S. Hahner, "Architectural Access Control Policy Refinement and Verification under Uncertainty," in
Companion Proceedings of the 15th European Conference on Software Architecture (ECSA-C), 2021

Architectural Access Control Policy Refinement and Verification under Uncertainty

Sebastian Hahner

KASTEL – Institute of Information Security and Dependability, Karlsruhe Institute of Technology (KIT), Germany

Abstract

In our connected world, confidentiality is a central quality requirement. A commonly used mechanism to meet confidentiality requirements is access control. However, access control policies are usually not defined on the architectural abstraction level and are imprecise during design time due to the high degree of uncertainty. This impedes early considerations of confidentiality as implied by "Privacy by Design". We propose an approach to refine and verify access control policies while handling uncertainty that fills the gap between high-level confidentiality requirements and low-level access control.

Keywords

Software Architecture, Access Control, Uncertainty, Confidentiality

1. Introduction

In today's world, a lot of data is measured, collected, and exchanged, e.g., in the context of social media, online shopping, smart home, or the Internet of Things (IoT). Here, confidentiality demands that "information is not made available or disclosed to unauthorized individuals, entities, or processes" [1]. As implied by "Privacy by Design", confidentiality requirements should be considered early in system design [2], e.g., by minimizing data collection and applying access control. Access control policies can be used to declare fine-grained rules on whether requests to data and resources should be accepted or rejected [3].

However, the information required to define and verify precise policies is limited at design time. We identify the following problems: First, policies are enforced using access control systems—a low-level security mechanism [3]—and are usually not defined in architectural abstraction (**P1**). Second, real-world confidentiality requirements are complex [3] and too abstract to make clear assumptions about confidentiality during design time (**P2**). This can result in over-estimations that restrict functionality by denying legitimate access [4]. Last, the high degree of uncertainty impedes early refinement and verification of policies (**P3**).

The impact of uncertainty on software architectures has already been discussed in related work [5, 6]. Uncertainty-aware access control has been proposed to incorporate trust in access decisions [7, 8]. Also, policy refinement has been discussed to close the gap between definition and verification [9, 10]. To the best of our knowledge, none of these refinement approaches do consider uncertainty and its impact on confidentiality. Additionally, architectural uncertainty is more often discussed in the context of performance, cost, or reliability analyses [11].

ECSA'21: 15th European Conference on Software Architecture, September 13–17, 2021, Växjö, Sweden
sebastian.hahner@kit.edu (S. Hahner)

Our vision to tackle these limitations is to consider uncertainty in definition, refinement, and verification of access control policies. We propose architecture-level modeling and refinement of policies and the adaptation of existing confidentiality analyses. Based on this vision and the problems (**P1** – **P3**) stated above, we define the following research questions:

RQ1 How to treat uncertainty on different abstraction levels and in varying context regarding its *impact* on confidentiality?

RQ2 How to *refine* high-level confidentiality requirements based on architectural modeling?

RQ3 How to *verify* refined policies against system architectures while considering uncertainty?

The benefits of this approach include a more precise impact assessment of uncertainty on access control policies. Based on the extended modeling, we aim for early feedback on confidentiality.

After summarizing related work in Section 2, we propose our approach in Section 3 and discuss the planned evaluation in Section 4. Section 5 concludes this paper.

2. Related Work

We group related work in three categories: Handling uncertainty in architectural modeling, uncertainty-aware access control, and policy refinement. We consider our approach to be in between these categories because they all lack either the architectural abstraction, the explicit treatment of uncertainty, or the refinement of confidentiality requirements.

Uncertainty-aware modeling approaches consider uncertainty as first-class entity in software architectures. Noppen et al. [5] discuss design decisions under imperfect information by explicitly modeling uncertain aspects of the architecture based on fuzzy techniques and design trees. Esfahani et al. [6] present *GuideArch*, an approach to explore the architectural solution space under uncertainty. This shall enable software architects to identify critical design decisions. Although these approaches consider uncertainty and fuzziness on architectural abstraction level, they do not consider confidentiality or other privacy-related quality properties.

In contrast, uncertainty-aware access control directly considers imperfect information in modeling access decisions. Bures et al. [7] propose situational patterns to cope with uncertainty in highly dynamic environments like Industry 4.0. Hengartner and Zhong [8] present an access control model for distributed systems that incorporates trust by explicitly specifying remaining uncertainty in access decisions. The common gap of uncertainty-aware approaches to model access control is the lack of refinement of high-level confidentiality requirements whose abstraction is also a source of uncertainty.

