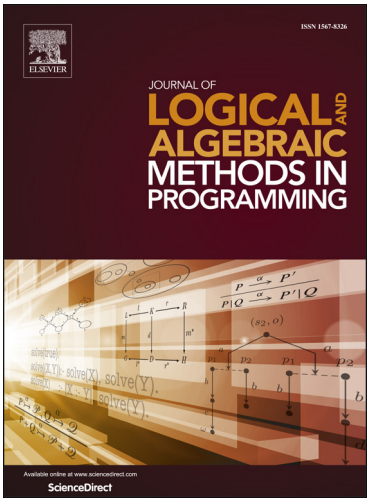
Accepted Manuscript



A secrecy-preserving language for distributed and object-oriented systems

Toktam Ramezanifarkhani, Olaf Owe, Shukun Tokas

PII: S2352-2208(17)30060-3
DOI: https://doi.org/10.1016/j.jlamp.2018.04.001
Reference: JLAMP 222

To appear in: *Journal of Logical and Algebraic Methods in Programming*

Received date: 15 March 2017
Revised date: 6 March 2018
Accepted date: 20 April 2018

Please cite this article in press as: T. Ramezanifarkhani et al., A secrecy-preserving language for distributed and object-oriented systems, *J. Log. Algebraic Methods Program.* (2018), https://doi.org/10.1016/j.jlamp.2018.04.001

A Secrecy-Preserving Language for Distributed and Object-Oriented Systems☆
May 3, 2018

Toktam Ramezanifarkhani, Olaf Owe, and Shukun Tokas[1]

Department of Informatics, University of Oslo, Oslo, Norway

Abstract

In modern systems it is often necessary to distinguish between confidential (low-level) and non-confidential (high-level) information. Confidential information should be protected and not communicated or shared with low-level users. The *non-interference* policy is an information flow policy stipulating that low-level viewers should not be able to observe a difference between any two executions with the same low-level inputs. Only high-level viewers may observe confidential output. This is a non-trivial challenge when considering modern distributed systems involving concurrency and communication.

The present paper addresses this challenge, by choosing language mechanisms that are both useful for programming of distributed systems and allow modular system analysis. We consider a general concurrency model for distributed systems, based on concurrent objects communicating by asynchronous methods. This model is suitable for modeling of modern service-oriented systems, and gives rise to efficient interaction avoiding active waiting and low-level synchronization primitives such as explicit signaling and lock operations. This concurrency model has a simple semantics and allows us to focus on information flow at a high level of abstraction, and allows realistic analysis by avoiding unnecessary restrictions on information flow between confidential and non-confidential data.

Due to the non-deterministic nature of concurrent and distributed systems, we define a notion of *interaction non-interference* policy tailored to this setting. We provide two kinds of static analysis: a secrecy-type system and a trace analysis system, to capture inter-object and network level communication, respectively. We prove that interaction non-interference is satisfied by the combination of these analysis techniques. Thus any deviation from the policy caused by implicit information leakage visible through observation of network communication patterns, can be detected. The contribution of the paper lies in the definition of the notion of *interaction non-interference,* and in the formalization of a secrecy type system and a static trace analysis that together ensure interaction non-interference. We also provide several versions of a main example (a news subscription service) to demonstrate network leakage.

Keywords: Concurrent Objects; Active Objects; Asynchronous Methods; Communication Patterns; Non-Interference; Interaction Non-Interference; Information Flow; Secrecy; Confidentiality; Distributed Systems; Network Leakage; Inter-Object Leakage.

1. Introduction

Programming languages can provide fine-grained control for security issues because they allow accurate and flexible security information analysis of program components [1]. In particular, to specify and enforce information-flow policies, the effectiveness of language-based techniques has been established. Information-flow policies are essentially specified based on a mapping from the set of logical information holders to

[1]email: {toktamr,olaf,shukunt}@ifi.uio.no