mit
media
lab

How Formal Analysis and Verification Add Security to Blockchain-based Systems

January 26, 2017
Shin’ichiro Matsuo (MIT Media Lab)
Pindar Wong (VeriFi Ltd.)

Outline of this talk

Security Definition of Blockchain-based system

Technology and Security Layer

Applicability of Formal Analysis and Verification

Four layers are suitable: Implementation, Backbone Protocols, Application Protocols and Application Logic

Idea toward Domain specific language

Background: The case of “the DAO”

Had chance to lose 50M Dollars by this attack.

Caused by vulnerability of the code

The way of workaround is still not decided.

Problems

Vulnerability handling

Procedure for work around

Over-investment to uncertified technology and codes

Intersection of technology and financial incentive

Security definitions of blockchain

Several proposals on back-bone protocol

Few consideration for security of entire system

Security Definitions for backbone-protocol [1]

Two definitions

Common Prefix Property

If two players prune a sufficient number of blocks from their chains, they will obtain the same prefix.

Chain Quality

Any large enough chunk of an honest player's chain will contain some block from honest players.

There are results on provable secure protocol but needs assumptions [KKRDO16]

Provable Secure Blockchain with Proof of Stake [KKRDO16]

Prove Two Requirements of Blockchain

Persistence and Liveliness [1]: Robustness of the Blockchain

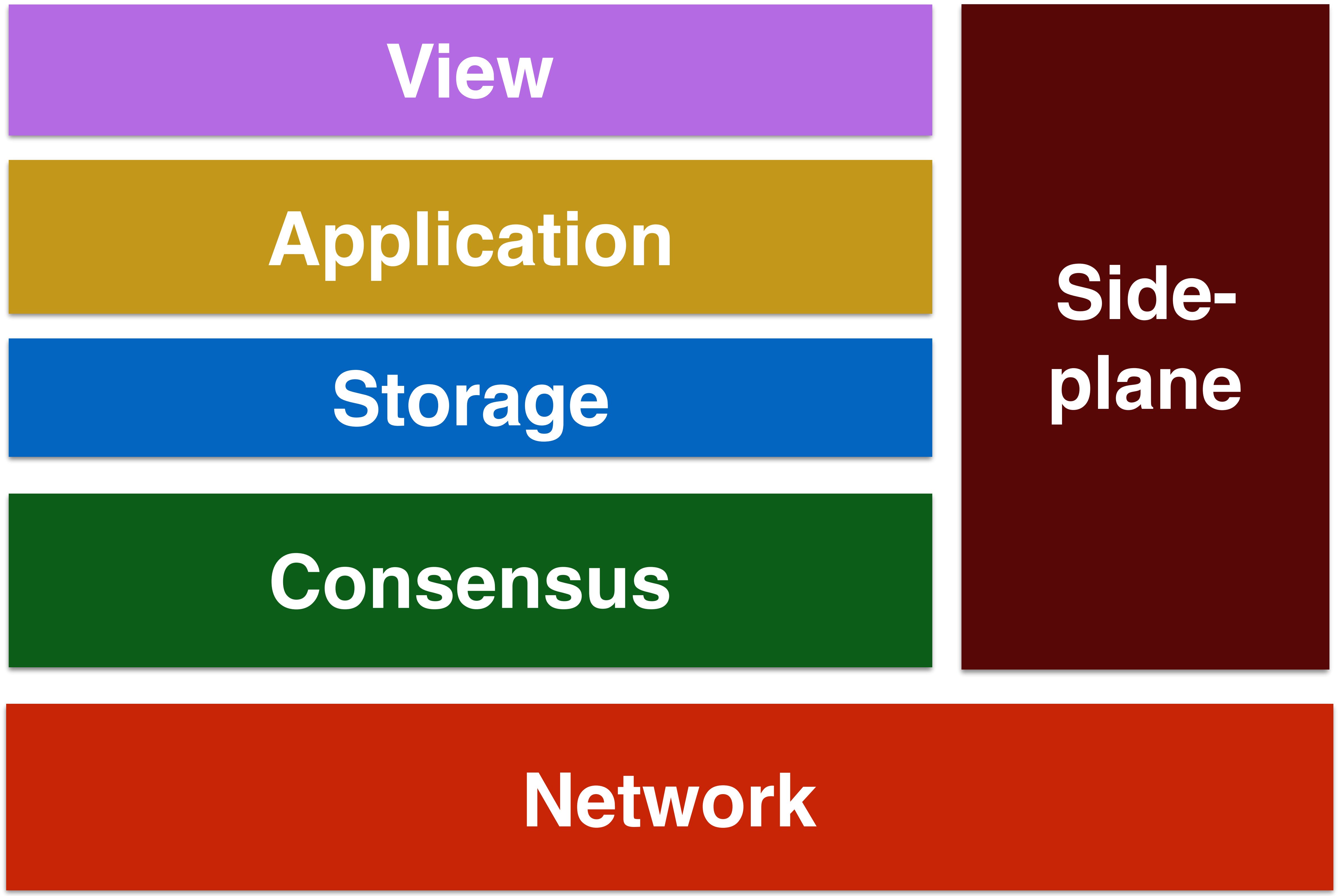
Propose Provable Secure Protocol

Use Multi-Party Coin Flipping for leader election to produce randomness

Many Assumptions

Highly Synchronous
Majority of Selected Stakeholder is available
The Stakeholders do not remain offline for a long time

Example of Blockchain Technology layer [3]



Network: broadcasting transactions and blocks

Consensus: the agreement-reaching engine

Storage: bootstrapping new nodes, storing archival data

Application: transaction graph, scripting language semantics

View: cached summary of the transaction log

Side-plane: off-chain contracts

Layers for security consideration

Operation	Key Management, Audit, Backup	ISO/IEC 27000
Implementation	Program Code, Secure Hardware	ISO/IEC 15408
Application Logic	Scripting Language for Financial Transaction, Contract	Secure coding guides
Application Protocol	Privacy protection, Secure transaction	ISO/IEC 29128