To guide software architects from high-level requirements to low-level policies, access control policy refinement techniques have been proposed. Su et al. [9] discuss the automated decomposition of policies based on the resource hierarchy in distributed applications. He and Antón [10] present and approach to define and refine access control policies by analyzing the specification of requirements and the system's database design. Unfortunately, these approaches do not explicitly consider any kind of uncertainty.

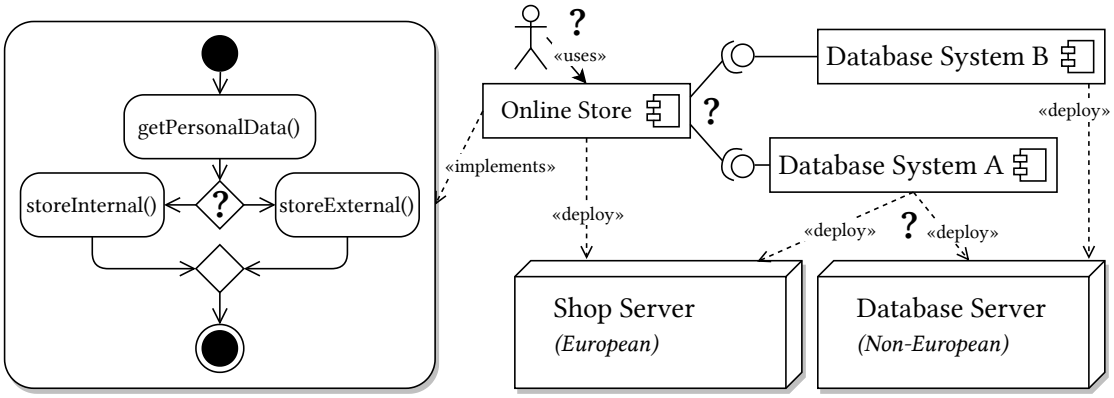

Figure 1: An example of a software architecture with different sources of uncertainty (denoted by "**?**")

3. Proposed Approach for Architectural Policy Analysis

Our approach is based on existing classifications of uncertainty that define the dimensions nature, level, and location [12]. The nature describes whether the uncertainty originates due to lack of information (i.e., *epistemic*) or inherent variability (i.e., *aleatory*). The level states how much is known about the uncertain influence. The location describes where the uncertainty occurs, e.g., in context, model structure, or input. Regarding software architectures, the sources system structure, system behavior and system environment have also been proposed [7].

Figure 1 shows different sources of uncertainty based on an exemplary software architecture that consists of multiple components, deployment locations as well as a modeled system behavior. Here, epistemic uncertainty occurs by the lack of information on architectural design decisions such as component or deployment choices. Additionally, the runtime behavior is a source of uncertainty, especially regarding imprecise access control policies in complex systems [3]. The environment and system input are sources of aleatory uncertainty, e.g., caused by imperfect sensor information or unexpected user behavior. Although this example is far from being comprehensive, it illustrates that uncertainty—even only regarding known unknowns—is wide-ranging but can be precisely described, e.g., by using and extending existing taxonomies.

Our approach includes architecture-level modeling, refinement, and verification of access control policies for business information systems. Based on design-time specification of uncertainty in architectural models, architects shall be able to estimate the impact of uncertainty on confidentiality, to iteratively refine high-level confidentiality requirements and verify uncertainty-afflicted access control policies [13]. This addresses the problems of abstraction (**P1**), over-estimation (**P2**) and uncertainty (**P3**). Our approach provides the following contributions:

C1 A *metamodel* for architecture-level access control policies under uncertainty.

C2 An uncertainty *impact analysis* of architectural design decisions on confidentiality.

C3 An uncertainty-aware, design-time access control policy *refinement* process.

C4 Policy *verification* based on adapting existing architecture-level confidentiality analyses.

To close the gap between high-level confidentiality requirements and low-level policy enforcement, we envision a continuous method that includes definition, refinement, and verification. We plan to expand the architecture description language Palladio [14] with means to express uncertainty in the architectural model and in policies (**C1**). This shall enable architects to describe confidentiality requirements in a structured way [10]. Based on characterizing different types of uncertainty and analyzing their propagation through the architecture, we aim to support architects with impact analysis capabilities to identify crucial design decisions for confidentiality early (**C2**). A refinement process resolves known uncertainty based on the modeled architecture and yields refined policies with higher precision and defined assumptions (**C3**). To ensure the validity of such refined polices (**C4**), they can be modeled, e.g., as constraints for existing architectural data flow analyses [15]. Note, that this process is iterative and thus can also be used to verify already refined policies against changes in requirements, system, or environment.

