Literature review of Formal Methods applied to Blockchain technology

Henoch Vitureira
Polytechnic Institute of Setubal
Setúbal, Portugal
2016041081@estudantes.ips.pt

ABSTRACT

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum enim nibh, aliquet sit amet ex suscipit, efficitur fringilla enim. Vestibulum elementum fringilla sodales. Cras malesuada sollicitudin consectetur. Cras elit nibh, tempus elementum dui vitae, sagittis porttitor elit. In sollicitudin purus neque, eu tristique felis varius in. Morbi sed commodo eros. Pellentesque ornare sed felis et bibendum.

CCS CONCEPTS

• **Software and its engineering** → **Process validation**; • **Security and privacy** → **Cryptography**.

KEYWORDS

Software Quality, Formal Methods, Blockchain, Smart contracts

1 INTRODUCTION

The rising complexity of features offered by software gives us the ability to use it for the most integrity and reliability dependent needs of our lives. Information systems that rely on these base principles can be financial and health related services, and in most recent times, blockchain based applications. Blockchain, being a decentralized and distributed network protocol that can be used as the core of a monetary system that performs peer-to-peer transactions, it needs the assurance of coordination and consensus of its economy's state [2]. Knowing that the protocol of a given blockchain project is what dictates how transactions and their coordination is maintained, the assurance of security and safety is a given. Formal methods can be introduced in order to provide unequivocal evidence that a given blockchain system and its consensus algorithm are secure and in according to expected software quality.

Formal methods are a rigorous description of a system or process using mathematics that aims to provide evidence of its reliability and robustness, according the specification in question [7].

//TODO FINISH

2 FORMAL MODELING APPLICATION

Blockchain systems can be divided into five main layers [2]: Data layer, Consensus layer, Smart contract layer and Application layer. As seen on Figure 1.

The data layer is responsible for block generation, the construction of the blockchain and storage [2]. The network layer connects the data and filters packets, and also manages the nodes. The rules of how the blockchain protocol must work are defined on the consensus layer. This includes agreement between nodes, fault tolerance and data consistency. Smart contracts have their execution lifecycle on the smart contact layer. The application layer is the upper level layer of used applications such as games or decentralized financial services.

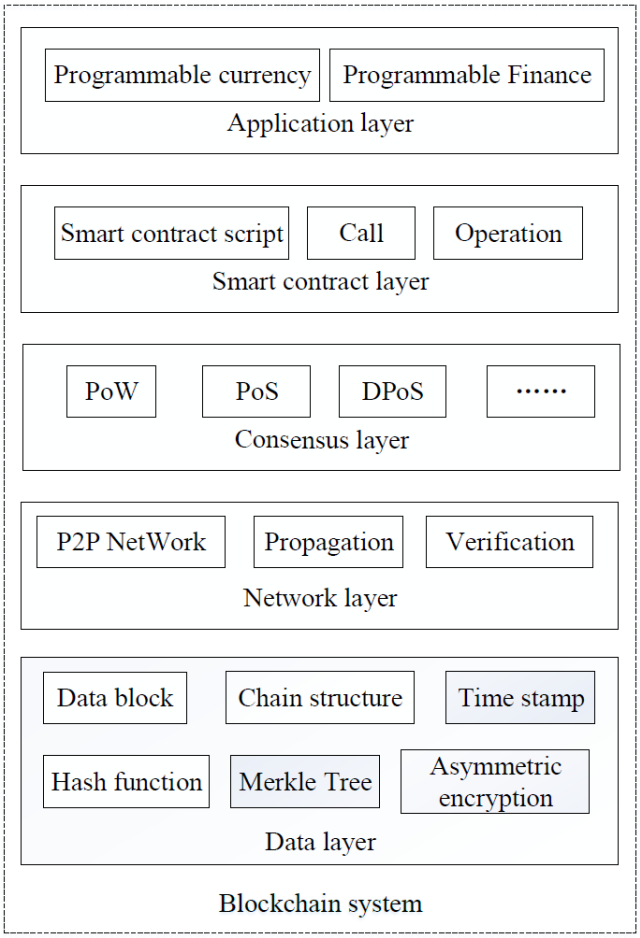

Figure 1: Blockchain infrastructure, from [2]

It is possible to use languages to define the standards of a blockchain system with mathematical rigor. One of them is SDL (Specification and Description Language), an Object-Oriented formal language defined by the International Telecommunications Unit. It is capable of mathematically describing the structure, behavior, and data of real-time and distributed communicating systems. It is also used by IEEE for standard definition [2].

