UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science
Computer Science Curriculum

Alice Cooper

Type Inference for Fourth Order Logic Formulae

Master's Thesis (30 ECTS)

Supervisor(s): Axel Rose, MSc
May Flower, PhD

Tartu 2023

Type Inference for Fourth Order Logic Formulae

Abstract:
Many interpreting program languages are dynamically typed, such as Visual Basic or Python. As a result, it is easy to write programs that crash due to mismatches of provided and expected data types. One possible solution to this problem is automatic type derivation during compilation. In this work, we consider study how to detect type errors in the WHITESPACE language by using fourth order logic formulae as annotations. The main result of this thesis is a new triple-exponential type inference algorithm for the fourth order logic formulae. This is a significant advancement as the question whether there exists such an algorithm was an open question. All previous attempts to solve the problem lead lead to logical inconsistencies or required tedious user interaction in terms of interpretative dance. Although the resulting algorithm is slightly inefficient, it can be used to detect obscure programming bugs in the WHITESPACE language. The latter significantly improves productivity. Our practical experiments showed that productivity is comparable to average Java programmer. From a theoretical viewpoint, the result is only a small advancement in rigorous treatment of higher order logic formulae. The results obtained by us do not generalise to formulae with the fifth or higher order.

Keywords:

List of keywords

CERCS:

CERCS code and name: `https://www.etis.ee/Portal/Classifiers/Details/d3717f7b-bec8-4cd9-8ea4-c89cd56ca46e`

Tüübituletus neljandat järku loogikavalemitele

Lühikokkuvõte:

One or two sentences providing a basic introduction to the field, comprehensible to a scientist in any discipline.

Two to three sentences of more detailed background, comprehensible to scientists in related disciplines.

One sentence clearly stating the general problem being addressed by this particular study.

One sentence summarising the main result (with the words “here we show´´ or their equivalent).

Two or three sentences explaining what the main result reveals in direct comparison to what was thought to be the case previously, or how the main result adds to previous knowledge.

One or two sentences to put the results into a more general context.

Two or three sentences to provide a broader perspective, readily comprehensible to a scientist in any discipline, may be included in the first paragraph if the editor considers that the accessibility of the paper is significantly enhanced by their inclusion.

Võtmesõnad:

List of keywords

CERCS:

CERCS kood ja nimetus: `https://www.etis.ee/Portal/Classifiers/Details/d3717f7b-bec8-4cd9-8ea4-c89cd56ca46e`

Contents

1 Introduction

One is where and when one claims to be - this is the underlying principle of most of today's location-based services. This principle, however, hides a whole set of preceding premises that implicitly assert trust in the subject's honesty in reporting its correct location. Having this trust delegated, the reliance on a trusted third party, frequently an atomic computing entity, is still subject to tampering, repudiation, inaccuracy, punctual and single failure, or any other kind of Byzantine behaviour.

The trust levels required to testify to one's alibi remain unmeasurable in modern arrangements. Strategic interactions between rational agents often support this trust. One party provides a location-based service, and another party makes use of it for its individual benefit and by providing a non-tampered time-conscious piece of location claim. This interaction appears to be one of a non-zero-sum game that can be observed in most GPS-based services, mapping platforms, navigation systems, mobility and ride-hailing apps, among many others. If driven by the reasoning goal of extracting correct information from the interacting system, users are logically motivated to report an accurate location. The services, having the higher goal of not losing users due to their reported malfunctioning or inaccuracy, are thus motivated to provide maximized quality when operating and consuming the location claims.

This paradigm is now ubiquitous, which may lead to its fallacious use in other very distinct scenarios. Those scenarios are, therefore and inversely, the ones that fundamentally require verifiable proof of location to assert a particular state or derive a conclusion. Consider, for example, scenarios requiring location-based authentication or authorization in adversarial environments that rely on information gathered in a trustless setup. These materialize into services requiring, for instance, a digital certificate as proof that a given user is within a particular geographical area, to enable certain functionalities or assert liability, as in location-based access control, review or reward systems, augmented reality games, social networks, etc... Security against geo-tampering or location spoofing in a relatively trustless environment is needed to achieve the required integrity.

