Formal and Computer-Aided Design of Secure Distributed Applications

Matteo Maffei

Saarland University

Developing secure distributed applications is a challenging task. Each application provides different functionalities and comes with different, possibly sophisticated, security requirements. Open-endedness, lack of central authorities, and other specific features may add further complexity dimensions. The designer has to devise a cryptographic realization that is efficient, does not conflict with the intended functionality, and provably enforces the security requirements. The programmer has to provide an implementation, possibly in different platforms, that is not vulnerable to code-level attacks. The whole process requires coordination, creativity, and expertise in cryptography and formal verification.

Ideally, designers should be provided with high-level security APIs to state in a simple, yet precise, manner which functionality and security properties are to be realized, without necessarily having to think how this can be achieved. These high-level specifications should then be processed by a compiler, which is in charge of automatically synthesizing cryptographic protocols and executable code. Formal verification techniques and computational soundness results should finally ensure that the cryptographic protocols and the executable code output by the compiler fulfill the intended security properties.

In this talk, we will take the first steps in this direction by introducing a new framework for the specification and enforcement of security policies, focusing on privacy and authorization, two fundamental, and seemingly contradictory, properties. On the one hand, the access to sensitive resources should be granted only to authorized users; on the other hand, these users would like to share as little personal information as possible with third parties. These opposing goals make it hard to implement and enforce privacy-aware authorization policies. We will see how to specify these policies as logical rules, in which the traditional says modality from authorization logics is accompanied by existential quantification in order to express the secrecy of sensitive information. The implementation of these policies is obtained by a powerful combination of digital signatures and zero-knowledge proofs. This approach is flexible, is suitable for open-ended applications, and, depending on what information is kept secret, can be used to express and enforce secrecy as well as anonymity properties. This framework can be plugged smoothly into existing authorization languages in order to enrich their expressiveness and to offer support for privacy properties. In particular, we will see how to generate provably secure cryptographic protocols and executable code from high-level declarative specifications. We will finally discuss open challenges and directions of future research.