Reviewing Crypto-Agility and Quantum Resistance in the Light of Agile Practices

Lodovica Marchesi(✉), Michele Marchesi, and Roberto Tonelli

Department of Mathematics and Computer Science, University of Cagliari, Cagliari, Italy
{lodovica.marchesi,marchesi,roberto.tonelli}@unica.it

Abstract. The term crypto-agility means the ability to quickly and securely change cryptographic algorithms and related data, in the case of their compromise. In this context, the advent of quantum computing constitutes a new paradigm, which poses existential threats to current cryptographic algorithms. Even if these attacks are not an imminent danger, we must be prepared to change the cryptographic algorithms at risk with new, quantum resistant ones. This is by no means an easy task, because cryptographic algorithms are used everywhere and are often also implemented on the hardware. In this paper, we analyze the similarities and the differences between traditional agility and crypto-agility, and investigate the prospects of using agile and lean practices in the context of crypto-agility to introduce quantum resistant algorithms. In particular, for the main agile and lean practices we discuss if and how they can be useful for obtaining crypto-agility. We also investigate how the features key to crypto-agility can be helped by the agile and lean approach.

Keywords: Agile methods · Cryptographic agility · Encryption algorithms · Quantum resistance

1 Introduction

The introduction of agile methodologies in the late 1990s and the term "agile" itself in the 2001 Agile Manifesto had a deep impact on software engineering and computer science [1]. In a short time, "agile" became a buzzword used wherever you had to emphasize the ability to respond quickly and well to challenges and requirement changes.

A few years after the Agile Manifesto, people in the cybersecurity community introduced the term "cryptographic agility" or "crypto-agility". This term was defined first in a paper by LaMacchia and Manferdelli in 2006 [9]. Their paper presented the Microsoft's new core cryptographic API, "Crypto Next Generation" (CNG), which was claimed to have "cryptographic agility". In the context of CNG, crypto-agility means the ability to change its cryptographic algorithms (CAs) and related data (setting parameters, key storage, etc.) in the case of their compromise, in a quick a secure way.

P. Kruchten and P. Gregory (Eds.): XP 2022/2023 Workshops, LNBIP 489, pp. 213–221, 2024.
https://doi.org/10.1007/978-3-031-48550-3_21

In fact, practical cases in which CAs had to be changed quickly to avoid damage due to their compromise are very rare. However, the advent of quantum computing constitutes a new paradigm which poses new challenges to cryptography because robust quantum computers (QCs) have been proven to be able to break several important CAs currently used. Important milestones still remain before a marketable and truly usable QC can exist to solve real-world problems. The most optimistic experts estimate that it will take 5 to 10 years from now to build a viable QC. The more cautious ones predict 15 to 30 years [13].

We know that agility means the ability to react quickly to changes. Agile and lean software development were introduced to support traditional programming in projects where timing is essential. Crypto-agility for quantum resistance (QR), on the other hand, typically deals with changes that likely will be mandatory at the end of this decade, thus operating in time intervals completely different.

In this paper, we analyze the links, the similarities, and the differences between crypto-agility and traditional agility. We discuss which agile and lean principles and practices are still relevant to cryptographers who are in charge of finding and substituting traditional algorithms with new quantum-resistant ones and which are not relevant to them.

The rest of the paper is organized as follows: Sect. 2 provides a short review of the literature; Sect. 3 presents the concept of crypto-agility, highlighting its main properties, while Sect. 4 explains what QC is and what QR means. In Sect. 5 we describe the relationship between crypto-agility for QR with agile and lean practices, to dive into how agile software development can help the development, testing, and deployment of new quantum resistant algorithms.

2 Related Work

The first use of the term "crypto-agility" was made in 2006 in [9]. After that date, the term compares in technical reports of standard working groups (IETF, ETSI, NIST). Only from 2018 onwards, also due to the efforts to choose and standardize quantum resistant algorithms, which were announced at PQCrypto conference in 2016, the term has been used and discussed in many forums. The related wikipedia page also dates back to November 2018.

