Formal Verification of Blockchain-based IoT Applications

Jaime Arias, Kaïs Klai, and Carlos Olarte

LIPN, CNRS UMR 7030, Université Sorbonne Paris Nord, France

Context. *Internet of Things* (IoT) and *blockchain* are both emerging technologies with great potential where security arises as a major concern that has hindered their large-scale deployment. On one hand, the IoT is a network of low-resource devices equipped with sensors that collect and exchange information about the physical environment. Usually, IoT ecosystems rely on a model where the communications, authentication, and authorizations are conducted by a central trusted authority, leaving them vulnerable to a wide variety of attacks [9]. On the other hand, the *blockchain* [7] is a transparent, decentralized, distributed and immutable ledger that allows for peer-to-peer transactions in an environment that does not require mutual trust. It solves the problems of high cost, low efficiency, and insecure data in third parties. One of the most relevant applications of blockchain technology is *Smart Contracts* (SCs). An SC is a digital agreement (*i.e.,*piece of code) between two or more parties, which is executed automatically on a computer without the intervention of a centralized third party. SCs can be applied in many scenarios such as e-commerce, electronic voting, healthcare services, and so forth. Thus, exploiting contract vulnerabilities can lead to terrible losses [13]. For instance, one of the most infamous attacks on Ethereum was the one exploiting a reentrancy vulnerability in the DAO (a Decentralized Autonomous Organization built upon Ethereum) and resulting in 3.6M of stolen Ether. Thus, to satisfy high performance, scalability, *correctness* and *security* requirements, the blockchain SCs need to be well-designed and *verified* before their deployment on a blockchain. The combination of blockchain and IoT has broad potential for the creation of a marketplace of services between devices, and gives the opportunity to create value from collected data. The growing number of emerging blockchain protocols, partnerships and IoT device providers, already indicates that there is a good fit for blockchain in the IoT sector.

This project aims at combining two formal methods widely studied in the LoVe team at LIPN, thus proposing a robust framework for the analysis and verification of IoT and SCs applications. These methods include Coloured Petri Nets (CPN) [8] and Rewriting Logic (RL) [12]. The former is usually equipped with very efficient procedures and the latter is more flexible as a modelling language, where distributed systems are specified via algebraic data types and conditional rewrite rules. As a result of the interaction of these two formal methods, we expect to combine the best of the two worlds: to equip RL with more efficient verification techniques and to explore how RL models may inspire further analyses in CPN.

Objectives. The objectives of this PhD project are depicted as yellow boxes in Figure 1 and described below. The remaining boxes are part of a general verification framework currently being developed at LIPN, to which this project is associated.

O1. Design and implement an IoT model checker.
IoTs applications can be designed in Node-RED (`https://nodered.org/`), a visual language that allows for easily describing the communication and interactions between the different components in the network. The first objective of this project is to propose two models for the formal analysis of Node-RED programs: one based on CPNs and one based on RL. The former will open the possibility of applying efficient decision procedures for verification. The obtained model will be supplied to the Helena model checker [3] in order to check specific temporal properties. The latter will serve as input to Maude [1], a high-level system supporting RL. Besides providing a more declarative model, RL and Maude will open the possibility of performing verification tasks using symbolic techniques [2] such as rewriting modulo SMT.

O2. SOGs for CPNs and RL.
The symbolic observation graph (SOG) [6] strives at taming the state space explosion problem during Model Checking. For that, states are aggregated according to the formula to be verified and a subset of *observable* transitions. In the context of RL, similar techniques, inspired by partial order reduction, have been considered [4] but restricted to state-based observations (and not event-based ones). In order to endow our framework with the capacity of verifying larger specifications, we aim to: (i) propose a SOG version for CPNs; and (ii) define a theory transformation for building the SOG from RL specifications. Hence, we shall integrate into Maude the efficient SOG-based model checkers proposed in [11, 10].

O3. Checking security properties of Blockchain-based IoT applications.
Based on steps **O1.** and **O2.**, we expect to extend the work presented in [5] to detect and check vulnerability properties as well as specific temporal properties of Blockchain-based IoT applications.

