The special case of data protection and self-adaptation*

Zoltán Ádám Mann and Andreas Metzger

paluno – The Ruhr Institute for Software Technology
University of Duisburg-Essen
Essen, Germany

Abstract

In this extended abstract, we consider one important aspect of security: the protection of sensitive data from unauthorized access. We argue that (i) self-adaptation may facilitate the efficient protection of sensitive data; (ii) data protection has peculiar properties that make its treatment different from other quality attributes; and (iii) data protection should be considered in combination with other quality attributes like performance and costs.

Keywords: self-adaptive systems, data protection, privacy, security

1 The case for adaptive data protection

Our society increasingly relies on capturing, transferring, storing, and processing ever growing amounts of data [4]. Some of those data may be sensitive, e.g., because it is personal data, the disclosure of which would constitute a violation of privacy, or because it is a business secret. There is growing public concern about the protection of sensitive data, also mirrored by emerging relevant legislation, like the General Data Protection Regulation (GDPR) of the European Union. The GDPR increases the breadth and depth of control that data subjects have about their data.

There are well-known security mechanisms for protecting sensitive data: encryption, secure hardware, access control etc. However, all these mechanisms have certain drawbacks or limitations [12], e.g., performance overhead in the case of encryption. Therefore, which kind of security mechanisms to apply and when should be carefully considered.

Modern computing paradigms – distributed systems, cloud computing, the Internet of Things, or fog computing – exhibit specific properties that make data protection challenging. First, such systems are complex, with rich interactions among many components and actors. Second, such systems are dynamic: the components and actors, as well as their interactions, may change at run time [6]. As a result, the circumstances that are important for data protection may also change at run time. E.g., software components that store or process sensitive data may be migrated between data centres located in different countries, which may have different data protection levels and different legislations [10]. Another example is that servers with special hardware security properties may become available or unavailable [5].

To cope with changing circumstances at run time, self-adaptation is a promising approach. Relating to data protection, self-adaptation offers several advantages:

- Possibility to react to new threats. For example, if a software component handling sensitive data is migrated to a non-secure data centre, encryption can be activated to sustain the desired level of data protection.
- Possibility to react to changes in the available security mechanisms. E.g., if a server with special security features becomes available in the cloud, it can be exploited for improved data protection.
- Avoiding unnecessary overheads. E.g., if two components that exchange sensitive data are migrated to the same server, encryption of the data exchange can be switched off since the data exchange does not travel across the network anymore.

*Published in the *Proceedings of the 13th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2018)*, pp. 190-191, 2018

2 Special characteristics of data protection

To apply self-adaptation to data protection, three important characteristics have to be addressed that differentiate data protection from many other quality goals: (i) lack of simple mathematical models to specify and measure data protection, (ii) non-localizability, i.e., impossibility to determine and assure data protection levels with local observations respectively modifications, and (iii) the need for perfection. We elaborate on these aspects next.

Although some other quality goals may also possess one or the other of these characteristics, their combination seems to be unique.

While the characteristics are not specific to self-adaptation, they make self-adaptation for data protection challenging. In particular, each of the three characteristics prevents the application of some known approaches to self-adaptation. The combination of the properties makes it especially difficult to apply approaches that were designed with other quality goals in mind.

2.1 Lack of simple mathematical models

Some quality goals, like costs or performance, are easy to quantify. Several existing approaches to self-adaptation need quantified goals [3, 2, 7], which is a difficulty for data protection. Of course, some way of quantifying data protection is possible. For example, the estimated risk of violating data protection policies may be quantified using risk assessment techniques. However, such quantification may not be sufficient, because many existing approaches require that the targeted quality goal follows some simple mathematical model.

Some existing approaches presume that the quality goal can be *monotonously* controlled by an appropriate parameter [2]. E.g., in a cloud computing setting, the number of servers used for a load-balanced service is a parameter that controls both performance and costs monotonously: increasing the number of servers improves performance but increases costs.

Other existing approaches presume that the quality goal is *additive*. Continuing the cloud example, the total cost is the sum of the costs of the used servers, and the same holds for energy consumption as well. Another example is that the total response time of a composite service may be estimated as the sum of the response times of the atomic services used on the longest path [3].

We are not aware of a metric that quantifies data protection well, and also has one of these simple mathematical properties.

2.2 Non-localizability

By non-localizability, we mean that in general, (i) the level of data protection cannot be determined based on local observations and (ii) data protection cannot be assured using local actions.

Rather, data protection requires consorted efforts from all components of a system and its socio-technical context, including hardware, software, processes, and humans. E.g., special hardware security features alone do not guarantee data protection: also the software must be able to exploit those hardware features to ensure data protection [5]. Data protection, or the lack thereof, often relates to interconnections among entities [8], the topology of the interconnections [9], or specific interconnection patterns [11].

In contrast, many existing approaches to self-adaptation for other quality goals assume that local observations suffice to determine the satisfaction of the quality requirements, and local changes suffice to recover from requirement violations. E.g., monitoring server load is sufficient to detect performance problems of a web application, and disabling optional content in dynamically generated web pages may be sufficient to cope with performance problems [7].

