Cryptography Standards in Quantum Time

- New wine in old wineskin?

Lidong Chen, NIST[1]

The National Institute of Standards and Technology (NIST) of the U.S. Government announced a call for proposals for quantum-resistant public-key cryptographic algorithms on December 15, 2016. The scope of the call covers all public-key cryptographic primitives currently standardized by NIST, which are public-key encryption, key agreement, and digital signatures schemes. The submission deadline is November 30, 2017.

It feels like just yesterday that we, as a community, were developing public-key cryptography standards in IEEE 1363 ([1]) and ASC X9 (financial services standards). The memory of the discussions at those development meetings is still fresh. Thanks to standardization, public-key cryptography schemes are deployed in every Internet router, computer, tablet, and cell phone, to enable secure communication and applications. Now that public-key cryptography schemes like Diffie-Hellman key agreement ([2]) and RSA digital signatures ([3]) have become indispensable for our digitized life, the recent progress made on quantum computers compels us to look for quantum-resistant counterparts.

Research on quantum-resistant public-key cryptography, also called post-quantum cryptography (PQC), has been fast-paced and very productive in recent years. Many newly developed schemes appear to be good candidates for the next generation of public-key cryptography standards. With almost three decades of experience and a mature deployment environment for public-key cryptography, will plugging in post-quantum cryptography in the existing applications be as easy as replacing a bulb? This article will discuss the challenges and opportunities in developing and deploying post-quantum cryptography standards.

History doesn't always repeat itself

When public-key cryptography was invented in the 1970s, people were fascinated by the idea of using number theory and finite fields to resolve key distribution problems. For thousands of years, enabling encryption had demanded a secret channel to distribute keys. Public-key cryptography allows communicating parties to establish a shared secret key without a secret channel. The concept of public-key cryptography also enables digital signatures for public authentication and authorization. In the beginning of the 1990s, revolutionary advances in computing technology and digital communications provided the commercial opportunity for public-key cryptography deployment. The Public-Key Cryptography Standards (PKCS) series were developed by RSA Laboratories as the first de facto standards ([4]). In particular, PKCS#1 provides the basic definitions of and recommendations for implementing RSA public-key cryptosystems. In 1994, the P1363 project was approved by Institute of Electric and Electronic Engineering (IEEE) to develop a public-

[1] The opinions expressed in this article are the author's own and do not necessarily represent the views of the National Institute of Standards and Technology (NIST).

key cryptography standard. About the same time, X9, a standards organization for financial services, also started to develop public-key cryptography standards in working group X9F1. The standards developed in IEEE P1363 and X9F1 focus on algorithm specifications for general usage. The Internet Engineering Task Force (IETF) was probably the first organization to standardize public-key cryptography for real applications, that is, Internet protocols. Internet Key Exchange (IKE) ([5]) and Transport Layer Security (TLS) ([6]) are two protocols where public-key cryptography is used for mutual authentication and key establishment.

In the early stage of standardization, the purpose was to make use of public-key cryptography in the emerging network for communication and commerce. Security notions and proofs were not as well developed as they are today. The ideas underlying the RSA and Diffie-Hellman schemes can be explained to people with a high school mathematics background. The relationship between the hardness of integer factorization and RSA, and between the hardness of the discrete logarithm problem (DLP) and Diffie-Hellman, are intuitive enough to be widely understood. Research at the time focused on the computing complexity of factorization and discrete logarithm computation. The theory focused on reduction proofs and the existence of (trapdoor) one-way functions, pseudorandom functions, etc. At that time, many details about securely implementing public-key cryptography were not understood. For example, the padding scheme in PKCS#1 has several versions. Some of the padding methods have security flaws. That is, the hardness of factorization cannot guarantee the security of the RSA scheme in practice unless every detail is handled properly. RSA-OAEP was proposed as a provably secure method for RSA encryption at Eurocrypt 1994 ([7]). RSA-OAEP was not simply a new way to randomize plaintext messages to hide every bit of the plaintext, it introduced the concept of non-malleable security against adaptively chosen ciphertext attacks (NM-CCA2, a.k.a. IND-CCA2) under the random oracle model. In the past two decades, more security notions have been established and used to prove security for a given cryptosystem. The rich theory of security models can certainly provide additional confidence in the provable security of new cryptography systems. On the other hand, how much to weigh provable security in selecting algorithms for standardization remains a challenge. For a given cryptographic scheme, if a provably secure version is less efficient than the one which seems secure but does not have a security proof, then shall the more efficient one or the one with proof be adopted? It turns out that, as security theories advance, the decision may be harder to make.