Backbone Protocol	P2P, Consensus, Merkle Tree	ISO/IEC 29128
Cryptography	ECDSA, SHA-2, RIPEMD160	NIST,ISO

Cryptography Layer

Security goals in Blockchain
Realizing authenticity and integrity

Digital Signature: ECDSA
Hash Function: SHA-2, RIPEMD-160
Underlying Mathematics: Secure parameter of elliptic curve

Firm analysis model

Provable Security
Estimation of security margin

Many theoretic results and evaluations

Academic proof, Standardization by NIST, ISO/IEC, IETF(IRTF), IEEE

Backbone Protocol Layer

Security goals in Blockchain

Realizing de-centralization and robustness by P2P network
Realizing consistency of transaction by consensus algorithm
Ensuring order of transaction by Merkle hash tree and chaining

Security definition, requirements and evaluation

No fixed security definition (researches are ongoing)
Evaluation by mathematical proof or formal analysis

Standard for evaluation

ISO/IEC 29128 for cryptographic protocols

Application Protocol Layer

Security goals in Blockchain

Privacy Protection
Secure data transmission
Secure transaction

Security definition, requirements and evaluation

Need application specific security definition
Evaluation by mathematical proof or formal analysis

Standard for evaluation

ISO/IEC 29128 for cryptographic protocols

Application Logic Layer

Security goals in Blockchain

Soundness and completeness in application logic

Security definition, requirements and evaluation

Checking the existence of bug

Implementation Layer

Security goals in Blockchain

Protection of signing key and prevent forgery of digital signature

Against black box attacker (main channel), gray box attacker (Side channel) and white box attacker (rooted device)

Security definition, requirements and evaluation

Capability of the adversary

Standard for evaluation

ISO/IEC 15408

Operation Layer

Security goals in Blockchain

Key management
Audit of operation

Security definition, requirements and evaluation

Depends on security policy of each system

Standard for evaluation

ISO/IEC 27000 Series (Information Security Management System)

Formal Analysis and Formal Verification

Formal Analysis

Evaluating the possibility of attack on the specification of the protocol, products or system by conducting some mathematical formalization of the security requirements, specifications and operational environment (an adversarial model).

