use of royalty-bearing technologies in Firefox today (the H.264 video codec) was only possible because an existing license holder offered to cover royalty costs for Firefox users and because significant engineering effort was spent enabling the codec to be distributed within the terms of the license.

2. Algorithms need to be evaluated as they will be used

It is crucial for the success of this process that NIST not evaluate submitted algorithms in the abstract, but as they will be deployed in modern information security systems. To that end, we are glad to see that the evaluation criteria place the impact on Internet protocols as a primary measure of an algorithm's utility.
Along these lines, it should be noted that verifying that an algorithm is IND-CCA2 secure in the abstract might not mean that it provides this level of security in practice. For example, if there are assumptions underlying the IND-CCA2 proof that a protocol cannot meet, then the algorithm might not provide an acceptable level of security for that protocol. The evaluation criteria should make clear that algorithms must not rely on assumptions in security proofs that cannot be satisfied by common security protocols.

While we agree with NIST's choice to rule hybrid algorithms out of scope for this process, it is nonetheless true that hybrid algorithms will be an important part of the deployment process for post-quantum algorithms. We encourage NIST to consider the suitability of algorithms for use in hybrid schemes as an evaluation criterion, with preference for algorithms that are more amenable to hybridization.

Looking at how public-key algorithms are used in modern Internet protocols, it is clear that key establishment and signature are much more important features to implement than public-key encryption. Forward secrecy in particular has been a feature that the community of TLS operators has worked very hard to make pervasive, in order to guard against temporary compromises. Even messaging security protocols, which have traditionally relied fairly heavily on public-key encryption, have been moving toward frameworks that provide more forward secrecy by relying more on key establishment instead of public-key encryption. We would be comfortable if NIST de-emphasized or dropped public-key encryption from evaluation, especially given that in many cases, it can be replaced by a combination of key establishment and symmetric encryption.

Finally, the selection of x64 as a reference platform is understandable, but perhaps not a complete reflection of modern computing environments. It is increasingly common for cryptography to be done on mobile devices, mostly using ARM architectures, and operators are increasingly selecting algorithms based on their performance in mobile environments (e.g., preferring ChaCha20 over AES). Emerging platforms for the Internet of Things will likely be bringing similar constraints in the near future. We would encourage NIST to include one or more mobile and/or IoT platforms in their evaluations, either directly or by working with the community to ensure that algorithms are evaluated in these contexts.

We are grateful to the NIST for the opportunity to comment on on this process. We look forward to working with NIST and the broader community to ensure that the Internet can be kept secure even if quantum cryptanalysis becomes feasible.

Respectfully submitted,

Richard Barnes, Firefox Security Lead
James Jones, Cryptographic Engineering Manager