We also need to remember that efficiency in any particular computing environment had been a critical factor for adoption. In other words, a small advantage in performance may differentiate one algorithm from another. For example, being able to select small public-key sizes to speed up encryption and signature verification for RSA algorithms was considered a remarkable advantage. In the 1990s, great effort was made to improve the performance. Open source implementations were not available. On the other hand, attackers were also limited by computing capacity. The “Oakley group 1” for Internet Key Exchange (IKE) used a prime modulus of less than 800 bits in Diffie-Hellman key agreement, which is considered very weak with today’s discrete logarithm algorithms and computing power. Equally small integers were also used for RSA as moduli. Today, computing power has increased tremendously. Even though efficiency is still extremely important, for many of today’s implementation platforms, resource demands for implementing cryptography are not a major show-stopper. Furthermore, the most recent proposed post-quantum cryptography algorithms like lattice-based, coding-based, and multivariate cryptosystems all seem efficient enough to be plugged in to most environments where public-key cryptography has been implemented. Therefore, processing efficiency may not be the major competing factor to

differentiate algorithms. But for those very constrained devices and bandwidth limited networks, key size, signature size, and ciphertext expansion may become barriers for some applications.

The security and performance challenges we are facing today may be different from the time when public-key cryptography was first introduced into real-world applications. We will need to prepare to meet new challenges and resolve new problems.

Uncertainties

It has taken about a quarter of a century for us, as a community, to understand the currently deployed public-key cryptographic systems. On the journey to deploy them, we have learned to avoid security pitfalls step by step. We know there is no easy path. The NIST announcement to commence the process of standardizing post-quantum cryptography has raised some concerns. The major concerns are due to uncertainties.

The first uncertainty is whether now is the time to standardize new cryptographic schemes? We have not yet seen quantum computers which can crack RSA and Diffie-Hellman cryptographic schemes. The progress made on quantum computers is very promising, but we still cannot guarantee a timeframe with any certainty. If quantum computers do not appear along a predictable timeline, then when shall we make up our minds to move toward quantum-resistant cryptosystems?

In addition to the cost of replacing deployed cryptosystems with new schemes, is there a risk in deploying cryptographic systems which are so new? The second uncertainty is the classical security for the newly emerging post-quantum cryptography algorithms. Indeed, many proposed schemes have been broken in the past decade due to classical security flaws. Considering that it took so long for us to understand the security of the cryptosystems currently in use, it seems risky to move quickly to any of the new schemes.

The third uncertainty is probably the most worrisome one. The properties of quantum computers are much less well known than classical computers. Is it possible that new quantum algorithms will be discovered which can lead to new attacks on those algorithms supposed to be resistant to quantum attacks? Performance characteristics of future quantum computers, such as their cost, speed, and memory size, are also not well understood. This uncertainty results in differing opinions on quantum security strength levels for setting parameters and key sizes.

Most of Shakespeare's plays focus on the impossibility of certainty. Doesn't this also seem to be the case in the history of cryptography? We believed factorization and discrete logarithm are hard and then quantum computers and quantum algorithms emerged to shake our beliefs. As we move forward toward PQC standardization, the first step is to understand and work with the uncertainties.