Formal Verification

To verify the correctness of the specification of the protocol, products or system formal methods such as automated axiomatic theorem proving or model checking.

Applicability of formal verification

Operation	Key Management, Audit, Backup	ISO/IEC 27000
Implementation	Program Code, Secure Hardware	ISO/IEC 15408
Application Logic	Scripting Language for Financial Transaction, Contract	Secure coding guides
Application Protocol	Privacy protection, Secure transaction	ISO/IEC 29128
Backbone Protocol	P2P, Consensus, Merkle Tree	ISO/IEC 29128
Cryptography	ECDSA, SHA-2, RIPEMD160	NIST,ISO

Formal analysis methods and tools for cryptographic protocol

	Model checking		Theorem proving
Symbolic	NRL FDR AVISPA	SCYTHER ProVerif AVISPA (TA4SP)	Isabelle/HOL
Cryptographic		CryptoVerif	BPW(in Isabelle/HOL) Game-based Security Proof（in Coq)

Unbounded

Formal analysis of Implementation

Both software/ hardware implementation

Security mechanisms which use cryptographic algorithms, protocols, random number generator and key management mechanisms

Target of Evaluation

- Crypto-token wallet (Hardware/Software)
- HSM (Hardware Security Module)

Standards and Examples for Implementation Layer

<u>Industrial Standard</u>
Common Criteria (ISO 15408)
Define seven EALs (Evaluation Assurance Levels)

EAL6 requires semi formal analysis on the design and implementation
EAL7 requires fully formal analysis on design and implementation

Examples of formal analysis for implementation
EAL6

FeliCa IC chip RC-SA00
Crypto Library V1.0 on P60x080/052/040yVC(Y/Z/A)/yVG
Microcontrôleurs sécurisés SA23YR48/80B et SB23YR48/80B