The basic infrastructural concept is somewhat understood, and theoretical or experimental solutions have been delivered throughout the years. These solutions have evolved parallel with their trust assumptions, beginning with a fully trusted setup and progressively shifting towards modern requirements for operational decentralization - of power and profit. Most recent attempts contemplate the need for a permissionless means of reaching consensus between a quorum of witnesses that can attest to one's presence at a given point in space and at a given moment in time. These concepts take shape with a combination of tools: wireless technologies as message-exchanging means, cryptographic protocols as confidentiality, integrity, or authentication enablers, and distributed ledgers as publicly trusted record keepers.

The quest for a solution that could make these location-based services as prevalent

and ubiquitous shall aim to address a set of design challenges. These challenges are, among others, the solution's flexibility and deployability, preferably by making use of existing infrastructure, or at least accessible technology, and the solution's security and privacy, obeying the modern cryptographic standards and requirements, to guarantee some level of privacy, and resiliency to attacks. This thesis, aiming to address these matters, delivers the following contributions:

1. A semi-formalization of the location-based services' paradigm, including the underlying premises and the strategic interactions between rational agents, along with a review of the state of the art in the field. The review is discriminated in terms of trust levels, from fully trusted to permissionless environments, and in terms of the underlying technology, from centralized to decentralized.

2. The design and implementation of a proof-of-concept that can be deployed in a permissionless manner, using existing infrastructure, and that can be employed to attest to one's presence at a given point in space and time. Specifically, the proof-of-concept is based on the use of routing protocols for multi-hop mobile ad hoc networks to set up a mesh network of witnesses that can attest to one's presence in a given geographical area.

The structure of the work is as follows. In chapter 2, an introduction to the underlying concepts and hypotheses is provided, together with a mention to the technology involved in the practical implementation. Chapter 3 examines and analyzes similar work discriminated in terms of trust levels. In chapter 4, a general overview of the requirements for the proposed solution is given. Chapter 5 details the architecture's design, implementation, and evaluation. Finally, chapter 6 presents the conclusion and recommendations for future work.

2 Prelude

This section introduces not only the underlying concepts that sustent the work, but also the technology involved in implementing the proposed proof of concept.

2.1 Proof-of-Location

The problem of attesting to one's location is a fundamental act of metaphysical reasoning that happens everywhere, at every moment. Unconsciously and unwittingly, we do claim to be somewhere at an indiscriminated point in time, and we do expect others to believe us. This act is, however, grounded on informal and implicit levels of trust that are not explicitly asserted, as liability is often not categorically assigned. When it does happen, trust is usually delegated to a trusted third party or distributed between multiple parties, that may be able to testify to one's presence, synchronously, at the very same location. The act of witnessing is, therefore, a regular yet fundamental part of our interactions with physical reality. When we do claim our presence at an event, report our alibi to authorities, or assert our location to a service provider, a protocol for location attestation is implicitly followed. Some may require a physical interaction of any kind, while others may find digitalized and infrastructural means to gather the required location proof [1].

A digital proof-of-location can then be defined as an electronic certificate that assuredly attests one's relative position in both space and time [2]. The relativity of the attestation is, however, not a trivial matter. It is, in fact, a complex and multi-faceted process that requires the simultaneous existence of various untrusted or semi-trusted parties, especially in an environment with no individual honesty guarantees. According to [3], a proof-of-location protocol may be considered secure if complete, spatio-temporally sound, non-transferable and tamper-evident. The system that materially backs the implementation of such a protocol is, therefore, expected to provide fault-tolerance, reliability, and availability guarantees. More advanced protocols may also explore the possibility of providing privacy and anonymity assurances [4], as well as the possibility of being used in a trustless environment [2]. Following is the conceptualization of the common entities of a proof-of-location protocol.

2.1.1 Parties Involved

The general act of witnessing alludes to the simultaneous spatiotemporal existence of a set of entities with distinct roles. The majority of the protocols convey a clear contrast between these roles, highlighting the relative dynamism that differentiates those entities.

In comparable terms, highly dynamic entities do not maintain a fixed geographical location for long periods of time. They are often observed in movement, thereby repeatedly starting and finishing communication procedures with neighbouring entities. On the other hand, static entities are expected not to engage in frequent position changes,

expressing continuous and fairly invariable communication availability around a fixed point in space as time passes [3]. The act is, however, only completed with another type of entity from whom neither the relative staticity nor the relative dynamism frankly matters. These protocol parties are often external and asynchronous to the witnessing process, but they do effectively take a non-negligible part in incentivizing and giving significance to the witnessing act.

