Analysing Cryptographically-Masked Information Flows Using Slicing

What is it all about?

Computer security is concerned with system architectures that can allow untrusted application programs to access sensitive information yet not allow such information to “leak” and become available to unauthorised entities. The latter can happen through covert channels that expose information which should be kept secret. Elimination of such channels requires analysis of the information flows within and between the components of the system architecture. One standard analysis approach is based on the notion of non-interference, which demands that public outputs are unchanged as secret inputs are varied. However, this requirement is violated in the presence of cryptographic operations. This raises the challenge of distinguishing between breaking non-interference because of legitimate use of (sufficiently strong) encryption and breaking non-interference due to an unintended leak.

This distinction can be achieved by relaxing the requirement of non-interference, e.g. by considering possibilistic non-interference instead [1]. Based on this notion, type systems have been developed that ensure proper information flow [2]. However, they suffer from weaknesses such as the insufficient consideration of control flow and the distribution of encryption keys.

The goal of this thesis is to follow an alternative approach that is based on slicing, a form of static analysis that can be used to determine (potential) dependencies between the inputs and outputs of system components [3]. While slicing in its original form is inherently control-flow sensitive, extensions are required to support the special handling of cryptographic operations. It is the aim of this thesis to develop such extensions. The work shall be carried out in the context of the D-MILS Project (http://www.d-mils.org/), in which compositional system construction and assurance methods are being investigated. It employs the MILS-AADL specification language [4], a variant of AADL (Architecture Analysis and Design Language), to describe both the hardware/software architecture of a system and its operational behaviour. An initial version of such a slicing algorithm has been developed in [5] for a fragment of AADL.

What has to be done?

The goal is to develop and implement a static analysis based on slicing for determining, for a given MILS-AADL component $C$ and one of its outgoing event and/or data ports $p$, the set of incoming ports of $C$ on which $p$ (potentially) depends, taking into account the encryption keys that are accessible to $C$. More concretely, this will involve the following steps:

2. Implementation of the algorithm in python, using the MILS-AADL frontend that is being developed within the D-MILS Project.

Requirements

Basic knowledge in algorithms and data structures

Contact

Thomas Noll, noll@cs.rwth-aachen.de, Tel. +49-241-80-21213.

References