Dolev-Yao Unsecrecy Checking in Python

Seth Ahrenbach

1 Introduction

This report describes the work conducted to implement a language and algorithm for symbolically checking whether a protocol leaks a secret to a Dolev-Yao attacker. The language is based on a Backus-Naur Form definition of cryptographic primitives including both symmetric and asymmetric key encryption, including signature and verification. The algorithm I implement checks whether a set of terms in the language allows for the deduction of a given term, representing a secret. It does so by representing a set of terms as a minimal *Directed Acyclic Graph* (DAG), with each node representing a subterm from the set of terms. It then recursively marks nodes, *i.e.* subterms, that the attacker can deduce, based on a set of deduction rules. The algorithm runs in time polynomial in the size of the set of terms. Solving this deduction problem represents the core computational work required for symbolic protocol checking.

2 Background

3 Algorithm

4 Implementation

5 Conclusion