Concisely and in concrete terms, these location-proof arrangements expect the existence of a *prover* that engages in any communication protocol with nearby participants, the *witnesses*, with the goal of gathering a verifiable proof-of-location claim, to be later presented to a *verifier*, therefore convincing it of one's existence within a geographical area at a given moment in the past [5].

Prover. A prover is a dynamic entity, both in movement and availability terms, that is expected to be able to communicate with the witnesses, to gather a proof of its location, and to be later able to provide a location claim to the verifier. Communication with nearby witnesses is thought to happen wirelessly, using any short-range message transmission means. Provers are also expected to be associated with a verifiable but desirably private identity, often as a pseudonym.

Witness. A witness is an entity that is expected to be able to communicate with the prover via the same short-range communication channel and to provide it with a verifiable piece of location attestation. The witnesses are envisioned to seldomly change their absolute location and maintain a relatively stable neighbouring list of nearby witnesses. These references aim at attaining the figurative creation of coverage zones as strongly connected graphs that form the boundaries of the atomic units of a polygonal mesh. Witnesses are as well expected to be identified, usually by a pseudonym.

Verifier. A verifier is an external entity that is able to receive a location claim from a prover and verify its validity. Although possible and predicted for trusted setups, in a trustless environment and with the general assurances of a permissionless protocol, verifiers shall not have the need to communicate directly with the witnesses. Verifiers' identity is also of no measurable importance for the protocol, as the interaction between the prover and the verifier is usually asynchronous and external to the witnessing process.

2.1.2 Common Threat Models

As with any technology that involves the collection and processing of sensitive and tamper-prone location data, proof-of-location systems must be designed and implemented with a keen awareness of the threat landscape. The threat models of these systems are complex and multifaceted, encompassing a diverse range of actors, motives, and attack

vectors. In this context, it is crucial to understand not only the technical mechanisms of proof-of-location systems, but also the broader factors that shape their security and privacy risks.

Some common scenarios that may affect the security of proof-of-location systems are, for instance, malicious provers that may attempt to forge location claims, or witnesses that may attempt to collude with other entities to falsify location claims. Adversary efforts may also be observed in the form of malicious provers, or witnesses, that may try to respectively impersonate other peers. Sybil attacks are also on the horizon of possible threats, often employed to disrupt the operation of the system by flooding it with fake participants [3]. Other works have also considered semi-honest adversaries that, despite following the protocol rules, may try to learn additional information from the messages exchanged [5].

2.2 Wireless Mesh Networks

The envisioned fourth industrial revolution has set the track for modern advancements in achieving a global web of pervasive connectivity between all sorts of machines [6, 7]. New means of radio and wireless communication have been pushing for the technological heterogeneity of protocols, architectures, devices, and consequent performance levels in order to find their design suitability for different coverage or range scenarios, transmission or bandwidth rates [8]. Additionally, requirements for more complex, adaptable, and resilient topologies have captured broad interest, in both academic and industry domains [7].

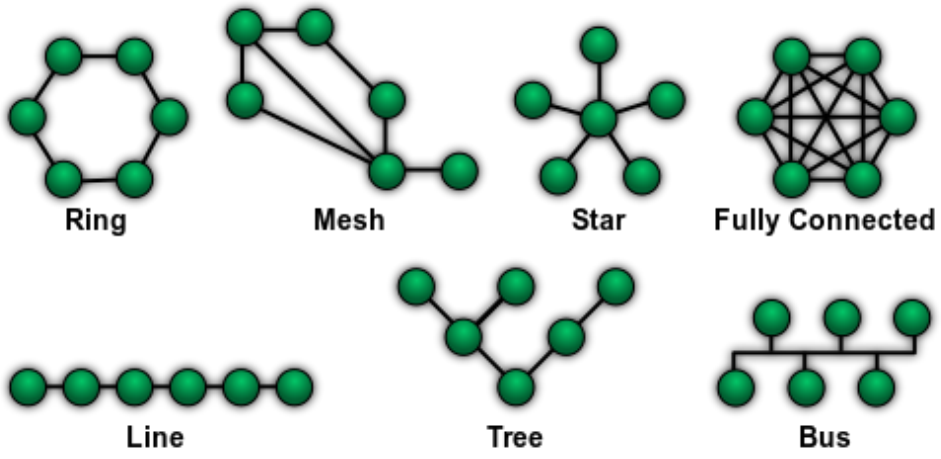

Figure 1. Different topologies of a computer network.

