

Preventing timing attacks through the type

Erik and Mads

Timing attacks is something programmers traditionally give very little attention. It is a type of attack that initially seems insievable. Even after one has convinced herself that timing attacks are real threats, the prevention mechanisms are hard. We propose a technique to prevent timing attacks that takes advantage of the type system. The approach forces the programmer to prove the absence of timing attacks.

Introduction

The Language

Core Types

As the most important core types we have

- **Unit:**
- **Sum:**
- **Product:**
- **List:**

Perifical Types

Many target architectures utilize special operations to perform fast computations for specific types. It is therefore very common to have special types built into a source language, to allow the programmer to utilize these coconstructs.

To make it easier for us to program applications we have built-in support for following types:

- **Int:** These fixed-width integers. We built in literal constructor and the usual operations. We assume all operations take constant equal time.
- **Booleans:** Booleans along with typical operations. The assumption is, again, that the operations take constant, equal time.

Applications

It is hard to directly write in expressed language. Even though we rely on much inference, the type errors and other things are.

User Authentication

Conclusion