2.3 Need for perfection

Many existing approaches to self-adaptation work in a best-effort manner: they try to achieve requirement satisfaction as much as possible, but cannot guarantee that requirements will be *always* satisfied. It is challenging to give formal guarantees for self-adaptive systems [1], but also approaches with rigorously proven properties guarantee only that, under suitable conditions, the system will eventually reach a state that satisfies the requirements [2], not that it would always be in such a state.

For other quality goals, this is often not a problem. E.g., if the response time of a web application sometimes goes above the specified threshold for a short time, this may be acceptable. If the energy consumption of a system is higher than desired, this is also acceptable in most cases.

However, a violation of data protection goals, even if only for a short time, may allow unauthorized parties to get access to and exploit sensitive data. This is an irreversible problem: even if the system goes back to a state in which data protection requirements are satisfied, the damage cannot be undone. Depending on the nature of the

data and the malicious intents of the unauthorized party, the one-time data breach may have serious financial, legal, or reputation consequences.

Therefore, data protection requirements should be satisfied throughout system operation, not only most of the time. Moreover, since it is sufficient for an attacker to attack the "weakest link in the chain" to breach security, it is important to keep all components in a secure state throughout system operation.

3 Interplay with other goals

Although we have argued in Section 2 that data protection has some properties that differentiate it from other quality goals, it should be pointed out that data protection should not be handled in isolation, but rather together with the other quality goals. This is important because the techniques for ensuring data protection impact also other quality goals like performance and costs.

For example, if performance and costs were not considered, one could always use fully-homomorphic encryption and special secure hardware. These techniques, however, introduce large overhead and costs and may not always be needed [12]. Therefore, the aim should be to find the best trade-off that ensures data protection with minimal impact on other quality goals. This is an important goal for further research.

Acknowledgments. This work was partially supported by the European Union's Horizon 2020 research and innovation programme under grant 731678 (RestAssured).

References

[1] Rogério de Lemos, David Garlan, Carlo Ghezzi, Holger Giese, Jesper Andersson, Marin Litoiu, Bradley Schmerl, Danny Weyns, Luciano Baresi, and Nelly Bencomo. Software engineering for self-adaptive systems: Research challenges in the provision of assurances. In *Software Engineering for Self-Adaptive Systems III*. 2017.

[2] Antonio Filieri, Henry Hoffmann, and Martina Maggio. Automated multi-objective control for self-adaptive software design. In *Proceedings of the 10th Joint Meeting on Foundations of Software Engineering*, pages 13–24, 2015.

[3] Carlo Ghezzi, Leandro Sales Pinto, Paola Spoletini, and Giordano Tamburrelli. Managing non-functional uncertainty via model-driven adaptivity. In *Proceedings of the 35th International Conference on Software Engineering*, pages 33–42, 2013.

[4] Meiko Jensen. Challenges of privacy protection in big data analytics. In *IEEE International Congress on Big Data*, pages 235–238. IEEE, 2013.

[5] Zoltán Ádám Mann and Andreas Metzger. Optimized cloud deployment of multi-tenant software considering data protection concerns. In *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pages 609–618. IEEE Press, 2017.

[6] Zoltán Ádám Mann, Andreas Metzger, and Stefan Schoenen. Towards a run-time model for data protection in the cloud. In *Modellierung 2018*, pages 71–86, 2018.

[7] Gabriel A. Moreno, Javier Cámara, David Garlan, and Bradley Schmerl. Efficient decision-making under uncertainty for proactive self-adaptation. In *IEEE International Conference on Autonomic Computing*, pages 147–156, 2016.

[8] Inah Omoronyia, Luca Cavallaro, Mazeiar Salehie, Liliana Pasquale, and Bashar Nuseibeh. Engineering adaptive privacy: on the role of privacy awareness requirements. In *Proceedings of the 2013 International Conference on Software Engineering*, pages 632–641. IEEE Press, 2013.

[9] Liliana Pasquale, Carlo Ghezzi, Claudio Menghi, Christos Tsigkanos, and Bashar Nuseibeh. Topology aware adaptive security. In *Proceedings of the 9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, pages 43–48. ACM, 2014.

[10] Eric Schmieders, Andreas Metzger, and Klaus Pohl. Runtime model-based privacy checks of big data cloud services. In *International Conference on Service-Oriented Computing*, pages 71–86. Springer, 2015.

[11] Stefan Schoenen, Zoltán Ádám Mann, and Andreas Metzger. Using risk patterns to identify violations of data protection policies in cloud systems. In *13th Intl. Workshop on Engineering Service-Oriented Applications and Cloud Services*, 2017.

[12] Marten Van Dijk and Ari Juels. On the impossibility of cryptography alone for privacy-preserving cloud computing. In *Proceedings of the 5th USENIX Conference on Hot Topics in Security*, pages 1–8, 2010.