With SDL, we can describe a structure at the levels of system, block or process. A system represents the object to describe. It is responsible for the communication with the outer environment

through the appropriate channels. A system can contain one or more blocks and blocks contain one or more processes. The majority of the functionality of an SDL system is described on the process level. Comparing to conventional programming languages, modules are represented by blocks in SDL, and funcionalities are represented by processes. In [2], the authors propose a hierarchical model focused on the consensus layer.

2.1 Data Structure

In [2] the modelled blockchain system used an Improved Byzantine Fault Tolerant Algorithm as the consensus algorithm. To define the data structure of a block in SDL, we need to represent them as a pair with a header BT and body BH, where:

$$B \equiv (B_H, B_T) \quad \text{with} \quad B_H \equiv (\mathbb{H}_p, \mathbb{H}_a, \mathbb{H}_t, \mathbb{H}_m, \mathbb{H}_h)$$
$$\text{and} \quad B_T \equiv (T_1, T_2, ..., T_{\mathbb{H}_l})$$

In these expressions, $\mathbb{H}_p$ represents the hash value of the previous block, $\mathbb{H}_p$ the hash value of the current block, $\mathbb{H}_t$ it's timestamp, $\mathbb{H}_m$ the merkle tree root (the hash of the transactions of a block), and $\mathbb{H}_h$ the current block's height.

The the block' structure definition can be seen on Listing 1.

```
newtype blockchain struct
  prehash integer;
  hash integer;
  length integer;/*the length of blocks*/
  merkleroot integer;
  ti Time;/*Timestamp*/
  translist list;
endnewtype;

newtype list
 array(maxit, integer)
endnewtype;

syntype maxit =Integer constants 0:25
endsyntype;
```

Listing 1: Definition of a block's structure

A blockchain can be regarded as a state machine [2] that contains a starting state, a non-empty state, the input transaction set, a state transition function, and an acceptance state set. Which can be formally defined as:

$$S \equiv (Q, \Sigma, \delta, s, F)$$

Where:

- Q is the non-empty state set, which represents all the states.
- Σ is the set of the newly generated and consensus blocks.
- δ represents the state transition function, $\delta : Q \times \Sigma \rightarrow Q$.
- s would be the starting state, which is the state of the system when it initializes. $\delta : s \in Q$.
- F representes the acceptance state set. $F \subseteq Q$.

The state of the model at the first block generation is the starting state. As transactions are made, the model a leader node to generate the block, which is transmitted to the other nodes on the blockchain network, triggering its validation for consensus. A consensus block stores the hash of the previous one and adds it to the end of the blockchain and completes the transfer of the blockchain state.

2.1.1 Diagrams. With a modelling language like SDL, we can use diagrams to define blockchain structural entities, like the core system of the consensus layer and blocks. Process flowcharts can model the flow of information and input validation, making a process like node validations of blocks and its subprocess of synchronization and voting can clearly illustrated.

3 CONSENSUS RULE

Blockchain system may choose to use consensus rules taken from popular protocols such as Bitcoin, but an alternative rule can be implemented, and also be formally modelled [3].

On a consensus rule, fault tolerances requirements are of utmost importance, including the collusion of nodes to alter a ledger. In [3], the author discusses a way to formally describe these tolerances with a scalable verification.

Endorsement policies, the rule that define the criteria of agreement os transaction results between nodes, can be described as a threshold function as follows:

$$T(m, \text{node}_1, ..., \text{node}_n), m \in \mathbb{N}$$

Where $T(2, \text{node}_1, \text{node}_2, \text{node}_3)$ would be an endorsement policy that means "among the three nodes in the argument, at least two nodes must return the same execution result". With this, we can mathematically model faulty state verification.