There is much uncertainty about when quantum computers will be available at scale. It is certain that it will take years to develop and deploy new cryptographic standards. Considering that some data protection requirements are to remain confidential for many years, we need to make sure that quantum-resistant cryptographic algorithms are in place ahead of the time to guarantee backward secrecy. If it is as predicted by the experts, "a one-in-seven chance that some fundamental public-key crypto will be broken by quantum by 2026, and a one-in-two chance of the same by 2031", ([8]) then we do not have much time to complete the standardization and deployment process. For

cryptography, a one-in-seven chance to be broken is indeed “non-negligible”. Taking cautious action to start the process is the only option to deal with the uncertainty.

Classical security certainly shall be the first consideration for quantum-resistant cryptosystems. Some of the possible candidates are pretty new, whereas some were proposed years ago and their security has stood up pretty well. For example, the code-based McEliece encryption algorithm ([9]) was proposed in the 1970s, while NTRUencrypt ([10]) was proposed in the 1990s. Uncertainty about the classical security arises for the newer versions of these algorithms which improve the performance or key sizes. The fact that some new schemes have been broken quickly actually proves that the cryptanalysis capability of our community to evaluate classical security has grown strong, and can probably effectively identify the security flaws. Compared with 25 years ago, we are more certain about our cryptanalysis capabilities. Furthermore, an open and transparent process will allow cryptographers to conduct a thorough analysis and the community to assess the security of any newly proposed algorithm.

Indeed, our understanding of quantum security is far less comprehensive than our understanding of classical security. However, during last few years, significant progress has been made in understanding quantum security. The standardization process certainly will promote research on quantum algorithms and quantum security. As further progress is made in quantum computing, we will know more about quantum security. In dealing with this uncertainty, NIST proposes to use five equivalent security classes to select parameters and keys for each proposed algorithm ([11]). The general assumption is that there are no known quantum attacks on the proposed scheme (e.g. Shor’s attack on factorization) besides generic quantum speedup (e.g. Grover’s quadratic speedup on AES key search). The five security classes reflect not only classical security strength but also the different effectiveness of quantum speedups at the same classical security level. For example, if a scheme can provide 128-bit classical security and if there is no quantum attack other than classical attacks with generic quantum speedups like using Grover’s attack, then it is likely to be able to provide 64-bit quantum security. However, if quantum speedups are not as effective as Grover’s key search on AES-128, then it can provide a higher quantum security level than 64 bits. Note that breaking 64-bit quantum security could be significantly more difficult and expensive than breaking 64-bit classical security. To obtain precise estimations on quantum security of post-quantum cryptographic algorithms, it requires extensive collaboration between researchers in both the classical and quantum worlds, to foresee new progress on quantum attacks on any of the algorithms.

Nevertheless, the uncertainties urge us to start, because it will take time to understand the uncertainties and to resolve the unknowns. It is going to be a long journey.

Can we make PQC drop-in replacements?

Today, public-key cryptography is used everywhere. Introducing quantum-resistant counterparts involves a transition stage. Therefore, finding post-quantum cryptography algorithms which can be used as drop-in replacements will make the transition less disruptive. The question is, can we?

As we discussed above, processing complexity may not be a barrier any more, considering the fact that the most emerging post-quantum cryptography algorithms are pretty efficient in terms of processing time. However, we must prepare to deal with what we have not been used to. One

example is stateful hash-based signatures ([12]). Hash-based signatures were introduced as one-time signatures in the 1970s. Compared with other categories of post-quantum cryptography, the security of hash-based signatures is better understood. The major challenge is that since they are essentially one-time signatures, each private key can only be used once. Thus, the task of managing private keys, also called state management, becomes a major challenge for large-scale applications of hash-based signatures. To overcome the state management challenge, stateless hash-based signatures have been introduced ([13]). However, they have a much larger signature size. For bandwidth limited applications, the transmission of signatures may require segmenting the data into multiple messages in the existing protocols. Some post-quantum signature schemes, like the family of schemes based on multivariate cryptography, offer a signature size which is compatible with standardized signature schemes like ECDSA. But the public-key and private key sizes can be hundreds of times larger. As a result, a given quantum-resistant signature scheme can be a drop-in replacement in one application but may not be suitable for another application. That is, there is no one-for-all drop-in replacement.

