

Verification of security protocols: composition issues

Laboratory, institution and university LSV, ENS Cachan and LORIA, Nancy universités

The internship will be located at Nancy or at ENS Cachan depending on the choice of the candidate.

Team or project of the Lab Team SecSI at LSV and team Cassis at Loria

Name and email address of the advisor Véronique Cortier, cortier@loria.fr and Stéphanie Delaune, delaune@lsv.ens-cachan.fr

Indemnisation The internship is supported by the European grant **ProSecure** (ERC Starting Grant) and the ANR grant **VIP** (Programme JCJC).

Context. Security protocols are short distributed programs designed to achieve various security goals on data-processing networks, such as data privacy and data authenticity, even when communications between parties take place over channels controlled by an attacker. The increasing penetration of these protocols in many important applications (Internet communications, Credit Card payment, pay-per-view devices, e-voting, ...) makes designing and establishing the security of cryptographic protocols a very important research goal.

Formal methods have demonstrated their usefulness when designing and analyzing security protocols. They indeed provide rigorous frameworks and techniques that have allowed to discover new flaws. For example, a flaw was discovered on the Gmail setting [1] and corrections have been made since.

However, most of existing techniques are dedicated to the analysis of a single protocol, without taking into account other protocols which may be used at the same time. This is unrealistic for several reasons. Firstly, a number of protocols are verified under the assumption that agents share some pre-distributed keys (e.g. public keys or symmetric keys between agents and servers). But these keys might have been established by some other sub-protocols. There is no guarantee that a protocol remains secure if a specific key-exchange protocol is used to establish the keys, even if both protocols have been proven secure in isolation. Secondly, even apparently isolated protocols might interact in unexpected ways. For example, a user might choose the same password for two different network services, or a server might use the same key for different protocols. Even if the network services (or the different protocols) were proven secure in isolation, there is no security guarantee which carries over when they share keys or passwords.

Objectives. The goal of the internship is to develop modular reasoning about security such that we can infer security guarantees for the composition of protocols from the security guarantees of the individual protocols. Some composition results have already been obtained, e.g. [3, 2]. However, these results can only be applied when the protocols under study satisfy some syntactic conditions : a unique identifier has to be embedded in each ciphertext. While

the use of such tag is a good design principle, real-world security protocols do not use explicit tags. As a result, the existing composition results cannot be applied for analyzing real-world protocols in a modular and faithful way. The goal of this internship will be to establish a composition result which do not assume the use of explicit tags.

Expected skills. We are looking for candidates with good skills in Foundations of Computer Science (logic, automatic deduction, ...). Some knowledge in security is an asset but is not mandatory. The candidate will assimilate this knowledge during the internship.

Références

- [1] Alessandro Armando, Roberto Carbone, Luca Compagna, Jorge Cuéllar, and M. Llanos Tobarra. Formal analysis of saml 2.0 web browser single sign-on : breaking the saml-based single sign-on for google apps. In *Proceedings of the 6th ACM Workshop on Formal Methods in Security Engineering, FMSE 2008*, 2008.
- [2] Stefan Ciobâca and Véronique Cortier. Protocol composition for arbitrary primitives. In *Proceedings of the 23rd IEEE Computer Security Foundations Symposium, CSF 2010*, 2010.
- [3] Véronique Cortier and Stéphanie Delaune. Safely composing security protocols. *Formal Methods in System Design*, 34(1) :1–36, February 2009.