Analysis of Cryptographic Protocols: Formal Verification vs UC Framework

Formal Verification

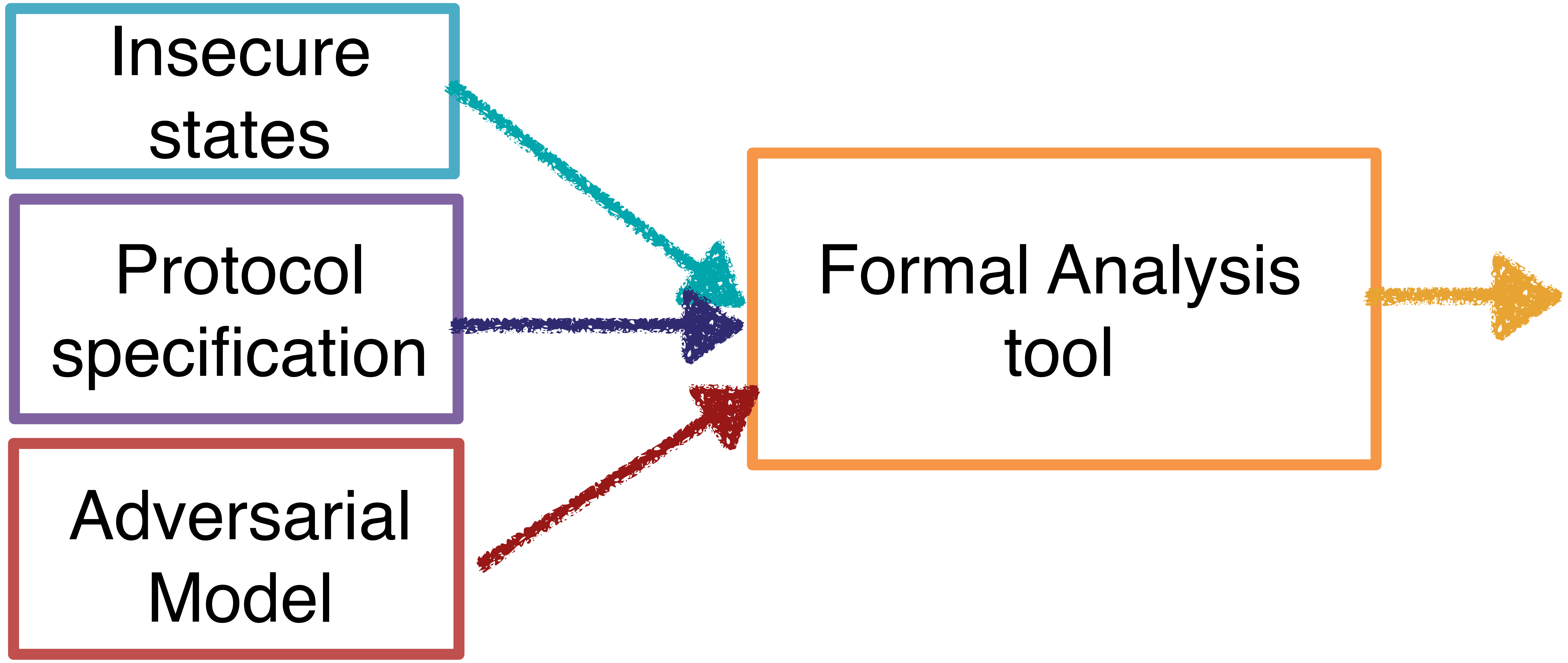
- Formal method
- Find the existence of insecure state
- Automated verification
- Tool-aided

Mathematical Proof

- Rigorous proof
- Estimate probability of attack
- Same as cryptographic Primitive

Formal Analysis of Cryptographic Protocols