Diffie-Hellman key agreement is a beautiful scheme in public-key cryptography for many reasons. Its first advantage is that it can provide perfect forward secrecy when ephemeral keys are used. The Perfect Forward Secrecy property can be described as follows: compromise of long-term keys does not compromise past session keys. It has become a very desirable property. In Transport Layer Security (TLS) as specified in IETF, version 1.2 and earlier versions, three key establishment schemes have been supported, RSA key transport, Static and Ephemeral Diffie-Hellman, and Ephemeral-Ephemeral Diffie-Hellman. In the newest version, TLS 1.3, Ephemeral-Ephemeral Diffie-Hellman is the only supported key establishment scheme. Therefore, it is on the top of the wish list to have a Diffie-Hellman quantum-resistant counterpart. Researchers have pursued along this direction. Currently, more than one Diffie-Hellman-like quantum-resistant key establishment scheme has been proposed and even prototyped. The properties are a little different, but in general are very close to Diffie-Hellman key agreement. A family of schemes based on the Ring-Learning-With-Errors (R-RWE) problem has been proposed for Diffie-Hellman-like key agreement. One of them is named “New Hope” ([14]). The proposed schemes are indeed as the name claims: a new hope for a drop-in replacement for Diffie-Hellman. The performance is quite reasonable. The difference is that operations are not symmetric for the two parties. Not only are the operations different, the responder needs to generate his message based on the initiator’s public value. The scheme has a possibility of failure even if both parties correctly select random values and conduct operations. It may not be a major concern for key establishment, but it can hardly be considered as exactly a drop-in replacement for Diffie-Hellman.

Another “new hope” for key agreement schemes is a family recently proposed based on isogenies between elliptic curves. The hardness of finding isogenies between supersingular elliptic curves was first introduced as the basis for cryptosystems more than 10 years ago by Charles, Goren, and Lauter in 2006 [15]. One of the versions is called Supersingular Isogeny Diffie-Hellman (SIDH) to emphasize the resemblance to Diffie-Hellman key agreement. Operationally it is more symmetric for the two parties. For those who are looking for drop-in Diffie-Hellman replacements, SIDH may look much closer to Diffie-Hellman key agreement. Performance-wise, it is significantly slower and more costly than ECDH ([16]). If performance cost is an issue for some applications or processors, then SIDH may not be suitable to replace Diffie-Hellman key agreement.

Furthermore, whether a new PQC standard can be used as a drop-in replacement may not depend completely on similarities in the key size, signature size, and the format. It may also rely on whether the security implementation technologies we have developed in the past are sufficient to assure security for post-quantum cryptography. We will see that some new implementation issues may require protocols and applications to introduce new mechanisms to guarantee secure implementation of the new schemes. That is, the existing protocols or applications may not provide sufficient countermeasures to deal with new issues.

Secure implementation issues for new algorithms

While developing and deploying the first generation of public-key cryptography standards, we learned a lot about dealing with secure implementation issues. For post-quantum cryptography, more implementation issues appear. The experience we gained before helps in our mental preparation to face the issues. But we need new techniques to deal with the new issues. Here are some examples.

Public-key validation

In discrete logarithm based cryptosystems, e.g. Diffie-Hellman key agreement, public-key validation is needed to assure that the public-key is in the right subgroup, because a small subgroup attack can force the established secret value into a small group which is vulnerable to exhaustive search. However, public- key validation is not straight-forward for all the new PQC algorithms. Some methods have been introduced to conduct indirect public-key validation. Some alternative indirect validation methods may have to require one party to reveal a function value of their secret key and jeopardize the security. Some other suggested methods may be very costly and not practical.

