Theoretical Computer Science 597 (2015) 62–87

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Quantifying information leakage of randomized protocols ☆,☆☆

Fabrizio Biondi [a,*], Axel Legay [a], Pasquale Malacaria [b], Andrzej Wąsowski [c]

[a] IRISA/INRIA Rennes, 263 Avenue du Général Leclerc, 35042 Rennes, France
[b] Queen Mary University of London, Mile End Road, E1 4NS, London, United Kingdom
[c] IT University of Copenhagen, 7 Rued Langgaards Vej, 2300 Copenhagen-S, Denmark

ARTICLE INFO

Article history:
Received 25 August 2014
Received in revised form 9 April 2015
Accepted 8 July 2015
Available online 21 July 2015
Communicated by B.P.F. Jacobs

Keywords:
Model checking
Quantitative information flow
Information leakage
Markov chain
Markov decision process
Channel capacity
Probabilistic system

ABSTRACT

The quantification of information leakage provides a quantitative evaluation of the security of a system. We propose the usage of Markovian processes to model deterministic and probabilistic systems. By using a methodology generalizing the lattice of information approach we model refined attackers capable to observe the internal behavior of the system, and quantify the information leakage of such systems. We also use our method to obtain an algorithm for the computation of channel capacity from our Markovian models. Finally, we show how to use the method to analyze timed and non-timed attacks on the Onion Routing protocol.

1. Introduction

Quantification of information leakage is a recent set of techniques in security analysis that evaluates the amount of information about a secret (for instance about a password) that can be inferred by observing a system. It has sound theoretical foundations in Information Theory [1,2]. It has also been successfully applied to practical problems like proving that patches to the Linux kernel effectively correct the security errors they address [3]. It has been used for analysis of anonymity protocols [4,5] and analysis of timing channels [6,7]. Intuitively, *leakage of confidential information of a program is defined as the difference between the attacker's uncertainty about the secret before and after available observations about the program* [1].

The underlying algebraic structure used in leakage quantification for *deterministic* programs is the *lattice of information* (LoI) [1]. In the LoI approach an attacker is modeled in terms of possible observations of the system he can make. LoI uses an equivalence relation to model how precisely the attacker can distinguish secrets based on the observations of the system. An execution of a program is modeled as a partial function from inputs to observables. In this paper we follow the LoI approach but take a process view of the system. A process view of the system is a more concise representation of behavior than an observation relation. Moreover a process view does not require that the system be deterministic, which allows us to handle randomized protocols—for the first time using a generic, systematic and implementable LoI-based methodology.

☆ An earlier version of this paper appeared in the 14th International Conference on Verification, Model Checking, and Abstract Interpretation (2013).

☆☆ The research presented in this paper has been partially supported by MT-LAB, a VKR Centre of Excellence for the Modelling of Information Technology.

* Corresponding author.
E-mail addresses: fabrizio.biondi@inria.fr (F. Biondi), axel.legay@inria.fr (A. Legay), p.malacaria@qmul.ac.uk (P. Malacaria), wasowski@itu.dk (A. Wąsowski).

http://dx.doi.org/10.1016/j.tcs.2015.07.034
0304-3975/

We use Markov Decision Processes to represent the probabilistic partial-information semantics of programs, using the nondeterminism of the model for the choices that depend on the unknown secret. We define the leakage directly on such model. With our method we can distinguish the inherent randomness of a randomized algorithm from the unpredictability due to the lack of knowledge about the secret. We exploit this distinction to quantify leakage only for the secret, as the information leakage about the random numbers generated is considered uninteresting (even though it is information in information theoretical sense). We thus work with both deterministic and randomized programs, unlike the previous LoI approach.

We give a precise encoding of an *attacker* by specifying her prior knowledge and observational capabilities. We need to specify which of the logical states of the system can be observed by the attacker and which ones he is able to distinguish from each other. Given a program and an attacker we can calculate the leakage of the program to the attacker.

We also show how to turn the leakage computation into maximization of leakage: we compute the maximum leakage over all possible prior information of attackers *ceteris paribus*, or in other words, the leakage for the worst possible attacker without specifying the attacker explicitly. This maximum leakage is known as the *channel capacity* of the system [8]. Since we are able to model a very large class of attackers the obtained channel capacity is robust. Computing channel capacity using this method requires solving difficult optimization problems (as the objective is nonlinear), but we show how the problem can be reduced to standard reward optimization techniques for Markovian models for a class of interesting examples.

Our method can be applied to finite-state systems specified using a simple imperative language with a randomization construct. It can also be used for systems modeled directly as Markov Decision Processes. We demonstrate the technique using an MDP model of the known Onion Routing protocol [9], showing that we can obtain the channel capacity for a given topology from an appropriate Markov Decision Process describing the probabilistic partial information behavior of the system. Also, our behavioral view of the system allows us to encode an attacker with time-tracking capabilities and prove that such an attacker can leak more information than the canonical attacker that only observes the traffic on the compromised nodes. Timing-based attacks to the Onion Routing protocol have been implemented before [10,11], but to our best knowledge the leakage of timing-attacks has not been quantified before.

Our contributions include:

- A simple partial-observability semantics for imperative programs allowing to model attack scenarios directly derived from code.
- A definition of leakage that generalizes the LoI approach to programs with randomized choices (strictly including the class of deterministic programs), and dually the first application of the LoI approach to system specifications.
- A method for computing leakage for scenarios modeled as described above. The method is fully implementable.
- A method to parameterize the leakage analysis on the attacker's prior information about the secret, to allow the computation of channel capacity by maximizing an objective characterizing leakage as a function of prior information.
- The worst-case analysis of the Onion Routing protocol when observed by non-time-aware and time-aware attackers able to observe the traffic passing through some compromised nodes.

The paper proceeds as follows. Section 2 provides the core background on probabilistic systems and the LoI approach. Section 3 gives an overview of our new leakage quantification method. The non-obvious steps are further detailed in Sections 4–6. In Section 8 we explain how to use the method for computing channel capacity, and we use this technique to analyze leakage in the Onion Routing protocol against untimed and timing attacks (Section 9). We discuss the related work (Section 11) and conclude (Section 12).

2. Background

We often refer to deterministic, stochastic and nondeterministic behavior. We use the adjective *deterministic* for a completely predictable behavior, *stochastic* for a behavior that follows a probability distribution over some possible choices, and *nondeterministic* for a choice where no probability distribution is given.

We define common concepts in probability theory and information theory that are used throughout the paper. We refer to basic books on the subject [12,13] for the definitions of sample space S, probability of event $P(E)$ and so on. We use $\mathcal{X}$ to denote a *discrete stochastic process*, i.e. an indexed infinite sequence of discrete random variables $(X_0.X_1, X_2, \ldots)$ ranging over the same sample space S. The index of the random variables in a stochastic process can be understood as modeling a concept of discrete time, so X_k is the random variable representing the system at time unit k.

2.1. Markovian models

A discrete stochastic process is a time-invariant *Markov chain* $\mathcal{C} = (C_0, C_1, C_2, \ldots)$ iff $\forall k \in \mathbb{N}.\ P(C_k|C_{k-1}, \ldots, C_0) = P(C_k|C_{k-1})$. A Markov chain on a sample space S can also be defined as follows:

Definition 1. A tuple $\mathcal{C} = (S, s_0, P)$ is a (time-invariant) Markov Chain (MC), if S is a finite set of states, $s_0 \in S$ is the initial state and P is an $|S| \times |S|$ probability transition matrix, so $\forall s, t \in S.\ P_{s,t} \geq 0$ and $\forall s \in S.\ \sum_{t \in S} P_{s,t} = 1$.