# Three projects around formal proof and blockchains

Thomas Sibut-Pinote

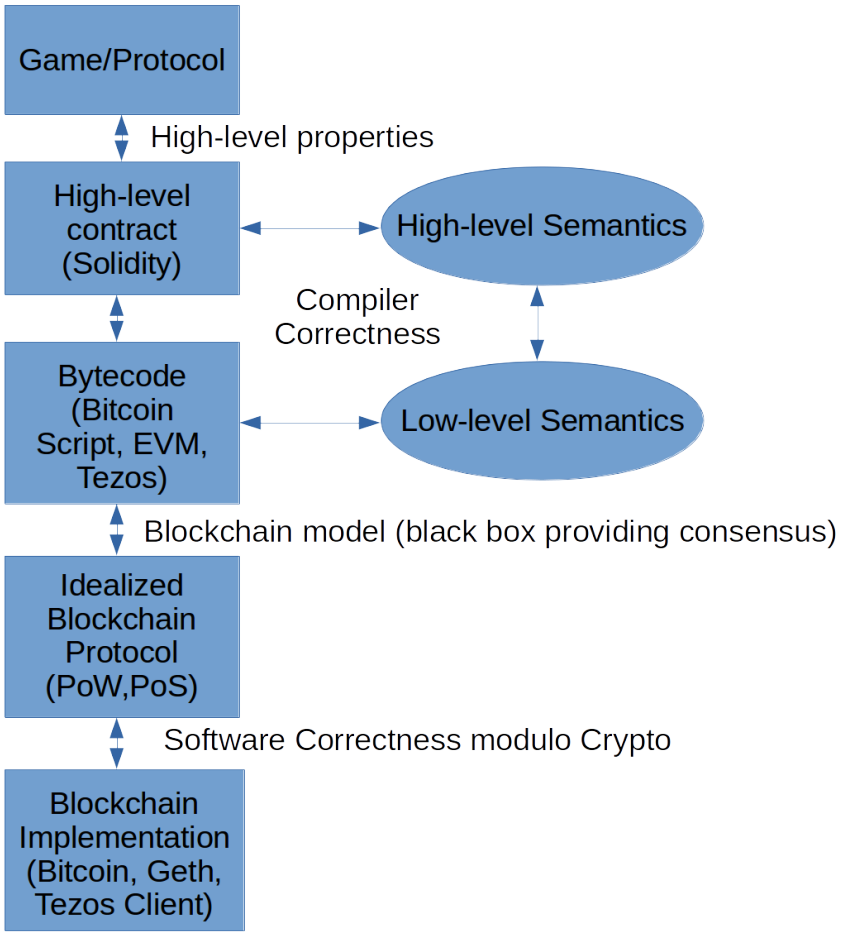
December 12, 2016

# Current Situation

PhD student with Assia Mahboubi, "Numeric Computations And Mathematical Proofs: From Rigorous To Formal Proofs": estimating integrals in Coq

Defense around Fall 2017

# The Cryptocurrency Stack Seen From Formal Proof



# Natural Questions

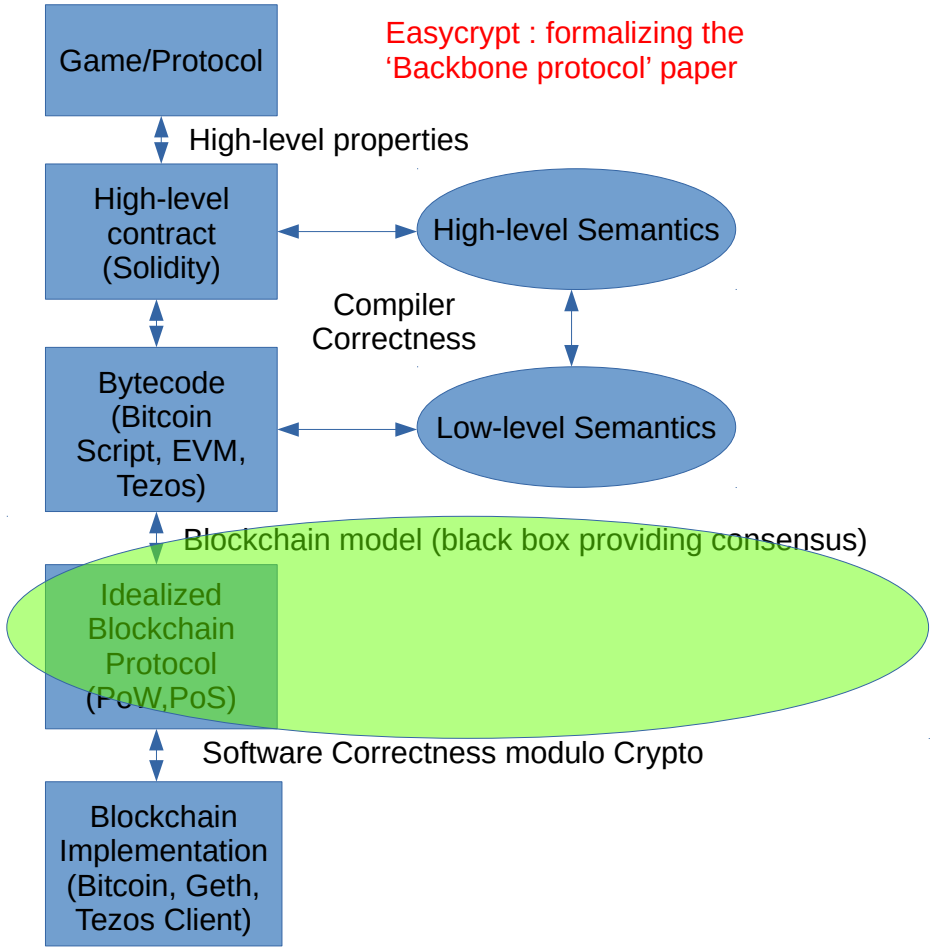
- Is this contract fair to its users (e.g. Ponzi Scheme)?
- Does this contract method do what it pretends to do? (e.g. DAO: no!)
- Is this EVM bytecode equivalent to the high-level code I'm being shown?
- Does this blockchain protocol reach consensus? Does it have safety? Does it have liveness?
- Is this Bitcoin client faithful to the Bitcoin protocol? (bonus question: what is the Bitcoin protocol?)