- Check if the insecure state may happen in execution
  - Protocol specification
  - Adversarial model
  - Insecure states to be avoided



- Output if the insecure states may happen.
- If yes, output trace by which the insecure state is happen.

Formal Analysis of Backbone protocols and application protocols

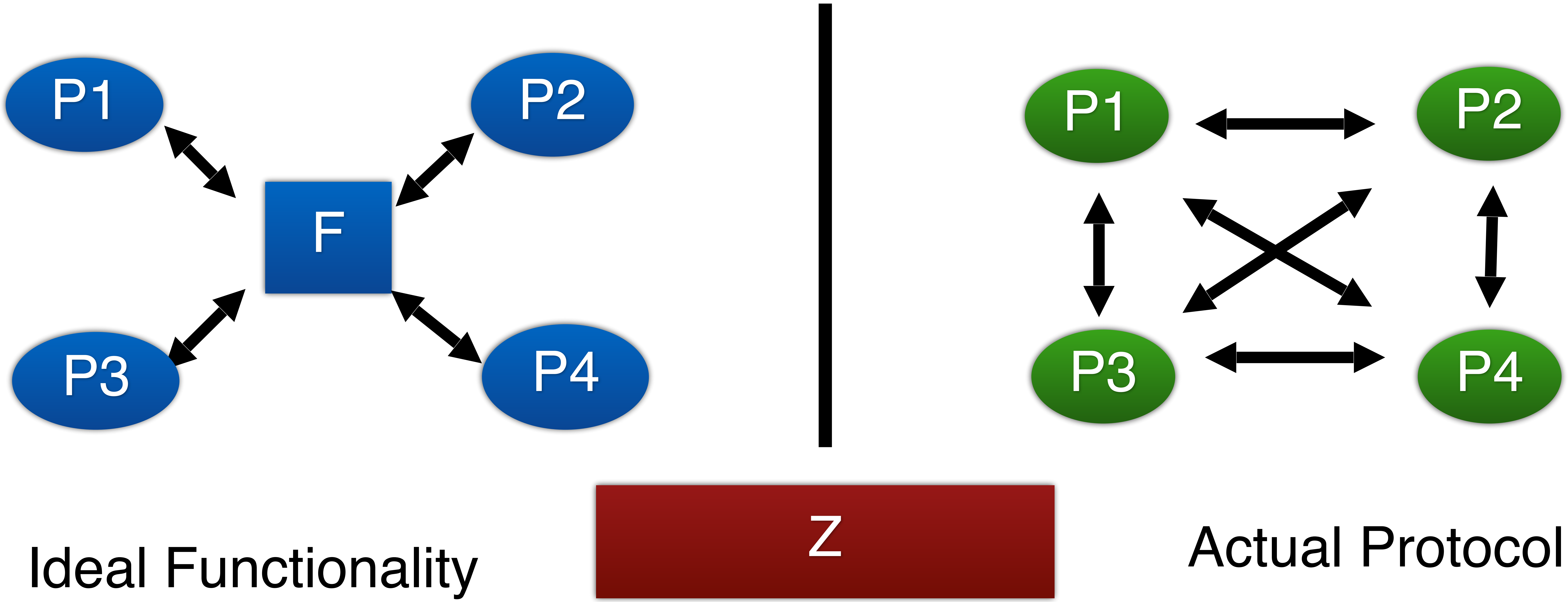
Explore the existence of state against security goals (Security Properties)

