How formal analysis and verification add security to blockchain-based systems

Shin'ichiro Matsuo
Keio University and BSafe.network

Extended Abstract of Tutorial Talk

***Abstract*—Blockchain is an integrated technology to ensure keeping record and process transactions with decentralized manner. It is thought as the foundation of future decentralized ecosystem, and collects much attention. However, the maturity of this technology including security of the fundamental protocol and its applications is not enough, thus we need more research on the security evaluation and verification of Blockchain technology This tutorial explains the current status of the security of this technology, its security layers and possibility of application of formal analysis and verification.**

***Index Terms*—Blockchain, Security Evaluation, Formal Method, Formal Verification, Domain Specific Language**

I. Introduction

A. Background

There are proposed many applications which aim to use blockchain technology as a fundamental distributed ledger. We expect considerable commercial interest in many new and novel applications using a blockchain. In spite of this burgeoning interest, academic research on the security model of blockchain technology and its application are at an early stage. Due to the Ethereum DAO debacle, the importance of analysis of the security of blockchain-based systems is rapidly increasing. Current research issues are to find a good framework to analyze the security of blockchain technology including defining the security requirements and the way to evaluate their security. Several existing researchers deal with how to figure out the security of blockchain by using formal analysis. To facilitate this direction of research, we need a more well-organized framework.

B. Structure of the tutorial

In this tutorial, we firstly figure out the security requirements needed for blockchain based systems and smart contracts. Then we propose technology layers for such systems and application and security considerations for each layer. Next we explore the applicability of formal analysis for each layer and pick three layers which are good targets of evaluation by formal analysis. Then, we propose the framework of applying formal analysis to help secure blockchain-based systems. An explanation of the limitations of formal *verification* follows. At the end of this tutorial, we conclude the direction to the framework to design application code and system which facilitate formal analysis and formal verification.

II. Security requirements for system and smart contract

The security definition of blockchain backbone protocol was proposed in [1], [2]. This security definition focuses on the difficulty of forgery of the block by introducing CommonPrefix property and Chain Quality property. By using these properties, we can estimate the probability which the adversary succeeds to manipulate the blockchain. This is the requirement only for protocol specification of the backbone protocol. From the system and application viewpoints, we should care about more aspects of security. Even on the protocol security, there are many assumptions in achieving its security goals. Cryptographic protocol assumes that the private cryptographic keys are kept secret at all nodes. We should analyze if the assumption surely holds.

For the application logic, there is possibility that some critical bugs remain in the program code. An adversary takes advantage of this bug to attack the application based on blockchain. The Ethereum DAO case gives us an important study that such attack may cause a rollback and a hard fork.

From above, we should cover not only the security requirements for backbone blockchain protocol, but also all mechanisms to ensure the assumptions and scripting language and codes to realize blockchain-based applications.

III. Security layers

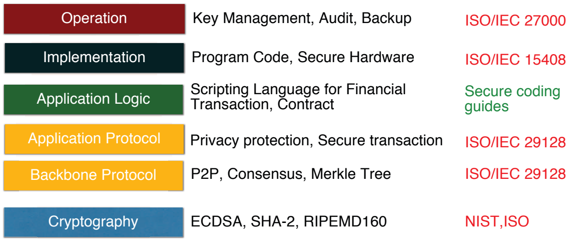
A. Technology layers and security consideration

In [3], Croman et al, proposed the technology layer of blockchain technology. This layers consist of network plane, consensus plane, storage plane, view plane and side plane. This structure is made to rethink the technology to provide more scalability.

From system and application security viewpoints, we set the technology layers by the target of evaluation. They consist of cryptography layer, backbone protocol, application protocol, application logic, implementation and operation (see Fig.1).

As this figure shows, each layer has international standards to analyze the security of the security mechanisms, except application logic. Cryptography layer is covered by standardization process of ISO, NIST and many effort by the cryptographic academic community. Security of backbone protocol is analyzed by using formal analysis and UC (Universal Composability) framework and ISO/IEC 29128 [8]. The security of implementation is certified by Common Criteria

ISBN: 978-0-9835678-7-5. Copyright owned jointly by the authors and FMCAD, Inc.



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

Fig. 1. Technology layers and security consideration

(ISO/IEC 15408) [7] and operation of the system is defined and audited using ISMS and the framework of ISO/IEC 27000 series. Unfortunately, the application logic layer, which contains a scripting language for financial transaction and contract, does not yet have good standard to provide security analysis. Further research here is clearly required.