# Natural Questions

- Is this contract fair to its users (e.g. Ponzi Scheme)?
- Does this contract method do what it pretends to do? (e.g. DAO: no!)
- Is this EVM bytecode equivalent to the high-level code I'm being shown?
- Does this blockchain protocol reach consensus? Does it have safety? Does it have liveness?
- Is this Bitcoin client faithful to the Bitcoin protocol? (bonus question: what is the Bitcoin protocol?)

**Many levels of properties!**

# Work with Easycrypt



# Quick description

## Tool: Easycrypt

A toolset for reasoning about relational properties of probabilistic computations, extended to reason about the security of cryptographic systems.

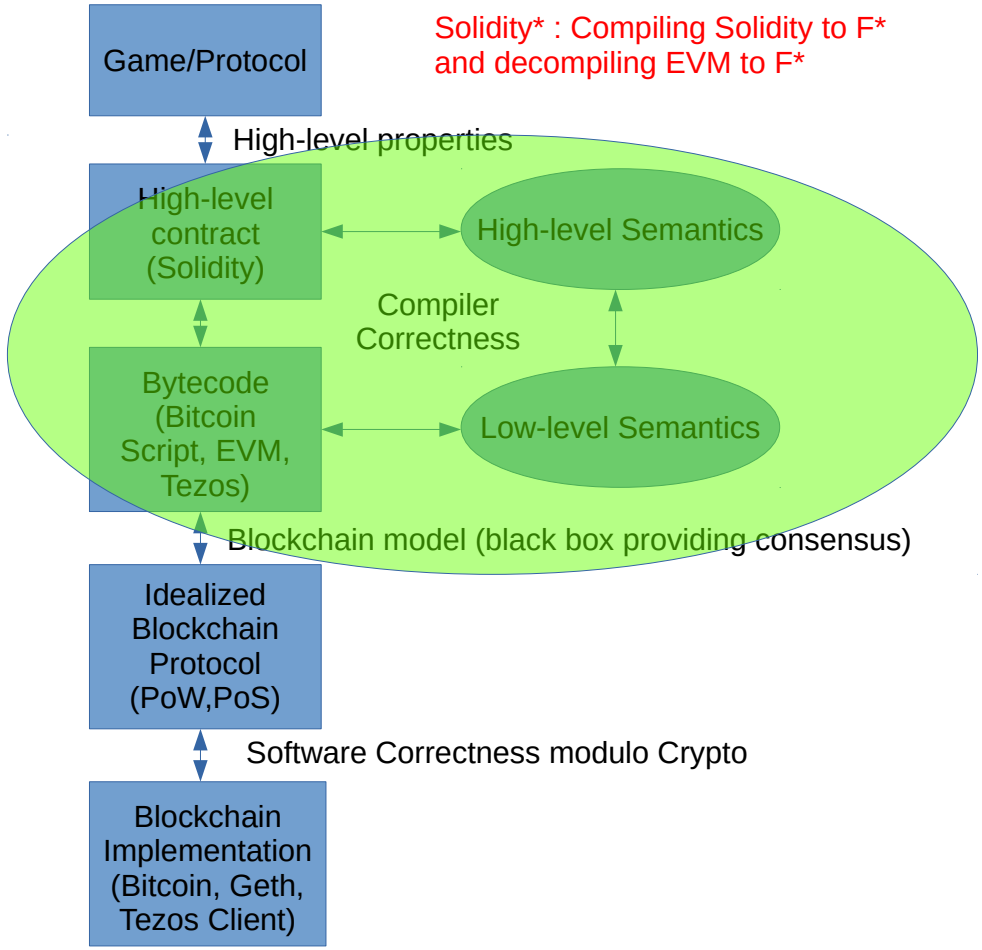
## Nature of work

- Improving tactics for proofs of programs using Hoare logic;
- Formalizing [1].