4. Planned Evaluation

We plan to evaluate our approach by using a *Goal-Question-Metric*-plan [16] with the goals:

G1 Evaluate the *expressiveness* of the architecture-level modeling of policies and uncertainty.

G2 Evaluate the *accuracy* of the uncertainty impact analysis for confidentiality.

G3 Evaluate the *correctness* of the uncertainty-aware access control policy refinement.

G4 Evaluate the *accuracy* of the verification of access control policies under uncertainty.

Regarding goal **G1**, we ask whether our metamodel can express different access control policy models and real-world software architectures under uncertainty. To evaluate goal **G2**, we plan to track the propagation and thus the impact of uncertainty in these architectures. Regarding goal **G3**, we evaluate the correctness by conducting a formal proof, e.g., by formalizing the system and its policies using a suitable formalism [15] and verifying the implication between coarse and refined policies. For goal **G4**, we evaluate the accuracy of the verification, e.g., by reusing already defined scenarios [17] and measuring precision and recall.

The biggest threat to validity of the planed evaluation is the existence of case studies which are important for external validity. We try to complement the case study-based evaluation whenever possible, e.g., by conducting a formal correctness proof in goal **G3**. Still, we rely on case studies with at least an architecture description and defined confidentiality requirements. A viable example is the German open-source contact tracing app *Corona-Warn-App* [18] that handles sensitive data and provides comprehensive public documentation. Another approach could be extracting failure causes from public data breaches and derive simplified scenarios.

5. Conclusion

We proposed our approach for uncertainty-aware policy refinement and verification. So far, we collected first results regarding architectural uncertainty and the refinement process. More comprehensive insights on the relation of uncertainty and confidentiality have yet to be gained.

Acknowledgments

This work was supported by funding of the Helmholtz Association (HGF) through the Competence Center for Applied Security Technology (KASTEL) (46.23).

References

[1] ISO, ISO/IEC 27000:2018(E) Information technology – Security techniques – Information security management systems – Overview and vocabulary, Standard, 2018.
[2] P. Schaar, Privacy by design, Identity in the Information Society 3 (2010) 267–274. Springer.
[3] P. Samarati, S. de Vimercati, Access Control: Policies, Models, and Mechanisms, in: Foundations of Security Analysis and Design, Springer, 2001, pp. 137–196.
[4] J. Juerjens, Principles for secure systems design, PhD Thesis, University of Oxford, 2002.
[5] Noppen, J., et al., Software development with imperfect information, Soft computing 12 (2008) 3–28. Springer.
[6] N. Esfahani, et al., GuideArch: Guiding the exploration of architectural solution space under uncertainty, in: ICSE, 2013, pp. 43–52.
[7] T. Bures, et al., Capturing Dynamicity and Uncertainty in Security and Trust via Situational Patterns, in: Leveraging Applications of Formal Methods, Verification and Validation: Engineering Principles, Springer, 2020, pp. 295–310.
[8] U. Hengartner, G. Zhong, Distributed, Uncertainty-Aware Access Control for Pervasive Computing, in: PerComW, 2007, pp. 241–246.
[9] Linying Su, et al., Automated decomposition of access control policies, in: POLICY, 2005, pp. 3–13.
[10] Q. He, A. I. Antón, Requirements-based Access Control Analysis and Policy Specification (ReCAPS), Information and Software Technology 51 (2009) 993–1009.
[11] D. Sobhy, et al., Evaluation of Software Architectures under Uncertainty: A Systematic Literature Review, ACM Transactions on Software Engineering and Methodology (2021).
[12] D. Perez-Palacin, R. Mirandola, Uncertainties in the modeling of self-adaptive systems, in: ICPE, 2014, pp. 3–14.
[13] S. Hahner, Dealing with Uncertainty in Architectural Confidentiality Analysis, in: Proceedings of the Software Engineering 2021 Satellite Events, 2021, pp. 1–6.
[14] R. H. Reussner, et al., Modeling and Simulating Software Architectures: The Palladio Approach, The MIT Press, 2016.
[15] S. Hahner, et al., Modeling Data Flow Constraints for Design-Time Confidentiality Analyses, in: ICSA-C, 2021, pp. 15–21.
[16] V. R. Basili, D. M. Weiss, A Methodology for Collecting Valid Software Engineering Data, IEEE Transactions on Software Engineering SE-10 (1984) 728–738.
[17] S. Seifermann, et al., Data-driven software architecture for analyzing confidentiality, in: ICSA, 2019, p. 1–10.
[18] Robert Koch Institute, et al., Corona-Warn-App Open-Source Project Website, 2020. URL: https://www.coronawarn.app/en/, accessed 7/29/2021.