IV. Applying formal analysis

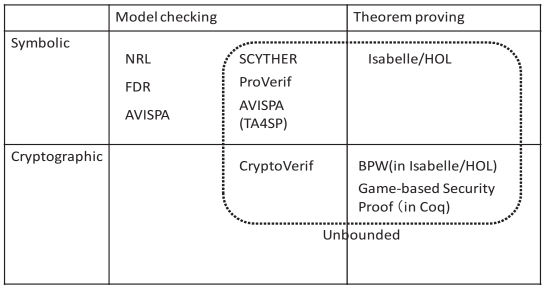
A. Abstract of formal analysis and formal verification

Here, we revisit the basis for the formal analysis and formal verification. Note that we distinguish between these two words. Formal analysis means 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). Is the description of the state spaces, axioms and changes both necessary and complete? Formal verification means to verify the correctness of the specification of the protocol, products or system formal methods such as automated axiomatic theorem proving or model checking. Formal analysis means a manner to use a mathematical formalization to evaluate the security and formal verification means checking if the specific protocol, product or system is qualified against the formal specification.

Formal analysis was originally used for check the existence of a bug in the circuit. Then it is applied to check the existence of bug in software code, design of the software and information system and security of cryptographic protocols.

B. State of formal analysis and checking Tools

The term formal methods refers to the use of methods for the mathematical modeling, calculation, and predication in the specification, design, analysis, construction, and assurance of hardware and software systems. These methods are distinguished as having a well-defined syntax, a semantics, and often a deductive system (or other machinery) for making semantically-sound statements about systems specified in the language of the formal method. Over the last two decades, the security community has made substantial advances in developing automated formal methods for analyzing cryptographic protocols and thereby preventing the kinds of attacks mentioned above. These methods and tools could be categorized by several points of view. Here we categorize them by “Symbolic versus Cryptographic”, “Bounded versus Unbounded”, and “Model checking versus Theorem proving” as Fig. 2.



	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	

Fig. 2. Categorization of Formal Analysis for cryptographic protocol

C. Which security layer can formal method be applied?

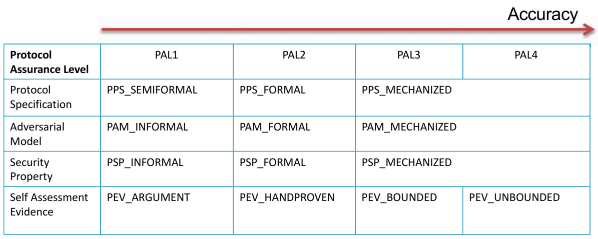
According to the past results and history of formal analysis, the following three layers are main targets of evaluation for formal analysis.

1) Implementation: This layer contains both software and hardware implementation of security mechanisms including cryptographic algorithm, protocols and key management mechanisms. Especially, crypto-token wallet programs used in general user device may become the weakest link and should be carefully implemented. In ISO/IEC 15408, there are seven EALs(evaluation assurance levels), and EAL 6 requires semiformal analysis on the design and implementation, and EAL 7 requires fully formal analysis on the design and implementation. There are many past examples and result of formal analysis in this layer.

2) Backbone protocol and application protocol: Formal analysis on the protocol specification has a long history and it gives many results to enhance the security of cryptographic protocols. ISO/IEC 9798 and 11770 are revised from results of formal analysis [4], [5]. Recently, formal analysis on TLS1.3, the latest version of TLS protocol, helps its sound development and the result is used in the IETF standardization process [6]. Recently, combination of mathematically rigorous proof (UC Framework and game-based proof) and formal analysis are used to apply formal analysis to a wider and complicated set of protocols.

3) Language for Smart Contracts: Checking the program code is the well-known application of formal analysis and we have extensive research in this area. It is not easy to check the highly complicated program by using formal analysis, there are many existing research to realize security assured language specification. For smart contracts, we will have good application by specifically defining new languages that are designed to lend themselves to formal analysis and verification.

ISBN: 978-0-9835678-7-5. Copyright owned jointly by the authors and FMCAD, Inc.



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

Fig. 3. PALs in ISO/IEC 29128

V. Proposal of the framework

A. Implementation

We can apply the same framework and methodology as Common Criteria (ISO/IEC 15408). Especially, wallet software or hardware should be secure against known attacking methodology like gray-box attack (side-channel attack) and white-box attack for software-only implementation. FIPS140-2 is also useful to make the framework for analyzing implementation. In this tutrial, we will provide past examples of formal analysis on the cryptographic implementation and how we can apply it to blockchain-based systems and devices.

B. Protocol

We can apply the same framework and methodology as ISO/IEC 29128 (verification of cryptographic protocols). It defines four PALs(Protocol Assurance Levels) according to the level of formalization for protocol specification, security requirements and operational environment. This framework covers combination of mathematical rigorous proof and formal analysis. In this tutrial, we will provide past examples of formal analysis on protocol specifications, how we write the report to align to this standard, and how we can apply it for analysis on backbone protocol and application protocol.

