# Claims of the chain – A formal analysis of the blockchain protocol in B

Richard Dennis, Gareth Owenson and Benjamin Aziz

School of Computing, University of Portsmouth
{richard.dennis,gareth.owenson,benjamin.aziz}@port.ac.uk

**Abstract.** This paper presents the first ever formal analysis of the blockchain protocol using a B language tool, ProB. We first discuss formal methods and conduct a critical analysis of their benefits before moving on to construct a comprehensive formal model of the blockchain, focusing in particular on its current security vulnerabilities and the scalability problem that still affects the growth of all blockchain-based networks. We then present our formal model of the blockchain to test whether the long-proclaimed, but to date formally untested, security properties of the blockchain are correct. We then propose a new and innovative approach to solving the scalability issue of the blockchain in the form of a "rolling blockchain". We then create this new blockchain and analyse whether, under the same formal specification, this newly proposed method is as secure as the traditional blockchain. We then consider the limitations of the newly proposed blockchain method, before using simulations and analysis to improve the security of not only the rolling blockchain, but all blockchain-based networks. We conclude by suggesting areas for future research and summarising our findings.

**Keywords:** B language, Blockchain, Bitcoin, Scalability, Cryptocurrencies, Formal Methods, Formal Proof, Cryptography

## 1 Introduction

Bitcoin is the decentralized, peer-to-peer electronic currency system that was first described by a developer using the pseudonym of Satoshi Nakamoto in 2008 [23]. Bitcoin has been very successful; it now has a market cap of over USD 8.5 billion and sees an average of 214,000 transactions being conducted on its network every day [5].

The underlying technology to the Bitcoin, and indeed all cryptocurrencies, is the blockchain. The blockchain is an immutable public ledger of transactions, reaching a consensus through a mechanism called proof-of-work. It is also the first method that solves the "Strong Byzantine Generals" (SBG) problem [21].

It is surprising, given the speed with which Bitcoin has grown, that no existing research sets out to challenge the security principles of the blockchain – Confidentiality, Integrity and Authenticity – by conducting a formal analysis of the blockchain.

Despite extensive development over many years and significant benefits having been demonstrated, formal methods remain poorly accepted both by industrial practitioners and in academic research [15]. The aim of formal methods is to discover ambiguity, incompleteness, and inconsistency in protocols or software. They have been used to unearth real-world security issues; with one such example being the use of the B language to discover a flaw in a major safety-critical system application concerning Line 14 of Paris Métro [19].

Formal methods allow the protocol to be expressed using unified notation, based on set theory and mathematical logic. This removes any ambiguity from the specification, and allows the formal specification to be refined to deployable code. Once a machine has been proven to be consistent and correct, these proofs should be valid in any context in which this machine is used as part of a more complex specification [3].

This paper conducts the first formal analysis using the B language of a traditional blockchain as first created and used by Bitcoin to prove whether the assumed security principles hold true, and also explores the rolling blockchain model, conducting a formal analysis on this new proposed model to see if it can achieve the same security principles as the traditional blockchain.

The structure of this paper is as follows; first we examine related work conducted in the field of the blockchain and also formal methods, before conducting a formal analysis of the traditional blockchain using the B language. We then propose a solution to blockchain's scalability problem – the rolling blockchain – and present the results of our formal analysis of this new model, which we thoroughly compare to those results obtained for the traditional blockchain model. We then propose future work for this area of research, before summarising our findings and concluding the paper.

## 2 Related work

### 2.1 Formal Methods

Formal methods aim to provide a method to prove that a specification is realizable, complete, consistent, unambiguous and verifiable. Even the most complex systems can be modelled using relatively simple mathematical objects, such as sets, relations and functions, which form the basis of all formal languages. Kossak and Mashkoor [15] expand on this definition by stating that all formal languages are based on set theory and First Order Predicate Calculus.

Verifying the system allows a high degree of confidence to be placed in it, however this statement is highly debated by Hall, who argues that this statement is the biggest "myth" in formal specifications, and that although all formal specifications involve a high degree of mathematical proofs, a formal specification can never be called "perfectly correct" however much you prove about the models [12].