Draw my own image for network topologies.

The development of new hardware architectures, protocols and applications started gaining momentum and branched their way forward to support the popularisation of Wireless Mesh Networks (WMNs). In mesh topologies (see Fig. 1), network nodes are directly and dynamically connected in a non-hierarchical way. This trait eventually allows for many-to-many communications between the devices to efficiently route data from a generic source to a generic destination. The infrastructure nodes that make up the mesh are expected to dynamically self-organize and configure themselves, resulting in beneficial distributed effects on the overall fault tolerance, ease of deployment, and workload allocation [7, 8]. WMNs follow these principles with the particularity of being made up of radio nodes that communicate via any sort of wireless technology.

Some of the most common wireless technologies that have been, throughout the years, ported to WMNs are IEEE 802.11, Bluetooth, IEEE 802.15, and LoRa. The first is the most popular and widely used, being the basis for the Wi-Fi standard, which, at the beginning of the last decade, saw an amendment that mainly targeted mesh networks, the IEEE 802.11s WLAN Mesh Standard [9]. The novelty came with the introduction of routing mechanisms operating at the ISO/OSI Layer 2, allowing for compatible information delivery in the layers above. The dynamic establishment of a topology for IEEE 802.11s-based mesh networks relies on the phased transmission of beacon messages that allow for the discovery, synchronization, and maintenance of the links between the peers. IEEE 802.11s has a default routing protocol, the Hybrid Wireless Mesh Protocol (HWMP), which is based on a series of flooding procedures for both proactive and reactive path finding and selection [10]. This protocol is, however, not strictly enforced by the standard and has been replaced by other, more popular solutions. One notable example is the Better Approach To Mobile Ad-hoc Networks (B.A.T.M.A.N.) routing protocol.

This thesis will explore the concept of WMNs and their potential for serving as the infrastructural topology that enables the relatively short-ranged wireless exchange of messages between the participants of a proof-of-location protocol. The following sections will present the B.A.T.M.A.N. routing protocol, OpenWRT and other relevant tools that will be later used to implement the proof of concept.

2.2.1 B.A.T.M.A.N. Routing Protocol

The Better Approach To Mobile Ad-hoc Networks (B.A.T.M.A.N.)[1] is a proactive routing protocol for WMNs that operates not at the network layer but at the data link layer, by asserting the reliability of radio links using routing metrics and a distance-vector approach [11]. Its newer wireless version, *batman-adv*, has gained traction and popularity and eventually made itself available in the Linux kernel.

[1] http://www.open-mesh.org/

Route discovery is preemptively replaced with neighbour discovery, and each infrastructural node is instructed to calculate its potential best next-hop, significantly reducing the overhead of requiring each peer to be aware of the whole network topology. Its version V introduced a throughput metric to evaluate the links' quality and routing choices, replacing the version IV packet loss-based metric, deemed unsuitable for larger network sizes [11].

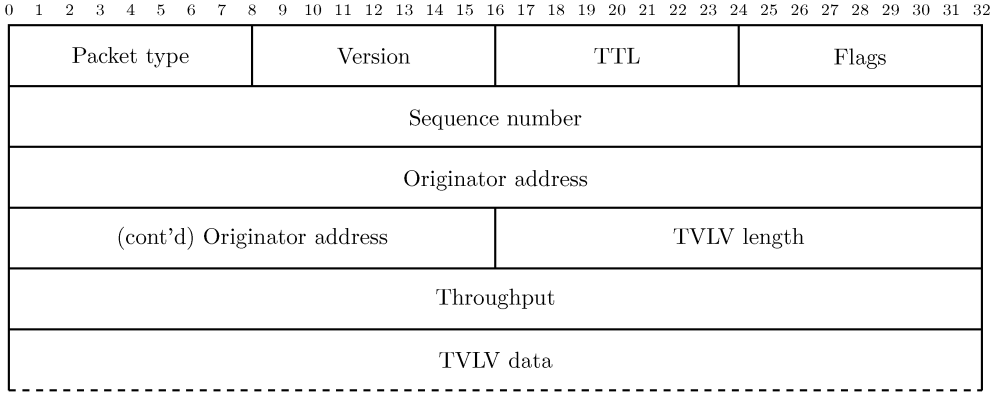

Figure 2. OriGinator Message version 2 (OGMv2) packet format [7, 12].