Dolev-Yao Model

- Basically Cryptographic algorithm is idealized
- Only a party who has a decryption key obtains plaintext.
- The other party obtains nothing.
- Same treatment for digital signature and others
- An adversary can control communication channel.

UC Framework

- Define the ideal functionality, then prove that the actual protocol is indistinguishable against the ideal functionality.



Combination of Formal Analysis and Mathematical proof

- **Combine the merit of formal analysis and mathematical rigorous proof.**
- **Many researches from 2002**
  - **Game-based evaluation**
  - **Cryptoverif**

The case of SSL/TLS

Many attacks/vulnerabilities are found during this 5 years.

Heartbleed, Poodle, FREAK, DROWN, CCS Injection

Problems

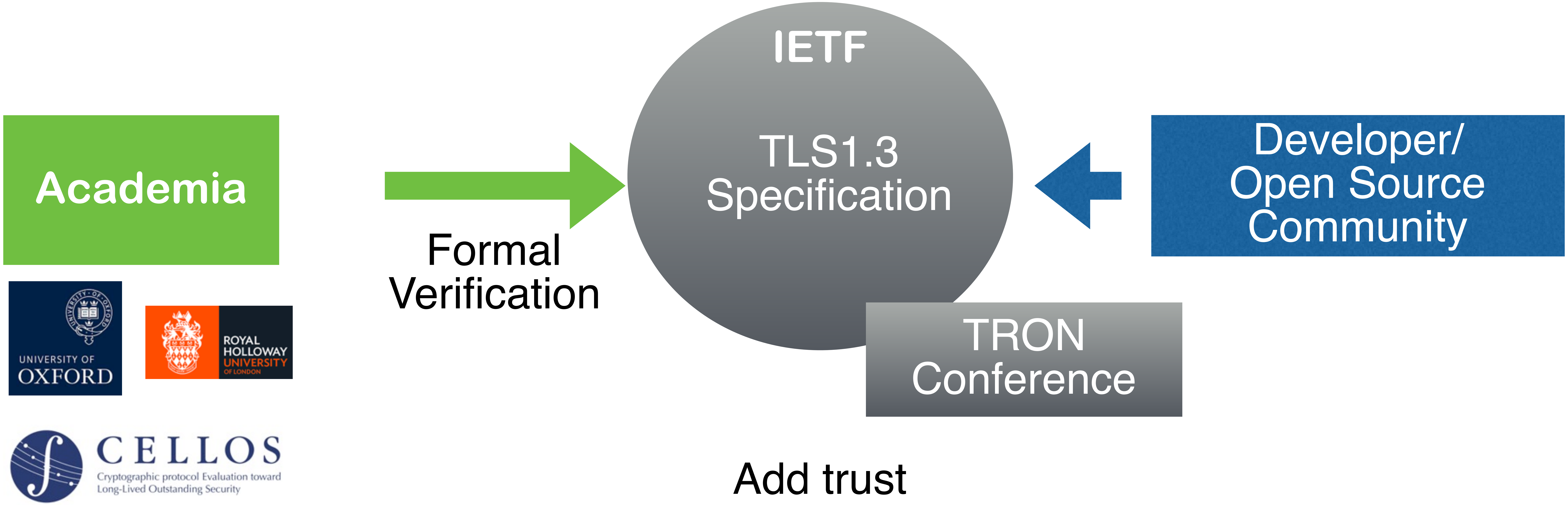
No security proof

No procedure for verification of technology.

No experts on the verification of cryptographic protocols

Insufficient quality assurance of program code

The case of TLS 1.3 [6]



International Standard: ISO/IEC 29128

Accuracy →

Protocol Assurance Level	PAL1	PAL2	PAL3	PAL4
Protocol Specification	PPS_SEMIFORMAL	PPS_FORMAL	PPS_MECHANIZED	
Adversarial Model	PAM_INFORMAL	PAM_FORMAL	PAM_MECHANIZED	
Security Property	PSP_INFORMAL	PSP_FORMAL	PSP_MECHANIZED	
Self Assessment Evidence	PEV_ARGUMENT	PEV_HANDPROVEN	PEV_BOUNDED	PEV_UNBOUNDED

Security consideration for Smart contract

Need completeness and soundness as an application logic

The DAO case was caused by bug

Checking program code is well-known application of formal analysis

Language for Smart Contract

Solidity
Flexible and General purpose language

Bhargavan et al. proposed a framework to analyze both the runtime safety and functional correctness of a Solidity contract [9]

Introducing intermediate functional programming language suitable for verification