Knight et al. [15], Voros et al. [31], and Bicarregui et al. [3] all demonstrate real world examples where the implementation of formal methods resulted in significant bugs being found in the specification, such as on the Paris Métro Line 14 and at the Darlington Nuclear Facility.

Today, there are a number of tools available to aid in the development of formal specifications. These tools are based on the three popular languages for formal methods: B language, Vienna Development Method (VDM) and the Z language [14]. These three languages are all model-based, with the specification being expressed as a system state model. The languages are "formal" in the sense that they have formal semantics and as a result can be used to express specifications in a clear and unambiguous manner. Pandey and Srivastava identify that formal languages such as Z, B, and VDM are only able to demonstrate sequential systems [24].

Z was the first formal language to be developed in academia, having been created in 1977 by J.R Abrial [1] and later being further researched and developed by Oxford University. Lano [18] summarises Z's focus as being the formalisation of requirements rather than the correct executable implementation of the specification. Kaur et al. further elaborate on this summary by explaining how Z is a high level abstract model of the system requirements, and only provides a base to design and test the system [14], while Diller and Docherty explain that there is no method to develop the abstract model to machine code [8].

The Z language formed the basis of the B language, which was developed to solve many of the fundamental issues and limitations of the Z language. Given B's foundations, it is perhaps unsurprising that B notations at an abstract level are almost identical to Z's [17]. Lano reports that, at present, the B language is the most popular formal method to be used in industry projects [18].

As highlighted by Diller and Docherty [8], Smith describes how the B language is the first formal language to allow refinement – an incremental development process to develop the model – from an abstract specification to machine code (C++) [28].

Leuschel and Butler further expand on the ability of the B language by describing two activities which no previous formal language has managed: consistency checking and refinement checking [20]. Consistency checking ensures the operations conducted by the machine do not invalidate the invariant, and the refinement checker ensures each machine is a valid refinement of a previous machine.

Whilst Bicarregui et al. argue that B provides less abstract specification than VDM [3], Kaur et al. dispute this by describing how B allows a more in-depth level of analysis to be conducted, and the B language focusses refinement to code in much greater detail than VDM [14]. Bowen and Hinchey [4] elaborate on this, stating that the B language is representative of the next generation of formal methods, and further criticise the Z language by pointing out that, although both B and VDM can be used to generate source code (C language) directly from the formal specification models created, the Z language does not provide such functionality.

## 2.2 Blockchain

The blockchain was first described in a self-published research paper entitled "Bitcoin: A Peer-to-Peer Electronic Cash System" written under the pseudonym of Satoshi Nakamoto. The blockchain is the underlying gossip protocol of all cryptocurrencies and is a novel peer-to-peer method of linking a sequence of transactions or events together in a way that makes them immutable [23]. McConaghy et al. accurately describe the main characteristics of the blockchain as decentralized control, immutability, and creation & movement of digital assets [21], and Pilkington credits the success of Bitcoin solely to the blockchain [25].

Drainville correctly describes how the blockchain is a collection of every transaction to have ever occurred on the Bitcoin network [9]. On creating a transaction, a user broadcasts this to all peers in the network. Kroll et al. expand on this by explaining how a select group of peers, called miners, collect broadcast transactions and attempt to gather them in a block that satisfies a cryptographic hash function [17]. The block must contain a cryptographic hash of the previous block; this is the method used to cryptographically link every block in the blockchain to its previous block, all the way back to the first or "genesis" block. Producing a block is both computationally intensive and probabilistic. Given a proposed block, each miner has a fixed and independent probability of successfully producing a block which satisfies the hash function for each unit of computation time. Whilst it is difficult to produce a block, it is not difficult to verify a correct block.

Kroll et al. [17] explain that the mining process requires vast computing power as only a "brute force, trial and error" method can be used to calculate the SHA-256 hash. Every two weeks, the complexity of the challenge is adjusted to ensure that, on average, a block is mined every 10 minutes. The financial incentive of 25 bitcoins (USD 14,419.50 [7]) is offered to the first miner to successfully calculate the hash. Barber et al. [2] argue that it is this financial reward that ensures the majority of the miners on the network act honestly and obey the network protocol.