The discovery of neighbouring nodes is accomplished by broadcasting OriGinator Messages (OGMv2, see Fig. 2), featuring a collision avoidance delay mechanism, the detection of new or duplicate messages, and other fields for throughput measurement and gateway discovery [7]. Hence, the OGM flooding protocol enables mesh routing procedures that, simultaneously but independently, allow for estimating the quality of the individual links. Additionally, the protocol enables OGM aggregations as an effort to reduce the overhead of sending many short-sized frames. Nevertheless, there is still a quest for optimizations that would allow for more efficient use of multiple interfaces. An implementation of a subset of the Internet Control Message Protocol (ICMP) was also made available, allowing, for instance, the use of the *ping* command to test the connectivity between nodes [11].

Building on the previous, the B.A.T.M.A.N. routing protocol has been, through multiple initiatives, successfully blended into the OpenWRT project, which will also be employed in the proof of concept. The following section will present OpenWRT and other relatable tools.

2.2.2 OpenWRT, QEMU, and Raspberry Pis

The OpenWRT project [2] is a Linux distribution for embedded devices, which, in the context of this thesis, will serve as the host operating system for running the proof of concept solution. The project is based on the Linux kernel, encapsulating several of its libraries and packages, and is designed to be used on resource-constrained devices. OpenWRT features not only a writable root filesystem and automation build tools with integrated cross-compiler toolchain, but also a package management system that allows for the installation of additional software. The project also provides extensive configuration options for networking capabilities, which includes enabling mesh networking support through the B.A.T.M.A.N. routing protocol.

To facilitate the development and testing of the proof of concept, the QEMU[3] emulator will be used. QEMU is a generic and open-source machine virtualizer that, through its versatile set of features, allows for the full-system emulation of a wide range of hardware and software. The emulator will run the OpenWRT-generated images and spawn multiple virtual machines. These machines will simulate the various protocol participants by establishing, with the help of the network emulation tools, a fully connected mesh network. The intention is to ease and accelerate the development process by allowing for testing the proof of concept in a controlled environment, without the management, maintenance and deployment hustle of physical devices. Later, the solution will eventually be deployed on a set of Raspberry Pis[4], the most widely used single-board computers for developing IoT solutions.

2.3 Permissionless Consensus

Long has been the time when consensus was still on the verge of being considered such a fundamental problem of distributed systems. Generally defined by Lamport, et al. [13, 14], consensus means reaching an agreement between multiple parties in the potential presence of faulty individuals. As per multi-agent systems, interacting over computer networks, consensus is thought to be the result of a coordination effort, that eventually leads the parties to agree on some value at a given moment. However, the evolution of the consensus problem has been invariably limited by a set of strong assumptions. The well-known Byzantine-Fault-Tolerant multiparty consensus systems, that have been designed over the years, are usually meant to work only with a set of known participants, being them faulty or not [15].

The other side of the coin is the permissionless consensus challenge, consisting of achieving agreement in an environment where the parties are unknown and untrusted [16, 17]. The relative openness and lack of any kind of central authority are other intrinsic

[2] https://openwrt.org/
[3] https://www.qemu.org/
[4] https://www.raspberrypi.org/

particularities of this type of networks, which inevitably adds complexity to the problem. The participants are not only unknown and untrusted but can also join or leave the network at any time, freely choosing if they care to participate in the consensus protocol. Nevertheless, the problem of permissionless consensus is still seen as a special case of the general consensus definition, but under more meticulous trust assumptions.