3.1 Faulty states

Let $\mathbb{B}$ and $\mathbb{N}$ be sets of Boolean and natural numbers, respectively. The state of the i-th node can be defined as: $s_i = (s_i.e, s_i.v)$, where for each transaction execution, $s_i.e \in \mathbb{B}$ is the existence of the result, and $s_i.v \in \mathbb{B}$ is the value of the result (correct or wrong).

Using the threshold function previously shown, $T(m, s_1, ..., s_n) = (T.e, T.v)$. With this, let

$$H_{=}\left(\sum_{i}^{n} I(s_i.e \wedge (s_i.v = b)) \geq m\right)$$

Where:

- m is a treshold
- $b \in \mathbb{B}$
- $I : \mathbb{B} \mapsto \{0, 1\}$ is the indicator function

With the expression above, the author of [3] infers:

$$T.e = H_{true} \vee H_{false}, T.v = \neg H_{false}$$

The endorsement policy $EP = (EP.e, EP.v)$ can be formed with T. When $EP.e$ = true, the policy is considered accordingly followed and the involved ledger is updated with the value of $EP.v$.

4 TEMPLATE OVERVIEW

As noted in the introduction, the "acmart" document class can be used to prepare many different kinds of documentation — a double-blind initial submission of a full-length technical paper, a two-page SIGGRAPH Emerging Technologies abstract, a "camera-ready" journal article, a SIGCHI Extended Abstract, and more — all by selecting the appropriate *template style* and *template parameters*.

This document will explain the major features of the document class. For further information, the *LaTeX User's Guide* is available from https://www.acm.org/publications/proceedings-template.

4.1 Template Styles

The primary parameter given to the "acmart" document class is the *template style* which corresponds to the kind of publication or SIG publishing the work. This parameter is enclosed in square brackets and is a part of the documentclass command:

```
\documentclass[STYLE]{acmart}
```

Journals use one of three template styles. All but three ACM journals use the acmsmall template style:

- acmsmall: The default journal template style.
- acmlarge: Used by JOCCH and TAP.
- acmtog: Used by TOG.

The majority of conference proceedings documentation will use the acmconf template style.

- acmconf: The default proceedings template style.
- sigchi: Used for SIGCHI conference articles.
- sigchi-a: Used for SIGCHI "Extended Abstract" articles.
- sigplan: Used for SIGPLAN conference articles.

4.2 Template Parameters

In addition to specifying the *template style* to be used in formatting your work, there are a number of *template parameters* which modify some part of the applied template style. A complete list of these parameters can be found in the *LaTeX User's Guide*.

Frequently-used parameters, or combinations of parameters, include:

- anonymous,review: Suitable for a "double-blind" conference submission. Anonymizes the work and includes line numbers. Use with the \acmSubmissionID command to print the submission's unique ID on each page of the work.
- authorversion: Produces a version of the work suitable for posting by the author.
- screen: Produces colored hyperlinks.

This document uses the following string as the first command in the source file:

```
\documentclass[sigconf]{acmart}
```

5 MODIFICATIONS

Modifying the template — including but not limited to: adjusting margins, typeface sizes, line spacing, paragraph and list definitions, and the use of the \vspace command to manually adjust the vertical spacing between elements of your work — is not allowed.

Your document will be returned to you for revision if modifications are discovered.

6 TYPEFACES

The "acmart" document class requires the use of the "Libertine" typeface family. Your TeX installation should include this set of packages. Please do not substitute other typefaces. The "lmodern" and "ltimes" packages should not be used, as they will override the built-in typeface families.

7 TITLE INFORMATION

The title of your work should use capital letters appropriately - https://capitalizemytitle.com/ has useful rules for capitalization. Use the title command to define the title of your work. If your work has a subtitle, define it with the subtitle command. Do not insert line breaks in your title.