Sompolinsky and Zohar argue that only an attacker controlling more than 51% of the network hashing power would have the ability to change past transactions [29], and demonstrate that the cost of resources required to control 51% would outweigh the potential rewards. Dumas et al. [10] question the 51% vulnerability claim, which was originally presented in Nakamoto's whitepaper [23], suggesting that it is a widespread security claim, but no analysis has been conducted to prove or disprove this assumption.

Vulnerability to attacks is not the blockchain's only issue. Poon and Dryja summarise the scalability problem facing all blockchain-based networks as not being a single problem, but rather the combination of multiple issues that ultimately affect the possible scalability of the blockchain [26]. Poon and Dryja [26] reinforce their scalability argument by demonstrating how the maximum theoretical number of transactions per second that Bitcoin's blockchain is able to process is 7, whereas VISA can process 20,000. McConaghy et al. [21] agree with Poon and Dryja and demonstrate that the Bitcoin's blockchain is currently 50GB – having grown by 24GB in 2015 – and also prove that in order to achieve the transaction rate of VISA by only increasing the block size, the blockchain would need to grow by 3.9 GB/day or 1.42 TB/year.

Overall, the blockchain is the most important invention of the original Bitcoin whitepaper. While it has seen impressive growth and now handles an average of 239,138 transactions per day [6], it is not a faultless system. Having been shown to be vulnerable to attacks, such as the 51% attack, and faced with scalability issues which impact on the potential growth, there is plenty of room for further research to solve these issues.

## 3 Our formal analysis method

No formal analysis has currently been conducted on the blockchain, there is still an unanswered research question: What does Bitcoin's traditional blockchain offer in terms of the three fundamental security properties of Information Systems Security?

We conduct the first analysis to see if the Bitcoin blockchain adheres to the three guiding principles in information security of Confidentiality, Integrity and Authenticity [30].

There are no formal requirements for the blockchain, however there are assumed goals and, in the context of our research, we defined the requirements for each of the core security principles as follows:

- Confidentiality – Whether bitcoins can be created, copied or stolen using a defined set of attacks. For this, we will look into the double spend attack as well as ensuring the underlying protocol is mathematically secure. We will not be looking at attacks where the user's wallet is stolen.
- Integrity – Data that is stored in the blockchain cannot be altered or modified. This is the key focus of our work; we aim to show whether the blockchain successfully defends against modification of data using a defined set of attacks
- Availability – Ensure authorised users are not denied service and, as long as a single node is still available and not under attack, the network is still able to function correctly.

While we will look at all three principles, the primary focus of our research will be Integrity and, as there is no formal specification, we have created our own specification of what the blockchain should do:

- Data can only be inserted into the blockchain if valid
- No data can be duplicated
- Once valid, no data can be removed
- A block can only be inserted if the data it contains is valid, not a duplicate of a previous block, and has successfully completed the required proof of work
- The blockchain should be able to be traced back to the first or "genesis" block

The models created for our experiments were all created using the B language and the ProB syntax. This language was chosen due to its ability to refine the model to a greater depth than alternative languages, and the fact that it allows accurate modelling of the complex data structure of the blockchain.

An invariant is a condition on the state variables that must hold true permanently when the operations are run correctly and which adheres to the machine properties.

We have defined the invariant I as:

I == okay => P /\ not okay => not P

Property "I" should always hold true, the invariant property is defined as "P", and "okay" is a Boolean history variable, which does not influence the behaviour but is true as long as no malicious actions were carried out and false once a malicious operation has been performed. We consider the model to be correct if the invariant holds true after each operation is run.

There are two main proof activities when using the B language, both of which we use during our experiments. The first is consistency checking, which shows all operations that are run preserve the invariant. The second is refinement checking, which is used to show a refinement machine model is a correct and valid refinement of a previous machine model. In addition, ProB also contains a temporal and a state-based model checker, both of which can be used to detect various errors in B specifications.

The model gets checked using an exhaustive model checking model, which restricts the sets to a small finite set and the integer variables to a small range, which allows the model checking tool to traverse all the reachable states of the machine to find any problems such as a violation of the invariant.