Further in this thesis, we will evaluate the different high-level proof-of-location protocols and draw a parallel between the evolution of their trust levels and the ultimate need for a low-level permissionless consensus algorithm that allows for establishing decentralized and time-conscious agreement, in an eventual trustless setup, between the multiple witnesses. The next subsections will briefly review some of the most relevant aspects and proof units that give practicality to the roots of the problem.

2.3.1 Proof-of-X

The solution is, nevertheless, unsettled and the scientific community has been reasoning about the need for permissionless consensus when there are already well known and established consensus protocols that work in trusted environments [15, 18]. However, even those protocols have their own limitations, not only in terms of trust, fault-tolerance, centrality, permissions, or bottlenecks, but also in terms of scalability [18], despite assuring deterministic finality [19]. The need for permissionless consensus is then justified by the fact that permissioned protocols are not compatible with the requirements of the new generation of distributed systems, especially in the context of Blockchain networks. These requirements include dealing with today's sparse networks of anonymously and dynamically participating devices, without interrupting consensus and while battling Sybil attacks [20, 21]. Fundamentally, the permissionless consensus problem is the need for a consensus protocol that can be run in a distributed and decentralized environment, where the participants are unknown and untrusted, and where the network is bigger, sparser and unpredictably less reliable.

Technically, permissionless environments allow for larger networks that depict lower connectivity between the participants. Operationally, everything is expected to happen in an asynchronous or partially synchronous fashion, and the number of transactions is predicted to be smaller than in the permissioned counterparts. Participation is free, and the governance is not centralized, but rather distributed and public. The identity of the participants is secured or semi-secured as it often relies on pseudonymity for protecting the nodes' identity, enabling, at the same time, full transparency concerning the rest of the network's content and operation [22]. Expectedly, the goal of permissionless consensus, as for any consensus protocol, is to reach agreement on a single value, or a set of values. However, due to the nature of the protocols, the values that are agreed upon end up establishing the serialization of the transactions, and so establishing time consciousness and total order of the events [20].

Also described in [21], very concisely, the way to achieve an operating protocol, as

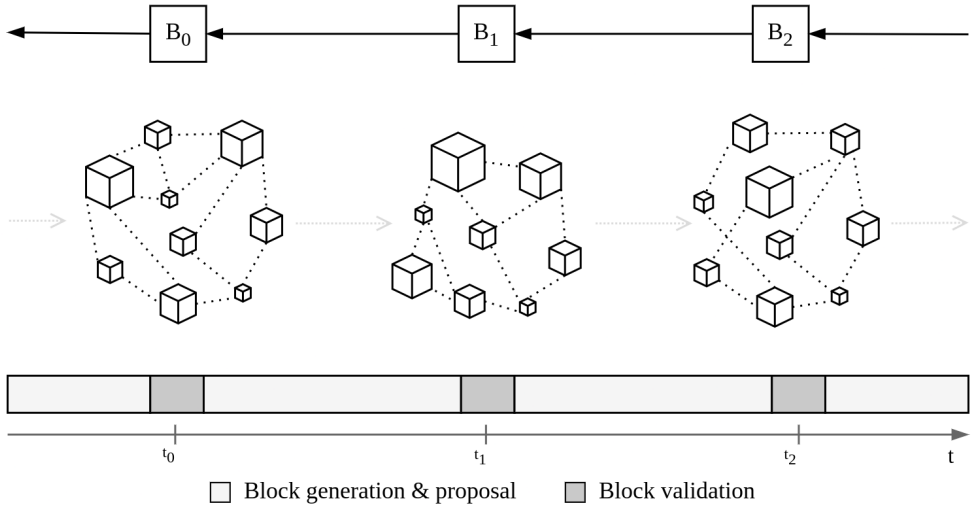

Figure 3. An illustration of the permissionless consensus building blocks - the block generation and proposal phases followed by the block validation, along with frequent network topology changes and the consequent serialization of the information.

seen in the mainstream Blockchain networks, is by first generating the agreeable value, in this particular case, a block and its proof, disseminating the information to the network, followed by the eventual validation and acceptance of the block by the rest of the nodes. This is the approximate moment when consensus is reached (See Fig. 3). During the whole process, a fair and somewhat predictable incentive mechanism is also needed, that rewards participants for their honest effort in reaching consensus, and punishes the ones that are not behaving correctly. These incentives are of major importance in this very context of permissionless consensus and all these building blocks form the basis of the inner functioning of Bitcoin itself [16], replicated with some variations in other permissionless networks [17]. The following is a short introduction to some relevant proof units that feature in the most popular Blockchain systems.