If your title is lengthy, you must define a short version to be used in the page headers, to prevent overlapping text. The title command has a "short title" parameter:

```
\title[short title]{full title}
```

8 AUTHORS AND AFFILIATIONS

Each author must be defined separately for accurate metadata identification. Multiple authors may share one affiliation. Authors' names should not be abbreviated; use full first names wherever possible. Include authors' e-mail addresses whenever possible.

Grouping authors' names or e-mail addresses, or providing an "e-mail alias," as shown below, is not acceptable:

```
\author{Brooke Aster, David Mehldau}
\email{dave,judy,steve@university.edu}
\email{firstname.lastname@phillips.org}
```

The authornote and authornotemark commands allow a note to apply to multiple authors — for example, if the first two authors of an article contributed equally to the work.

If your author list is lengthy, you must define a shortened version of the list of authors to be used in the page headers, to prevent overlapping text. The following command should be placed just after the last \author{} definition:

```
\renewcommand{\shortauthors}{McCartney, et al.}
```

Omitting this command will force the use of a concatenated list of all of the authors' names, which may result in overlapping text in the page headers.

The article template's documentation, available at https://www.acm.org/publications/proceedings-template, has a complete explanation of these commands and tips for their effective use.

Note that authors' addresses are mandatory for journal articles.

9 RIGHTS INFORMATION

Authors of any work published by ACM will need to complete a rights form. Depending on the kind of work, and the rights management choice made by the author, this may be copyright transfer, permission, license, or an OA (open access) agreement.

Regardless of the rights management choice, the author will receive a copy of the completed rights form once it has been submitted. This form contains LaTeX commands that must be copied into the source document. When the document source is compiled, these commands and their parameters add formatted text to several areas of the final document:

- the "ACM Reference Format" text on the first page.
- the "rights management" text on the first page.
- the conference information in the page header(s).

Rights information is unique to the work; if you are preparing several works for an event, make sure to use the correct set of commands with each of the works.

The ACM Reference Format text is required for all articles over one page in length, and is optional for one-page articles (abstracts).

10 CCS CONCEPTS AND USER-DEFINED KEYWORDS

Two elements of the "acmart" document class provide powerful taxonomic tools for you to help readers find your work in an online search.

The ACM Computing Classification System — https://www.acm.org/publications/class-2012 — is a set of classifiers and concepts that describe the computing discipline. Authors can select entries from this classification system, via https://dl.acm.org/ccs/ccs.cfm, and generate the commands to be included in the LaTeX source.

User-defined keywords are a comma-separated list of words and phrases of the authors' choosing, providing a more flexible way of describing the research being presented.

CCS concepts and user-defined keywords are required for for all articles over two pages in length, and are optional for one- and two-page articles (or abstracts).

11 SECTIONING COMMANDS

Your work should use standard LaTeX sectioning commands: `section`, `subsection`, `subsubsection`, and `paragraph`. They should be numbered; do not remove the numbering from the commands.

Simulating a sectioning command by setting the first word or words of a paragraph in boldface or italicized text is **not allowed.**

12 TABLES

The "`acmart`" document class includes the "`booktabs`" package — https://ctan.org/pkg/booktabs — for preparing high-quality tables.

Table captions are placed *above* the table.

Because tables cannot be split across pages, the best placement for them is typically the top of the page nearest their initial cite. To ensure this proper "floating" placement of tables, use the environment **table** to enclose the table's contents and the table caption. The contents of the table itself must go in the **tabular** environment, to be aligned properly in rows and columns, with the desired horizontal and vertical rules. Again, detailed instructions on **tabular** material are found in the *LaTeX User's Guide*.

Immediately following this sentence is the point at which Table 1 is included in the input file; compare the placement of the table here with the table in the printed output of this document.

To set a wider table, which takes up the whole width of the page's live area, use the environment **table*** to enclose the table's contents and the table caption. As with a single-column table, this wide table will "float" to a location deemed more desirable. Immediately following this sentence is the point at which Table 2 is included in the input file; again, it is instructive to compare the placement of the table here with the table in the printed output of this document.