Public-key reuse

In a Diffie-Hellman key agreement scheme, a public key can be ephemeral or static. Even for ephemeral key Diffie-Hellman key agreement, some existing protocols allow the ephemeral key to be used in more than one execution. However, for some quantum-resistant key agreement schemes, if a public key is reused, the key can be compromised. The reuse can be intentional, by an attacker, or carelessly, by a legitimate party due to a bad key generation function. If key agreement with such a protocol is deployed, it must include mechanisms to prevent or reduce the security risk brought about by reusing a public key.

Decryption failure

In a public-key encryption scheme like RSA, if the involved parties follow the rules to select keys, parameters and if they conduct the operations properly, then the plaintext will be obtained through the decryption operation. Similarly, in a Diffie-Hellman key agreement scheme, if parties execute as each of them is specified, at the end, they will obtain the same secret value. However, in some of the newly emerging public-key encryption schemes, correctness of decryption or key agreement is not always the case. That is, even if every parameter and key is selected per the specification and each operation is executed properly, it is still possible that the plaintext cannot be obtained through a decryption or that two parties do not share the same secret value at the end of an execution of the scheme. This case is called decryption failure. The failure may happen with a relatively small probability. But it may introduce security flaws. Handling decryption failures is a new issue in security implementation.

Auxiliary functions

Usually public-key cryptography schemes include using certain auxiliary functions. For instance, hash functions are used as an auxiliary function for digital signatures. For some post-quantum cryptography schemes, new auxiliary functions are needed for secure implementation. Here is an example.

In the past, we have depended on the notion that a value can be selected uniformly at random from a properly sized set. A robust and secure random number generator is critical for the security of the implementation of many cryptosystems in use today. We have concentrated on ensuring correct implementations of random number generators to output uniformly distributed elements in a given set. Now, some of the new post–quantum cryptography algorithms require certain values to be selected according to a specific non-uniform distribution. For example, in the R-LWE based schemes like New Hope we mentioned before, the "error" value must be selected according to a Gaussian distribution. Simulating these required distributions requires introducing new auxiliary functions. Since the security of the implementation relies on properly selected values with the required distribution, the simulation function is critical.

Along the path to deploy the new cryptosystems, we certainly will find new issues. Some of the security implementation issues for post-quantum cryptography may not be new. For example, counter-measures for side-channel attacks have been implemented for the cryptosystems currently in use. But we may need new methods and techniques to protect a given new algorithm from side-channel attacks. It is also noticeable that implementing the countermeasures to deal with security issues may increase processing and communicating complexity for a given scheme. Therefore, understanding the tradeoffs is critical in making right decisions.

PQC Standardization – Road ahead

For PQC standardization, we may not have what we wish for. That is, the new schemes may not be able to be introduced as drop-in replacements, and we have new issues to deal with to ensure secure implementations of new schemes. Even so, we have reason to be optimistic about the road ahead. First, cryptographic research has advanced tremendously compared with 25 years ago. The security notions and proofs which have been developed will certainly help us to have a better understanding about the security of a scheme. The research community has already demonstrated a strong capability to conduct effective cryptanalysis on schemes to be deployed. Secondly, the applications community has become more mature. Public-key cryptography has been implemented in many communication protocols and digital devices. The applications community has gained extensive experience deploying new cryptographic algorithms. Furthermore, open source implementations have become available for most cryptographic algorithms in use. The open source implementations have promoted collaboration and been a collective effort to ensure best practice. Finally, advanced computing and communication technologies can accommodate cryptographic functions which are more demanding of processing and communication resources.

Although we are confident to overcome the challenges arising while standardizing post-quantum cryptography, we indeed will need new strategies to deal with new situations in developing the next generation cryptography standards.