In 2019 Grote et al. described the strategy of post-quantum cryptography and crypto-agility. They reviewed the proposed quantum resistant algorithms and highlighted the need for crypto-agility to effectively replace non-quantum resistant CAs with quantum resistant ones [5]. In 2021 Mashatand and Heintzman recommended a crypto-agile process to assess and mitigate the exposure of organizations to quantum attacks. The proposed process includes steps such as *Determine Transition Path*, *Wait for Standardization*, *Invest in Crypto-agility, Establish and Maintain a Quantum-Resistance Roadmap*, *Implement Hybrid Cryptography* [11]. In the same year, Ma et al. proposed CARAF: Crypto Agility Risk Assessment Framework [10]. CARAF is aimed at analyzing and evaluating the risk that results from the lack of crypto agility, in order to determine an appropriate mitigation strategy commensurate with the risk tolerance. The application

of this framework was demonstrated with a case study regarding QR. Furthermore, Zhang and Miranskyy analyzed the threats posed by QCs and compared the related mitigation strategies to those used to address the Y2K bug [18]. They proposed a road map for software developers to address encryption-related challenges associated with quantum attacks, especially using crypto-agility. In 2022, Holm et al. proposed the Crypto-Agility Maturity Model (CAMM) to determine the state of crypto-agility of a given software or IT landscape and to improve it [7]. CAMM consists of five levels named: (0) Initial/Not possible, (1) Possible, (2) Prepared, (3) Practiced, and (4) Sophisticated. For each level, a set of requirements is formulated based on literature review. Initial feedback from field experts confirmed that CAMM has a well-designed structure and is easy to understand.

2.1 Agility and Cybersecurity

There are several studies on how agile practices can be made compatible with security assurance, starting from the seminal work of Bezsonov [2] on merging XP and security practices. Among these studies, two main areas emerge: updates to Scrum and in general to agile processes to manage security aspects, and user stories to specify security requirements.

Regarding the first area, Fitzgerald et al. identified the main incompatibility issues between agile characteristics and constraints imposed by regulated environments and illustrated through a detailed case study how an agile approach can be implemented successfully in a regulated environment [3]. Ghani et al. suggested adding Security Backlog and the role of a Security Master in Scrum [4]. Othmane et al. proposed a method to ensure the security of software increments, integrating security engineering activities into the agile software development process [14].

Regarding security requirements, Kongsli proposes the concept of "misuse story", derived from the "misuse case", and reports on experiences with the use of misuse stories and automatic security tests in the development of web applications [8]. Williams et al. suggested the Protection Poker game to find the relative security risk of each requirement [17]. For a recent and comprehensive survey of security in agile software development, we refer the reader to the work of Rindell et al. [15].

In all cited papers, the relationships between the agile approach and general cybersecurity are considered, including protection against multiple forms of cyber attacks. In this paper, we are interested to practices related to the development and integration of cryptographic methods and tools into software systems, which is only one of the several aspects of cybersecurity. We deal with security requirements of cryptographic tools related to asymmetric cryptography, document encryption and decryption, digital signatures, key storage and the like.

3 Definition of Crypto-Agility

Crypto-agility is strictly related to cybersecurity, but its scope is much narrower. As reported above, it regards the ability to easily change the algorithm implementations used by cryptographic protocols and to provide a high level of abstraction by the API for core cryptographic operations [9].

The key properties of crypto-agility were provided by Mehrez and El Omri [12]. They state that *"crypto-agility is the ability of a system to migrate easily from one CA to another, in a way that is flexible, scalable, and dynamic"*. The most important crypto-agility properties among those reported by them are:

- **Extensibility**: ability to add new algorithms or new parameters to the system as efficiently as possible.
- **Removability**: ability to gracefully retire cryptographic systems that have become vulnerable or obsolete.
- **Fungibility**: ease to swap security components; also, the ability for machines to select their security algorithms in real time and based on their combined security functions;
- **Interoperability**: Crypto-Agility solutions must be interoperable between independent implementations based purely on the information provided in the specification.
- **Updateability**: support of secure updates or patches of CAs in the system.
- **Compatibility**: if we replace software on a system, the new software modules and patches should be able to operate on the same hardware.
- **Reversibility**: if any software update fails, the system should be able to return to the previous working software version.