Table 1: Frequency of Special Characters

Non-English or Math	Frequency	Comments
Ø	1 in 1,000	For Swedish names
π	1 in 5	Common in math
\$	4 in 5	Used in business
Ψ_1^2	1 in 40,000	Unexplained usage

Always use midrule to separate table header rows from data rows, and use it only for this purpose. This enables assistive technologies to recognise table headers and support their users in navigating tables more easily.

13 MATH EQUATIONS

You may want to display math equations in three distinct styles: inline, numbered or non-numbered display. Each of the three are discussed in the next sections.

13.1 Inline (In-text) Equations

A formula that appears in the running text is called an inline or in-text formula. It is produced by the **math** environment, which can be invoked with the usual `\begin . . . \end` construction or with the short form `$ . . . $`. You can use any of the symbols and structures, from α to ω, available in LaTeX [**?**]; this section will simply show a few examples of in-text equations in context. Notice how this equation: $\lim_{n\to\infty} x = 0$, set here in in-line math style, looks slightly different when set in display style. (See next section).

13.2 Display Equations

A numbered display equation—one set off by vertical space from the text and centered horizontally—is produced by the **equation** environment. An unnumbered display equation is produced by the **displaymath** environment.

Again, in either environment, you can use any of the symbols and structures available in LaTeX; this section will just give a couple of examples of display equations in context. First, consider the equation, shown as an inline equation above:

$$\lim_{n\to\infty} x = 0 \tag{1}$$

Notice how it is formatted somewhat differently in the **displaymath** environment. Now, we'll enter an unnumbered equation:

$$\sum_{i=0}^{\infty} x + 1$$

and follow it with another numbered equation:

$$\sum_{i=0}^{\infty} x_i = \int_0^{\pi+2} f \tag{2}$$

just to demonstrate LaTeX's able handling of numbering.

Table 2: Some Typical Commands

Command	A Number	Comments
\author	100	Author
\table	300	For tables
\table*	400	For wider tables

14 FIGURES

The "figure" environment should be used for figures. One or more images can be placed within a figure. If your figure contains third-party material, you must clearly identify it as such, as shown in the example below.

Figure 2: 1907 Franklin Model D roadster. Photograph by Harris & Ewing, Inc. [Public domain], via Wikimedia Commons. (https://goo.gl/VLCRBB).

Your figures should contain a caption which describes the figure to the reader.

Figure captions are placed *below* the figure.

Every figure should also have a figure description unless it is purely decorative. These descriptions convey what's in the image to someone who cannot see it. They are also used by search engine crawlers for indexing images, and when images cannot be loaded.

A figure description must be unformatted plain text less than 2000 characters long (including spaces). **Figure descriptions should not repeat the figure caption – their purpose is to capture important information that is not already provided in the caption or the main text of the paper.** For figures that convey important and complex new information, a short text description may not be adequate. More complex alternative descriptions can be placed in an appendix and referenced in a short figure description. For example, provide a data table capturing the information in a bar chart, or a structured list representing a graph. For additional information regarding how best to write figure descriptions and why doing this is so important, please see https://www.acm.org/publications/taps/describing-figures/.

14.1 The "Teaser Figure"

A "teaser figure" is an image, or set of images in one figure, that are placed after all author and affiliation information, and before the body of the article, spanning the page. If you wish to have such a figure in your article, place the command immediately before the \maketitle command:

```
\begin{teaserfigure}
  \includegraphics[width=\textwidth]{sampleteaser}
  \caption{figure caption}
  \Description{figure description}
\end{teaserfigure}
```

15 CITATIONS AND BIBLIOGRAPHIES

The use of BibTeX for the preparation and formatting of one's references is strongly recommended. Authors' names should be complete — use full first names ("Donald E. Knuth") not initials ("D. E. Knuth") — and the salient identifying features of a reference should be included: title, year, volume, number, pages, article DOI, etc.

The bibliography is included in your source document with these two commands, placed just before the \end{document} command:

```
\bibliographystyle{ACM-Reference-Format}
\bibliography{bibfile}
```

where "bibfile" is the name, without the ".bib" suffix, of the BibTeX file.