2.3.2 Proof-of-Work and Proof-of-Stake

Without discrediting the previous attempts, the first practical permissionless consensus algorithm was proposed by Nakamoto in [16]. It is a Proof-of-Work consensus protocol that resembles a replicated state machine where the independent participants reach agreement not only about transactional values, but also about their order - naturally forming the underlying structure of what is now known as a Blockchain. The focus shifted for decentralized systems and after Proof-of-Work many other consensus mechanisms have been proposed, relying on different consensus units.

In the classical Nakamoto consensus protocol, the generation of a block, to be proposed for further network agreement, complies with the unit of computational work needed to create, or rather find, a verifiable proof of the effort spent on assembling the block [16]. This essentially requires brute forcing the search for a cryptographic hash value for the aggregation of the block information with a nonce. This value has to satisfy a difficulty threshold (See Procedure 1), which gets adjusted dynamically over time, to maintain the network overall requirement for the block generation interval [20, 21].

Procedure 1: BlockGeneration

Input: Transaction Merkel Tree Root, Hash of the last Block, Timestamp, Other.
Result: new $Block$.

$BlockHeader \leftarrow$ Transaction Merkle Tree Root
    | Hash of the last Block
    | Timestamp
    | Other;

// the preceding zero bits in $target$ depict the mining difficulty
while $Hash(BlockHeader \mid nonce) \geq target$ **do**
    Increment $nonce$;

// append transactional data
return new $Block$;

One can then exercise the reasoning line and extrapolate the previous block generation mechanism to a *Proof-of-Something* pseudo-random competition in which an entity in possession of a higher amount of a certain resource, either computational power, or stake, or certain currency, or, for instance, a higher amount of storage space, guarantees a higher probability of leading the block generation and proposal, and consequently winning the acceptance by the majority. This is the essence of Proof-of-Stake, as a derivative of the Proof-of-Work mechanism. Here, stake is a traceable and verifiable amount of a certain unit, token or currency, that is owned by a certain entity who wishes to participate in the consensus protocol. The stake works as a form of collateral that is used to guarantee everyone's honesty, in an attempt to reduce the Sybil attack likelihood. And, respectively as in Proof-of-Work with computational power, the higher the stake, the higher the probability of leading the block generation and proposal.

Idealized and inspired by Proof-of-Stake, extending or adapting Proof-of-Work became a popular trend in the Blockchain community. The main idea is to replace the computational power with some other resource, that is more scarce, or more valuable, or more verifiable, or more traceable, to combine multiple resources, or even to add extra requirements to pure Proof-of-Work [21]. Not that every one of the options has a considerable potential for entirely solving the permissionless consensus problem, but

each one of them may tackle different use cases where consensus needs to be reached, and where different resources are available to make the agreement happen [23, 24]. Nonetheless, the design of these consensus mechanisms shall aim for a protocolar choice between a set of properties that form a trilemma: Security, Scalability, and Decentralization. Briefly put, relaxing the security requirements may allow for more scalability, both of which, consequently, have hands tied with decentralization. These trade-offs are of practical consideration when defining the network goals and use cases [21]. Further dissection of various classes of Proof-of-Stake based protocols, diverging alternatives to the classic Nakamoto consensus, and comparisons between them can be found in [20, 21, 23–25].

With all of the above in mind, we will proceed to review some of the proposed proof-of-location solutions, discriminated by trust levels. Aiming at achieving spatiotemporal agreement among the witnesses, we will reason about the applicability of one of these lower-level permissionless consensus protocol, in the context of a fully decentralized and trustless environment.

References

[1] W. Luo and U. Hengartner, "Veriplace: a privacy-aware location proof architecture," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 23–32, 2010.

[2] M. Amoretti, G. Brambilla, F. Medioli, and F. Zanichelli, "Blockchain-based proof of location," in *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pp. 146–153, IEEE, 2018.

[3] B. Nasrulin, M. Muzammal, and Q. Qu, "A robust spatio-temporal verification protocol for blockchain," in *Web Information Systems Engineering–WISE 2018: 19th International Conference, Dubai, United Arab Emirates, November 12-15, 2018, Proceedings, Part I 19*, pp. 52–67, Springer International Publishing, 2018.