The object of crypto-agility is the development or updating of software systems that use CAs. These systems are typically used for secure data transmission and for the storage and retrieval of encrypted data. Another field impacted by the change in CAs is that based on a blockchain, which makes extensive use of asymmetric cryptography.

The CAs which are actually used follow standards enacted by various organizations, among which the most prominent is NIST. There are standard CAs for symmetric and asymmetric encryption, hash functions, key management. The standards include detailed specification of CA libraries API, so that systems using different libraries can exchange encrypted data, provided that these libraries follow the API specifications.

Typically, large, regulated organizations are more impacted by changes in CAs than small ones. Since CAs are largely regulated, including their APIs, the software that actually needs to be written is the software that uses these CAs to encrypt (maybe using more than one CA), send or store data, and decipher them.

4 Quantum Computing and Quantum Resistance

Quantum computing represents a significant breakthrough in computer science and will have a strong impact on many fields, such as science, finance, artificial intelligence, pharmacology, and many others. Unfortunately, it also has the power to breach current cryptography systems, among which secure Internet communications, digital signatures, digital currencies, and digital ledger technology (DLT). The cryptography used in these systems is composed of *Hash functions*, which guarantee immutability of data, and in blockchains are also applied in proof of work; *Symmetric cryptography*, used to encode and decode information that must remain confidential; *Asymmetric cryptography*, which is behind SSL/TLS protocol ensuring secure Internet communication, and in DLT guarantees the property of the assets linked to an address.

No classic computer in existence is capable of performing calculations fast enough to reverse this math in any usable time frame. However, the advent of QC constitutes a new paradigm which poses new challenges to cryptography. Important milestones still remain before a marketable and truly usable QC can exist to solve real-world problems. Experts estimate that it will take 5 to 30 years from now to build a viable QC [13].

When QCs will become operational, the currently understood menace they will pose to cryptography is based on Shor's algorithm for quickly factoring the product of two very large primes [16]. This algorithm will allow one to unhinge the RSA algorithm, and with a small variant also the ECDSA algorithm. These algorithms are currently the most used for asymmetric cryptography. Today, a digital computer that uses the most efficient algorithm known would carry out the factoring of a number of 300 digits in about 150,000 years. A QC using the Shor algorithm would find the solution in seconds.

Another threat is Grover's algorithm, which allows you to speed up the search for possible solutions to unhinge symmetric encryption algorithms (AES, DES, hash algorithms) and can reduce the difficulty of the problem from n to $\sqrt{n}$ [6]. However, the performance of Grover's algorithms is not as innovative and dangerous as for Shor's one because it can be easily countered by increasing the length of the encryption key.

Luckily, several CAs that are quantum resistant already existed in the nineties, and more have been introduced after the discoveries of Shor and Grover. Standardization bodies such as NIST (National Institute of Standards and Technology) and ETSI (European Telecommunications Standards Institute) have been working on standard quantum resistant algorithms for almost a decade, and a set of international standards is expected in 2025 [11].

One of the main issues faced by organizations to be prepared for quantum menace, is the fact that CAs are ubiquitous in the software and hardware systems used. Therefore, migrating to quantum resistant solutions will not be an easy journey. To this end, an organization can take advantage of a process that exhibits the properties of crypto-agility as defined in Sect. 3.

5 Can Agility Help Crypto-Agility?

Agile and Lean software development was introduced to shorten development times and accommodate changes, without compromising on quality. Its time-frames, depending on the specific activities, vary from hours to days or weeks.

Mimicking the well-known approach to software development, crypto-agility means the ability to react to CA changes effectively and timely. However, the need to protect a system from quantum attacks likely will be mandatory not before the end of this decade, thus its time-frame is of the order of years, or even decades.