To overcome the challenge that no exact drop-in replacements have been proposed for currently deployed cryptosystems, future standards may specify multiple algorithms for each cryptographic primitive according to the requirements of different applications, especially to deal with some non-ideal characteristics such as large signature size or large keys. These algorithms can be selected from different categories and based on different hard problems. The reason for doing so is that signature size or key size may not be a problem for some applications, but may be truly a show-stopper for others. In this way, the standards may allow different applications to deploy different algorithms. On the other hand, the existing protocols may need to be modified to handle larger signatures or key size, for example, through segmentation of messages. For new applications, implementations must keep the demands of post-quantum cryptography in mind and allow the new schemes to adapt to them. The requirements for post-quantum cryptography may shape the future application standards.

Secure implementation issues can be addressed through different approaches. Efforts have been made to reduce the probability of decryption failure through justifying the parameters and keys, such as the techniques used in NTRUencrypt. One can also add mechanisms at the protocol level to limit security flaws.

In conclusion, for post-quantum cryptography standardization, we need new wineskin to hold the new wine. Plugging in quantum-resistant cryptosystems to existing applications will be a new experience for both cryptographers and practitioners. We have accumulated valuable experience in the past 25 years working on first generation public-key cryptography standards which will help us to deal with the new issues and challenges.

References

1. IEEE P1363 Standard Specifications for Public-Key Cryptography http://grouper.ieee.org/groups/1363/
2. W. Diffie and M. Hellman, “New Directions in Cryptography”, IEEE Transactions on Information Theory. 22 (6): 1976, pp. 644–654
3. R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", Communications of the ACM. 21 (2): 1978, pp. 120–126.
4. PKCS#1 v2.2: RSA Cryptography Standard, RSA Laboratories, October 27, 2012
5. C. Kaufman, P. Hoffman, Y. Nir, P. Eronen and T. T. Kivinen, “Internet Key Exchange Protocol Version 2 (IKEv2)”, IETF RFC 7296, October 2014, www.ietf.org
6. T. Dierks and E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.2”, IETF RFC 5246, August 2008, www.ietf.org
7. M. Bellare, P. Rogaway, “Optimal Asymmetric Encryption – How to encrypt with RSA.” Advances in Cryptology - Eurocrypt '94 Proceedings, Lecture Notes in Computer Science Vol. 950, Springer-Verlag, 1995.
8. M. Mosca, Cybersecurity in an era with quantum computers: will we be ready? IACR Cryptology ePrint Archive Report 2015/1075, 2015. http://eprint.iacr.org/2015/1075
9. R. J. McEliece, "A Public-Key Cryptosystem Based On Algebraic Coding Theory" . DSN Progress Report. **44**: 114–116 (1978).
10. J. Hoffstein, J. Pipher, J.H. Silverman, NTRU: A ring-based public key cryptosystem. In: Buhler J.P. (eds) Algorithmic Number Theory. ANTS 1998. Lecture Notes in Computer Science, vol 1423. Springer, Berlin, Heidelberg.

11. NIST “Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process” December 15, 2016 http://csrc.nist.gov/groups/ST/post-quantum-crypto/documents/call-for-proposals-final-dec-2016.pdf
12. J. Buchmann, E. Dahmen, and A. Hűlsing, “XMSS - A Practical Forward Secure Signature Scheme based on Minimal Security Assumptions” Proceedings of PQCrypto 2011, Lecture Notes in Computer Science Vol. 7071, Springer, 2011.
13. D. Bernstein et al “SPHINCS: practical stateless hash-based signatures”, Proceedings of Eurocrypt 2015, Lecture Notes in Computer Science Vol. 9056, Springer, 2015.
14. E. Alkim, L. Ducas, T. Pöppelmann and P. Schwabe, “Post-quantum key exchange – a new hope”, proceedings of USENIX Security 2016.
15. D. X. Charles, K. E. Lauter, E. Z. Goren, “Cryptographic Hash Functions from Expander Graphs”, Journal of Cryptology January 2009, Volume 22, Issue 1, pp 93–113.
16. C. Costello, P. Longa, and M. Naehrig, “Efficient algorithms for supersingular isogeny Diffie-Hellman” Proceedings of Crypto 2016, Lecture Notes in Computer Science Vol. 9814, Springer, 2016.