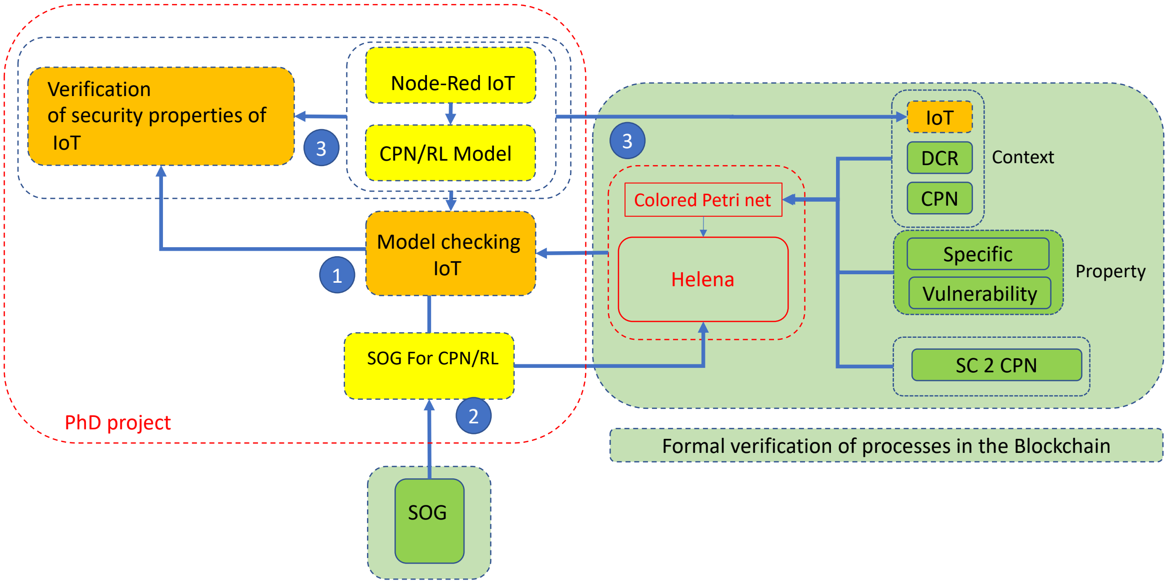

Fig. 1. PhD project

References

1. Clavel, M., Durán, F., Eker, S., Lincoln, P., Martí-Oliet, N., Meseguer, J., Talcott, C.L. (eds.): All About Maude - A High-Performance Logical Framework, How to Specify, Program and Verify Systems in Rewriting Logic, Lecture Notes in Computer Science, vol. 4350. Springer (2007)
2. Durán, F., Eker, S., Escobar, S., Martí-Oliet, N., Meseguer, J., Rubio, R., Talcott, C.L.: Programming and symbolic computation in maude. J. Log. Algebraic Methods Program. **110** (2020)
3. Evangelista, S.: High level petri nets analysis with helena. In: Applications and Theory of Petri Nets 2005. pp. 455–464. Berlin, Heidelberg (2005)
4. Farzan, A., Meseguer, J.: State space reduction of rewrite theories using invisible transitions. In: Johnson, M., Vene, V. (eds.) Algebraic Methodology and Software Technology, 11th International Conference, AMAST 2006, Kuressaare, Estonia, July 5-8, 2006, Proceedings. Lecture Notes in Computer Science, vol. 4019, pp. 142–157. Springer (2006)
5. Garfatta, I., Klai, K., Graïet, M., Gaaloul, W.: Model checking of solidity smart contracts adopted for business processes. In: Hacid, H., Kao, O., Mecella, M., Moha, N., Paik, H. (eds.) Service-Oriented Computing - 19th International Conference, ICSOC 2021, Virtual Event, November 22-25, 2021, Proceedings. Lecture Notes in Computer Science, vol. 13121, pp. 116–132. Springer (2021)
6. Haddad, S., Ilié, J., Klai, K.: Design and evaluation of a symbolic and abstraction-based model checker. In: Wang, F. (ed.) Automated Technology for Verification and Analysis: Second International Conference, ATVA 2004, Taipei, Taiwan, ROC, October 31-November 3, 2004. Proceedings. LNCS, vol. 3299, pp. 196–210. Springer (2004)
7. Hewa, T.M., Ylianttila, M., Liyanage, M.: Survey on blockchain based smart contracts: Applications, opportunities and challenges. J. Netw. Comput. Appl. **177**, 102857 (2021)
8. Jensen, K., Kristensen, L.M.: Coloured Petri Nets - Modelling and Validation of Concurrent Systems. Springer (2009)
9. Khan, M.A., Salah, K.: Iot security: Review, blockchain solutions, and open challenges. Future Gener. Comput. Syst. **82**, 395–411 (2018)
10. Klai, K., Abid, C.A., Arias, J., Evangelista, S.: Hybrid parallel model checking of hybrid LTL on hybrid state space representation. In: Nouri, A., Wu, W., Barkaoui, K., Li, Z. (eds.) Verification and Evaluation of Computer and Communication Systems - 15th International Conference, VECoS 2021, Virtual Event, November 22-23, 2021, Revised Selected Papers. Lecture Notes in Computer Science, vol. 13187, pp. 27–42. Springer (2021)
11. Klai, K., Poitrenaud, D.: MC-SOG: an LTL model checker based on symbolic observation graphs. In: Petri Nets. LNCS, vol. 5062, pp. 288–306. Springer (2008)
12. Meseguer, J.: Conditioned rewriting logic as a united model of concurrency. Theor. Comput. Sci. **96**(1), 73–155 (1992)
13. Samreen, N.F., Alalfi, M.H.: A survey of security vulnerabilities in ethereum smart contracts. In: 30th Annual International Conference on Computer Science and Software Engineering. p. 73–82. CASCON '20, IBM Corp., USA (2020)