[4] W. Li, H. Guo, M. Nejad, and C.-C. Shen, "Privacy-preserving traffic management: A blockchain and zero-knowledge proof inspired approach," *IEEE access*, vol. 8, pp. 181733–181743, 2020.

[5] A. Dupin, J.-M. Robert, and C. Bidan, "Location-proof system based on secure multi-party computations," in *Provable Security: 12th International Conference, ProvSec 2018, Jeju, South Korea, October 25-28, 2018, Proceedings*, pp. 22–39, Springer, 2018.

[6] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer networks*, vol. 47, no. 4, pp. 445–487, 2005.

[7] A. Cilfone, L. Davoli, L. Belli, and G. Ferrari, "Wireless mesh networking: An iot-oriented perspective survey on relevant technologies," *Future Internet*, vol. 11, no. 4, p. 99, 2019.

[8] M. L. Sichitiu, "Wireless mesh networks: opportunities and challenges," in *Proceedings of World Wireless Congress*, vol. 2, p. 21, 2005.

[9] G. R. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, and B. Walke, "Ieee 802.11 s: the wlan mesh standard," *IEEE Wireless Communications*, vol. 17, no. 1, pp. 104–111, 2010.

[10] S. Bari, F. Anwar, and M. Masud, "Performance study of hybrid wireless mesh protocol (hwmp) for ieee 802.11 s wlan mesh networks," in *2012 international conference on computer and communication engineering (ICCCE)*, pp. 712–716, IEEE, 2012.

[11] D. Seither, A. König, and M. Hollick, "Routing performance of wireless mesh networks: A practical evaluation of batman advanced," in *2011 IEEE 36th Conference on Local Computer Networks*, pp. 897–904, IEEE, 2011.

[12] "Open-mesh. originator message version 2 (ogmv2)." `https://www.open-mesh.org/projects/batman-adv/wiki/OGMv2`. Accessed: 2023-02-16.

[13] M. Pease, R. Shostak, and L. Lamport, "Reaching agreement in the presence of faults," *Journal of the ACM (JACM)*, vol. 27, no. 2, pp. 228–234, 1980.

[14] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," in *Concurrency: the works of leslie lamport*, pp. 203–226, Springer, 2019.

[15] M. Castro, B. Liskov, *et al.*, "Practical byzantine fault tolerance," in *OsDI*, pp. 173–186, 1999.

[16] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[17] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, vol. 3, no. 37, pp. 2–1, 2014.

[18] A. Miller, Y. Xia, K. Croman, E. Shi, and D. Song, "The honey badger of bft protocols," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pp. 31–42, 2016.

[19] C. Decker, J. Seidel, and R. Wattenhofer, "Bitcoin meets strong consistency," in *Proceedings of the 17th International Conference on Distributed Computing and Networking*, pp. 1–10, 2016.

[20] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22328–22370, 2019.

[21] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A survey of distributed consensus protocols for blockchain networks," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 2, pp. 1432–1465, 2020.

[22] Y. Xiao, N. Zhang, J. Li, W. Lou, and Y. T. Hou, "Distributed consensus protocols and algorithms," *Blockchain for Distributed Systems Security*, vol. 25, p. 40, 2019.

[23] S. Bouraga, "A taxonomy of blockchain consensus protocols: A survey and classification framework," *Expert Systems with Applications*, vol. 168, p. 114384, 2021.

[24] B. Lashkari and P. Musilek, "A comprehensive review of blockchain consensus mechanisms," *IEEE Access*, vol. 9, pp. 43620–43652, 2021.

[25] C. Natoli, J. Yu, V. Gramoli, and P. Esteves-Verissimo, "Deconstructing blockchains: A comprehensive survey on consensus, membership and structure," *arXiv preprint arXiv:1908.08316*, 2019.

Appendix

I. Glossary

II. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Alice Cooper**,
*(*author's name)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Type Inference for Fourth Order Logic Formulae,
   *(*title of thesis)

   supervised by Axel Rose and May Flower.
   *(*supervisor's name)

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Alice Cooper
dd/mm/yyyy