Verified Time Balancing of Securit Protocols

Donovan Crichton

January 2019

The Problem

- Can we formally prove that an implementation of a network security protocol is immune to timing side-channel attacks?
- In other words, can we show that a model of the protocol, along with its implementation is unable to leak any timing information to an observer.

Rationale

- Some departments in the ASD spent a large amount of time manually verifying cryptographic processes in vendor code.
- Requires large amounts of time and expertise, resulting in a slow verfication process
- Formal methods may allow the automation of these verification processes.

Formal Methods

- A mathematical approach to the development, specification and verification of software.
- Uses "theorem proving assistants", software that formally verifies against a specification.
- We can implement a network security protocol inside a theorem prover (namely Idris).

How do proof assistants work?

- Rely on on a relationship between proofs in mathematics and computer programs known as the Curry-Howard correspondance.
- The proof system of intuitionistic natural deduction can be directly interpreted as a model of computation known as lambda calculus.

A quick Idris example.

A simple function in Idris

```
f : Nat -> Nat
f x = x + 2
```

The same function in familiar notation.

$f : \mathbb{N} \mapsto \mathbb{N}$

$f(x) = x + 2$

Logic in the Idris language.

Logic Term	**Logic Symbol**	**Idris Symbol**	**Idris Term**
Implication	$p \Rightarrow q$	`p -> q`	Arrow
Conjunction	$p \wedge q$	`(p, q)`	Pair (Product)
Disjunction	$p \vee q$	`Either p q`	Enum (Sum)
Negation	$\neg p$	`p -> Void`	Void Type
IFF/Eq	$p \equiv q$, $p \iff q$	`(p -> q, q -> p)`	Pair Arrows
Universal	$\forall x.\ P\ x$	`p -> Type`	Π Type
Existential	$\exists x.\ P\ x$	`(x ** P x)`	Σ Type
???	???	`p = q`	Type Equality