We asked ourselves if agile principles and practices can be successfully used in the context of crypto-agility, and specifically to effectively address the development and adoption of quantum resistant software. To answer this question, we consulted some software practitioners working in the field of cybersecurity, and also took advantage of our experience in studying QR for blockchain applications. The research questions asked were: (i) "How suitable are the agile and lean practices to support crypto-agility?"; and conversely (ii) "How crypto-agility features can benefit from the agile approach?".

The context is the development of software that must strictly follow standards regarding its CA and API, and that uses standard libraries to send, store, and receive encrypted data, to decode them, to manage keys, and to combine CAs to obtain stronger security or to manage digital signatures and secure ownership of digital assets. Following a crypto-agile principle, this software should also be able to automatically choose the "right" CA, and manage the updating of CA libraries.

Table 1 reports the main Agile and Lean practices, with a judgment on their relevance to crypto-agility. You can see that most of these practices were deemed to be useful, or very useful. The requirements should be expressed as features, because most of them do not regard direct interaction with an user. Automated testing is of the utmost importance. Tests can also be defined before coding the CA and other software. We stress that the number of tests needed to assess the security of the developed software is much higher than in other systems.

Regarding Lean practices, most of them are very useful to apply also to this type of development. The "optimize the whole" practice may not be relevant to this kind of development because the standardized CAs are already defined. However, the choice of which specific quantum resistant CA to use should be carefully made because these CA have memory and CPU requirements much higher than traditional ones. The WIP limitation can, of course, be applied, but here the need of a continuous flow of delivered working features is uncommon, so this practice is not very relevant in most cases.

Regarding the second question "How crypto-agility can benefit from agility?", Table 2 summarizes the results of our study. For the sake of brevity, we are not able to discuss these results, but they are quite self-explanatory.

Table 1. Suitability of Agile and Lean practices for Crypto-Agility to obtain QR.

Agile practice	Suitability to Crypto-Agility	Agile practice	Suitability to Crypto-Agility
Iterations	Useful, on a longer timescale	Simple design	Useful, though not always applicable to algorithms
Customer-oriented approach	Very useful	Refactoring	Very useful
Product backlog	Very useful	Coding standards	Very useful
User stories or features (MMF)	See discussion	Collective code ownership	Each artifact should have an accountable owner
Agile roles	Useful, a Security Master can be added	**Lean practice**	**Suitability to Crypto-Agility**
Timeboxing	Useful, on a longer timescale	Eliminate waste	Useful, on a longer timescale
Scrum daily meetings	Useful	Build quality in	Very useful
Sprint meetings	Useful to plan and assess Sprint work	Create knowledge	Very useful
Retrospective meetings	Very useful to steer the pro-ject and upgrade the process	Defer commitment	Very useful
Daily integration	Frequent integration on a longer time scale (not daily)	Deliver fast	Useful, on a longer timescale
Test-driven development (TDD)	See discussion	Respect people	Very useful
Automated tests	Very useful	Optimize the whole	See discussion
Burndown chart	Useful, on a longer timescale	Visualize the Worflow	Very useful
Requirement prioritization	Very useful	Limit Work in Progress (WIP)	See discussion
Pair programming	Maybe for critical tasks	Cumulative flow diagram	Not very relevant

Table 2. Crypto-Agility principles and agile practices

Crypto-Agility Principles and Features	Relationship with Agility and notes	Source
Extensibility: ability to easily change the algorithm implementations used by cryptographic protocols. Non quantum resistant cryptographic algorithms will be replaced QR proved algorithms.	Agility is aimed to promote change, so most agile practices are key to obtain this principle.	[5], [9], [12]
Protocols with a high level of abstraction of the API and modular implementations to easily accommodate the insertion/updating/interoperability of new algorithms.	This feature allows to follow Updateability and Interoperability principles. It requires good design capabilites, so it is linked with the agile and lean practices of simplicity, continuous testing, build quality in, and create knowledge.	[9], [12]
Ability to adjust to select the security algorithms in real time without any or with as little as possible human intervention.	This is Fungibility feature. It is a requirement linked to good design, as above.	[7], [12]
Removability, ability to easily retire cryptographic systems that have become either vulnerable or obsolete.	It is a requirement linked to good design, as above.	[12]
Testing and validation should support all steps of cryptographic processes from implementation to roll out	Clearly linked to the agile practice of continuous, automated testing, and of refactoring. The number of needed tests is much higher than usual.	[7]
Compatibility, if cryptographic software is replaced with new software modules, it should be able to operate smoothly on the same hardware.	Linked to good design, modularity and continuous testing.	[12]
Reversibility, if any software update is not successful, the system should be able to return to the previous working software version.	This feature requires the use of a configuration management system, which is key also for continuous integration and testing.	[12]