Citations and references are numbered by default. A small number of ACM publications have citations and references formatted in the "author year" style; for these exceptions, please include this command in the **preamble** (before the command "\begin{document}") of your LaTeX source:

```
\citestyle{acmauthoryear}
```

Some examples. A paginated journal article [1], an enumerated journal article [6], a reference to an entire issue [5], a monograph (whole book) [2], a monograph/whole book in a series (see 2a in spec. document) [3], a divisible-book such as an anthology or compilation [4] followed by the same example, however we only output the series if the volume number is given.

REFERENCES

[1] Muhammad Fayez Afzaal, Aniqa Rehman, Saba Latif, and Nazir Ahmad Zafar. 2021. Blockchain-based Automated Formal Model for Smart Market System. *Proceedings of 18th International Bhurban Conference on Applied Sciences and Technologies, IBCAST 2021* (2021), 395–400. https://doi.org/10.1109/IBCAST51254.2021.9393231

[2] Zhangbo Duan, Hongliang Mao, Zhidong Chen, Xiaomin Bai, Kai Hu, and Jean Pierre Talpin. 2018. Formal modeling and verification of blockchain system. *ACM International Conference Proceeding Series* 86 (2018), 231–235. https://doi.org/10.1145/3177457.3177485

[3] Ryo Kawahara. 2020. Verification of customizable blockchain consensus rule using a formal method. *IEEE International Conference on Blockchain and Cryptocurrency, ICBC 2020* (2020), 14–16. https://doi.org/10.1109/ICBC48266.2020.9169472

[4] Zhentian Liu and Jing Liu. 2019. Formal verification of blockchain smart contract based on colored petri net models. *Proceedings - International Computer Software and Applications Conference* 2 (2019), 555–560. https://doi.org/10.1109/COMPSAC.2019.10265

[5] Shin'Ichiro Matsuo. 2017. How formal analysis and verification add security to blockchain-based systems. *Proceedings of the 17th Conference on Formal Methods in Computer-Aided Design, FMCAD 2017* (2017), 1–4. https://doi.org/10.23919/FMCAD.2017.8102228

[6] Yvonne Murray and David A. Anisi. 2019. Survey of formal verification methods for smart contracts on blockchain. *2019 10th IFIP International Conference on New Technologies, Mobility and Security, NTMS 2019 - Proceedings and Workshop* 2 (2019), 1–6. https://doi.org/10.1109/NTMS.2019.8763832

[7] Gerard O'Regan. 2014. *Software Quality Assurance Software Quality Assurance*. 143–150. https://doi.org/10.1007/978-3-319-06106-1_9

A RESEARCH METHODS

A.1 Part One

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi malesuada, quam in pulvinar varius, metus nunc fermentum urna, id sollicitudin purus odio sit amet enim. Aliquam ullamcorper eu ipsum vel mollis. Curabitur quis dictum nisl. Phasellus vel semper risus, et lacinia dolor. Integer ultricies commodo sem nec semper.

A.2 Part Two

Etiam commodo feugiat nisl pulvinar pellentesque. Etiam auctor sodales ligula, non varius nibh pulvinar semper. Suspendisse nec lectus non ipsum convallis congue hendrerit vitae sapien. Donec at laoreet eros. Vivamus non purus placerat, scelerisque diam eu, cursus ante. Etiam aliquam tortor auctor efficitur mattis.

B ONLINE RESOURCES

Nam id fermentum dui. Suspendisse sagittis tortor a nulla mollis, in pulvinar ex pretium. Sed interdum orci quis metus euismod, et sagittis enim maximus. Vestibulum gravida massa ut felis suscipit congue. Quisque mattis elit a risus ultrices commodo venenatis eget dui. Etiam sagittis eleifend elementum.

Nam interdum magna at lectus dignissim, ac dignissim lorem rhoncus. Maecenas eu arcu ac neque placerat aliquam. Nunc pulvinar massa et mattis lacinia.