C. Language for smart contract

Analyzing the existence of bug in the program code is still fundamental research topic in computer science. We still do not have perfect results for general purpose language. The main problem is the openness of general purpose of programming language. As for the smart contract, Bhargavan et al. proposed a framework to analyze and verify both the runtime safety and the functional correctness of a Solidity contract by introducing an intermediate functional programming language suitable for verification [9]. Although the paper does not cover all EVM functionality at the time of writing this tutrial abstract, it seems a good approach to add limitation to operational environment to facilitate formal analysis.

In this tutrial, we additionally propose another approach to define a domain specific language for certain application domain, which has enough capability to write business logic and also suitable for formal verification. Then, we will present an example of the domain specific language for trade finance and trade facilitation.

VI. Limitation of formal verification and how we facilitate the use of it

In this part of the tutrial, we discuss about the limitation of the formal verification. Automated and tool-aided formal verification is strong approach to check the correctness of specification and code. However, there are two major issues when we use such automated tool. The first is on the limitation of the time and memory of the computer which executes the verification. In many formal methods, the tool finds the possibility of bug and security problems by exploring as many execution states as possible. In this case, the upper bound of runtime memory of the computer and execution time become the essential limitation for complicated programs and protocols. While there are many techniques to reduce the number of states to be explored, they are not generally sufficient for complicated software implemented in a general programming language.

The second issue is the correctness of the formalization. When we use the formal verification tool, we formalize the specification (code), security goals and operational environment. The result of execution of the tool depends on the accuracy of the formalization. However, we do not have a good tool check the accuracy. For arbitrary formalized systems, we need to check the correctness by reviewing the formalized code by humans. This limits the applicability of formal verification in general. Here, we need some kind of templates and code patterns in formalization.

From above perspective, limiting the number of states by tightly defining the language and preparing code patterns or templates are good direction to facilitate the use of formal analysis and formal verification. As for the implementation, the protection profile is the actual template for formalization. In the verification of cryptographic protools, there already exists evaluation reports which aligns to ISO/IEC 29128 and they can be used as templates. As for the language for smart contract, defining a domain specific language helps to reduce the number of states to be explored and creates a template of formalization.

VII. Conclusion

In this tutrial, we proposed the way to facilitate the application of formal analysis and formal verification by considering technology layers and their security concerns. We picked three layers, implementation, protocol and language, as targets of applications of formal analysis. Then, we propose a framework to apply formal analysis to each layer by using existing standards and results. We can use the same framework as

ISBN: 978-0-9835678-7-5. Copyright owned jointly by the authors and FMCAD, Inc.

ISO/IEC15408 for implementation and ISO/IEC 29128 for protocol analysis. For the language, which was essential problem with the Ethereum DAO issue, defining a domain specific language is the new and effective solution and we showed an example for trade finance and trade facilitation. The domain specific language should have a design framework which facilitates formal analysis and, if possible, formal verification.

From the above, formal analysis research and technology development can deliver immediate value to the investments in blockchain technology with mutual benefits to all involved.

REFERENCES

[1] J. Garay, A. Kiayias and N. Leonardos, "The Bitcoin Backbone Protocol: Analysis and Applications," Proceedings of Eurocrypt 2015.

[2] R. Pass, L. Seeman and A. Shelat, "Analysis of the Blockchain Protocol in Asynchronous Networks," IACR ePrint Archive, https://eprint.iacr.org/2016/454.pdf

[3] K. Croman, C. Decker, I. Eyal, A. Efe Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. Gün Sirer, D. Song, and R. Wattenhofer, "On Scaling Decentralized Blockchains," Proceedings of Bitcoin Workshop 2016.

[4] C. Cremers and M. Hovert, "Improving the ISO/IEC 11770 standard for key management techniques," International Journal of Information Security, November 2016, Volume 15, Issue 6, pp 659?673.

[5] D. Basin, C. Cremers, S, Meier, "Provably repairing the ISO/IEC 9798 standard for entity authentication," Journal of Computer Security - Security and Trust Principles, Volume 21 Issue 6, November 2013, Pages 817-846.

[6] K. Paterson and T. van der Merwe, "Reactive and Proactive Standardisation of TLS," In Proc. of SSR 2016.

[7] "Information technology – Security techniques – Evaluation criteria for IT security – Part 1: Introduction and general model," ISO/IEC 15408-1:2009

[8] "Information technology – Security techniques – Verification of cryptographic protocols," ISO/IEC 29128:2011

[9] K. Bhargavan, A. Delignat-Lavaud, C. Fournet, A. Gollamudi, G. Gonthier, N. Kobeissi, A. Rastogi, T. Sibut-Pinote, N. Swamy and S. Zanella-Béguelin "Short Paper: Formal Verification of Smart Contracts," http://www.cs.umd.edu/ aseem/solidetherplas.pdf

ISBN: 978-0-9835678-7-5. Copyright owned jointly by the authors and FMCAD, Inc.