At this time, not covered all EVM functionalities

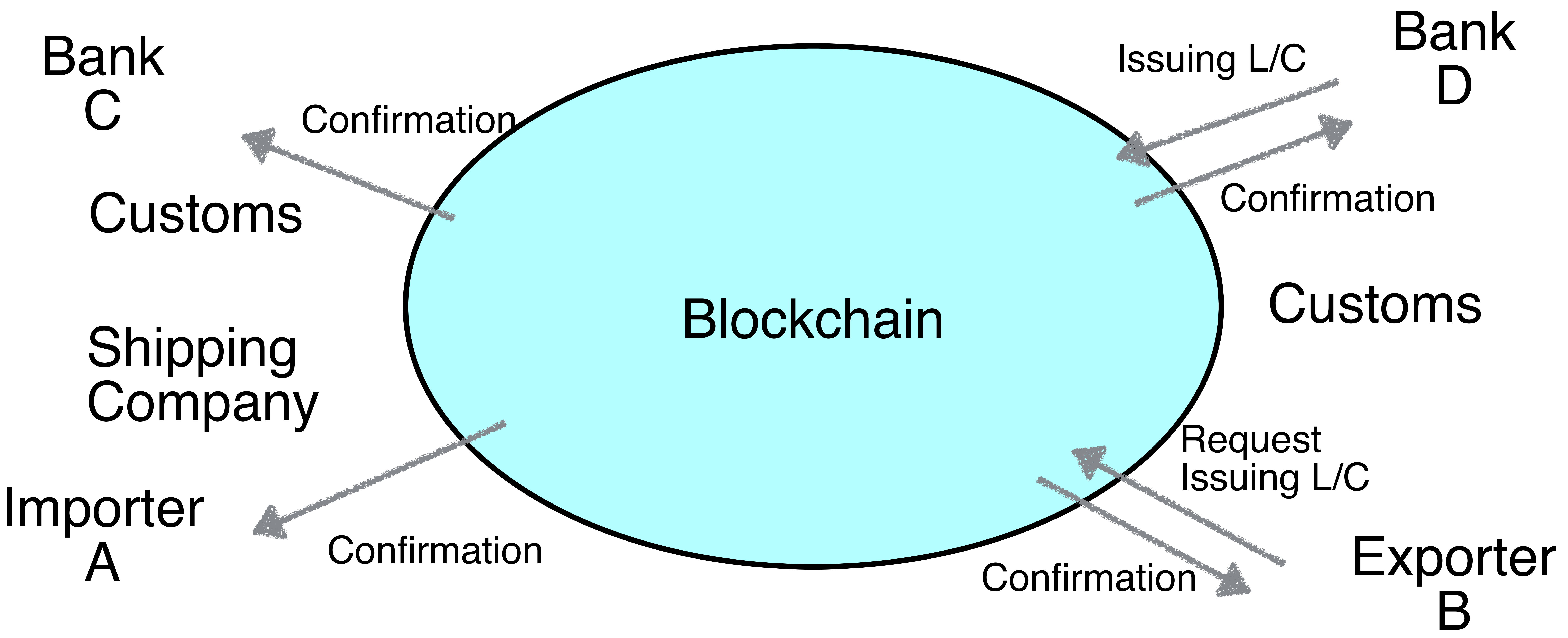
Designing Domain Specific Language

To limit possible execution states, which include "insecure" state, create new domain specific language

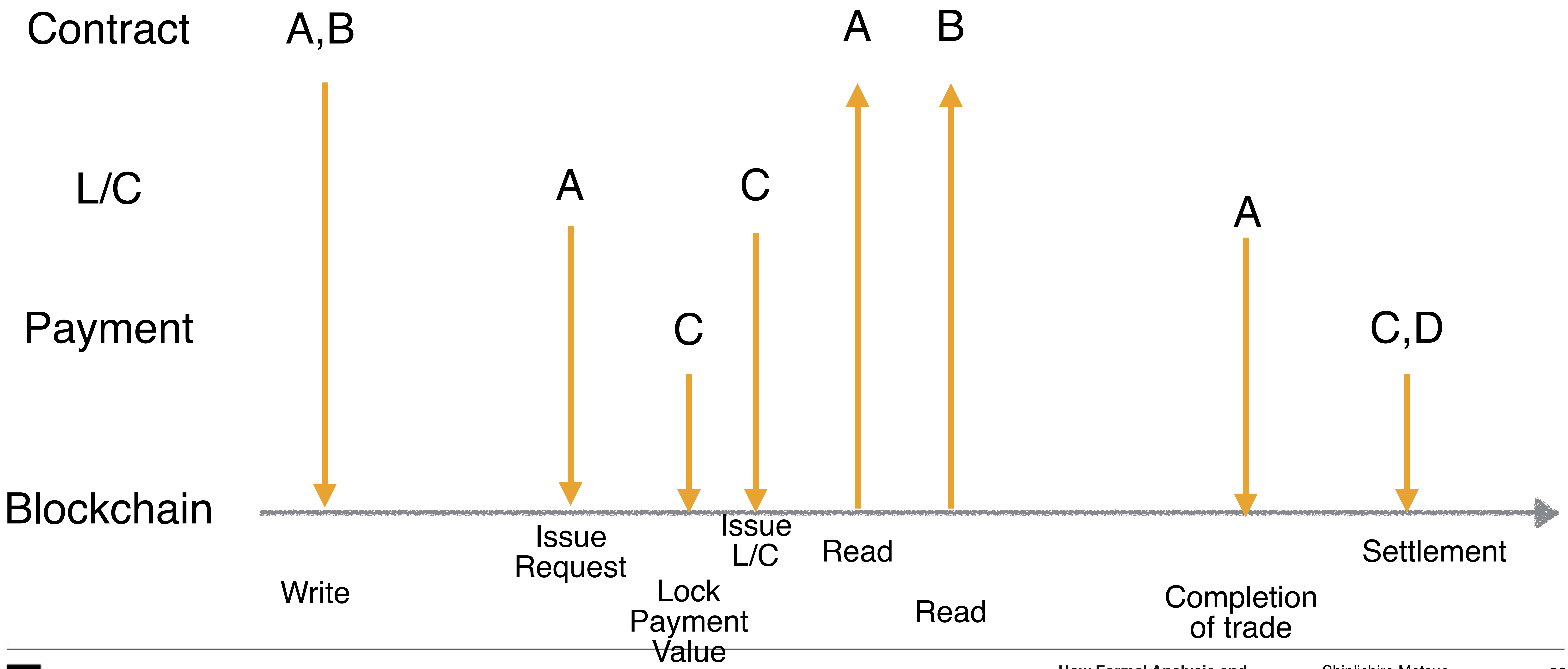
Has enough capability to write business logic