## Joint Work

- Ongoing joint work with Pierre-Yves Strub
- Based on 'The Bitcoin Backbone Protocol: Analysis and Applications' [1] by J.Garay, A. Kiayias and N. Leonardos

# Solidity*


Game/Protocol
Solidity* : Compiling Solidity to F*
and decompiling EVM to F*
High-level properties
High-level contract (Solidity)
High-level Semantics
Compiler Correctness
Bytecode (Bitcoin Script, EVM, Tezos)
Low-level Semantics
Blockchain model (black box providing consensus)
Idealized Blockchain Protocol (PoW,PoS)
Software Correctness modulo Crypto
Blockchain Implementation (Bitcoin, Geth, Tezos Client)

# Quick Description

## F*

ML-like functional programming language for program verification

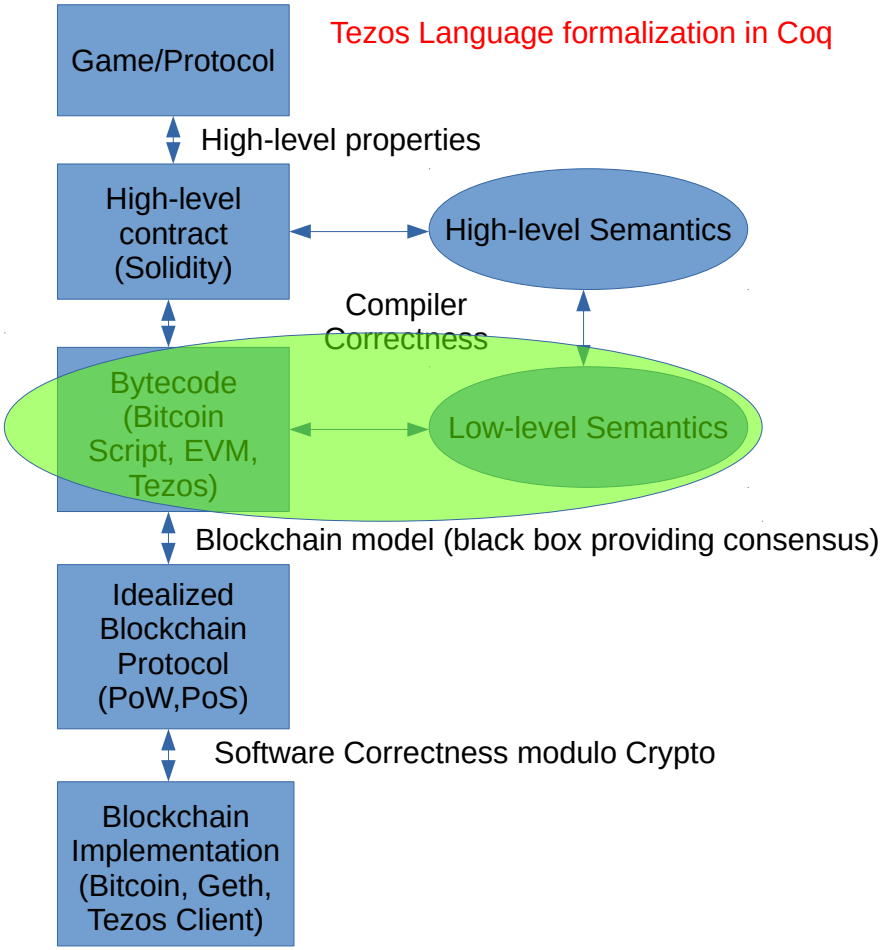
## Work

- Decompiler EVM $\rightarrow$ F* (e.g. for gas);
- Compiler Solidity $\rightarrow$ F* (e.g. for reentrancy);
- Possible goal: DSL for contracts inside F*, compiled to EVM.

## Context

- Started as a hackathon project at MSRC, mainly with (a large team of) MSR and Inria members
- Our PLAS paper: 'Formal Verification of Smart Contracts' [2]

# The Tezos contract language in Coq



# Quick Description

## Tool

The Coq theorem prover.

## Nature of work

- Defining the formal semantics of the Tezos smart contract language;
- Proving elementary properties about it (like type preservation);
- Toying with program proofs (but hard);
- For now, on my free time.

## With Whom?

Joint work with Anton Trunov and in regular contact with the Tezos team.

# Future

- Finishing PhD in $\sim$ 1 year
- Looking for a PostDoc in a related area :-)

# Thanks

Thanks for listening! Any questions?


Juan Garay, Aggelos Kiayias, and Nikos Leonardos.
The bitcoin backbone protocol: Analysis and applications.
Cryptology ePrint Archive, Report 2014/765, 2014.
http://eprint.iacr.org/2014/765.


Karthikeyan Bhargavan, Antoine Delignat-Lavaud, Cédric Fournet, Anitha Gollamudi, Georges Gonthier, Nadim Kobeissi, Natalia Kulatova, Aseem Rastogi, Thomas Sibut-Pinote, Nikhil Swamy, and Santiago Zanella-Béguelin.
Formal Verification of Smart Contracts: Short Paper.
In *ACM Workshop on Programming Languages and Analysis for Security*, Vienna, Austria, October 2016.