6 Conclusions

The advent of QC bring about new risks for cryptographic algorithms, which we may have to deal with within a decade. Being prepared to quickly and safely transform into quantum resistant algorithms is not a simple task, also due to the wide diffusion of cryptographic algorithms both at the hardware and software level. In this article, we have analyzed the similarities and differences between traditional agility and crypto-agility. We investigated whether and to what extent leading agile and lean practices are suited to support crypto-agility. Most have been rated useful or very useful, with an emphasis on automated and massive testing.

This work is part of a broader involvement of our research group in the field of quantum software engineering.

References

1. Beck, K., Beedle, M., Van Bennekum, A., et al.: The agile manifesto (2001)
2. Beznosov, K.: Extreme security engineering: on employing XP practices to achieve' good enough security' without defining it (2003)
3. Fitzgerald, B., Stol, K.J., O'Sullivan, R., O'Brien, D.: Scaling agile methods to regulated environments: an industry case study (2013)
4. Ghani, I., Azham, Z., Jeong, S.R.: Integrating software security into agile-scrum method. Trans. Internet Inf. Syst. **8**(2), 646–663 (2014)
5. Grote, O., Ahrens, A., Benavente-Peces, C.: Paradigm of post-quantum cryptography and crypto-agility: strategy approach of quantum-safe techniques (2019)
6. Grover, L.K.: Quantum mechanics helps in searching for a needle in a haystack. Phys. Rev. Lett. **79**, 325–328 (1997)
7. Hohm, J., Heinemann, A., Wiesmaier, A.: Towards a maturity model for crypto-agility assessment. arXiv preprint arXiv:2202.07645 (2022)
8. Kongsli, V.: Towards agile security in web applications (2006)
9. LaMacchia, B.A., Manferdelli, J.L.: New vistas in elliptic curve cryptography. Inf. Secur. Tech. Rep. **11**(4), 186–192 (2006)
10. Ma, C., Colon, L., Dera, J., Rashidi, B., Garg, V.: CARAF: crypto agility risk assessment framework. J. Cybersecur. **7**(1) (2021)
11. Mashatan, A., Heintzman, D.: The complex path to quantum resistance: is your organization prepared? Queue **19**(2), 65–92 (2021)
12. Mehrez, H.A., El Omri, O.: The crypto-agility properties (2018)
13. Mosca, M.: Cybersecurity in an era with quantum computers: will we be ready? IEEE Secur. Priv. **16**(5), 38–41 (2018)
14. Othmane, L.B., Angin, P., Weffers, H., Bhargava, B.: Extending the agile development process to develop acceptably secure software. IEEE Trans. Dependable Secure Comput. **11**(6), 497–509 (2014)
15. Rindell, K., Ruohonen, J., Holvitie, J., Hyrynsalmi, S., Leppänen, V.: Security in agile software development: a practitioner survey. Inf. Softw. Technol. **131**, 106488 (2021)
16. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Rev. **41**(2), 303–332 (1999)

17. Williams, L., Meneely, A., Shipley, G.: Protection poker: the new software security "game". IEEE Secur. Priv. **8**(3), 14–20 (2010)
18. Zhang, L., Miranskyy, A., Rjaibi, W.: Quantum advantage and the Y2K bug: a comparison. IEEE Softw. **38**(2), 80–87 (2021)