Suitable for formal verification

Letter of Credit (L/C) and Trade Finance over Blockchain



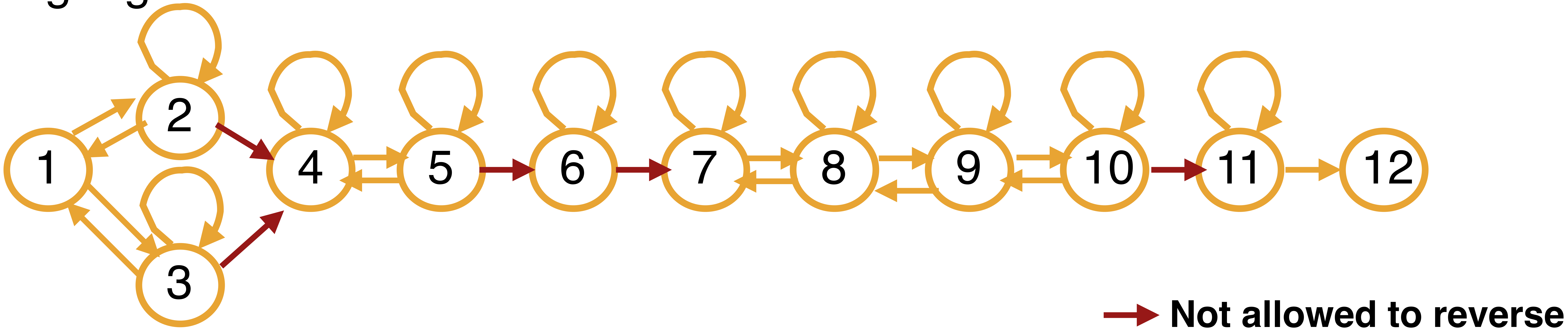
Sequence of process



State Transitions of common process of L/C

Four variables for state representation: Contract, L/C, Payment, Shipment

Create language and execution environment from state transitions and constraints



	1	2	3	4	5	6	7	8	9	10	11	12
L/C	Init	Init	Init	Init	Issue Req	Issue Req	Issued	Issued	Issued	Confirmed	Confirmed	Confirmed
Cash	Init	Init	Init	Init	Init	Cash Lock	Cash Lock	Cash Lock	Cash Lock	Cash Lock	Settled	Settled
Goods	Init	Init	Init	Init	Init	Init	Init	Shipped	Received	Received	Received	Received
Contract	Init	A signed	B signed	Both	Both	Both	Both	Both	Both	Both	Both	Fin

Limitation of Formal Verification

Limitation of automated tool
Upper bound of memory, …
Not sufficient for complicated protocols

How can we verify the correctness of formalization?

Formal verification does not assure the security in most cases

Need templates and languages which are suitable for formal verification

Conclusion

Applicability of Formal Analysis and Formal Verification

Current activities can help four layers of Blockchain Security

Possibility to define specific language for Application Logic Layer