In addition to this, the validation of a machine is ensured in ProB by conducting more than 1000 unit tests, monitoring pre- and post-conditions during run time, integration testing, as well as validating the parser.

ProB validation tools are valid for use in the Safety integrity level 4 development process. This is the most dependable of all the European functional safety standards, and ProB animation facilities give users the confidence that their specifications are correct and valid.

### 3.1 Our experiments

We define a complete formal model based on the specification we set out for the blockchain.

This model is the basis for all our experiments and it accurately models the data structure of the blockchain: it focuses on how blocks are added to the blockchain, with

```
SETS
  USER; BLOCKS; BLOCK_HASH; PREVIOUS_HASH; RESPONSE =
  {Yes, No}

VARIABLES
  accounts, transactions, cryptographic_link, confir-
  mation, blockid, nextid

INVARIANT
  accounts <: USER &
  confirmation : BLOCKS >+> BLOCK_HASH &
  cryptographic_link : BLOCK_HASH >+> PREVIOUS_HASH &
  transactions : accounts >+> BLOCKS &
  card(BLOCK_HASH) = card(PREVIOUS_HASH) &
  nextid :NATURAL1 &
  blockid : BLOCKS >+> NATURAL1 &
  card(confirmation) = card (blockid)

INITIALISATION
  accounts, transactions, cryptographic_link, confir-
  mation, blockid, nextid := {},{}, {},{},{},1

OPERATIONS
add_block(b, bh,ph) =
PRE
  b : BLOCKS &
  bh : BLOCK_HASH &
  b |-> bh /: confirmation &
  ph : PREVIOUS_HASH &
  bh |-> ph /: cryptographic_link
THEN
  confirmation := confirmation \/ {b |-> bh}||
  cryptographic_link(bh) := ph ||
  blockid(b) := nextid;
  nextid := succ(nextid)
END;
```

*Figure 1: Formal model of the blockchain*

a particular focus on the invariant to ensure that this mimics the current blockchain as accurately as possible.

The invariant defined in the model above sets out the rules which the model must follow to be considered correct; these were created based on the deployed blockchain system found in Bitcoin and the formal requirements described in section 3.

We ensure each block must have a single block hash, and ensure that no two blocks can have the same block hash. This was achieved using partial injections in the invariant. This is an accurate model of the real world hash function as a critical property of hash functions is that two different inputs must have different hashes.

Partial injections functions were also used in the invariant to specify that each block can only have a single ID, which is a positive natural number.

In this model, we achieve the cryptographic link, which in the deployed network links the blocks together, by linking the current block hash with that of the previous block using a partial injection function, ensuring only one link between blocks can exist.

The add_block function in the model achieves the specification requirement for ensuring only valid data (which we model as a block) can be inserted only if b (the block to be inserted) is an element of the set block, where we assume the set block contains only valid possible blocks. The same method has been used to ensure a correct and valid block hash has been calculated. To achieve the requirement of ensuring a replay attack is not possible, the prerequisites check that the block attempting to be added has not previously been included. To do this it ensures there is not an existing relationship between the block data and the block hash.

The prerequisites for the add_block operation mimic the real world system where each miner would check that the block contains valid data and the correct block hash as well as ensuring the block has not previously been included in the blockchain before attempting to include this block in the blockchain.

The add_block functionality also gives each block a unique ID which is chained together so the requirement of the blockchain being able to be traced back to the first or “genesis” block is also achieved.

The adversary we model as attacking the network is an adversary with less than a majority of computing power on the network, and one who follows the protocol behaviour correctly.

The experiments conducted were rigorously tested, all validation was conducted using the ProB validation tool, testing 1000 possible use cases to ensure the Safety integrity level 4 properties for validation of the machine were achieved. The operations were replicated 1000 times to ensure the accuracy and reliability of the results. Furthermore, all experiments were conducted in the same environment with the same amount of resources.

The formal model of the blockchain can be seen in Figure 1. This model’s invariant held true for all of the use cases used to test its validity. At no point did any of the operations invoke an invalid state of the invariant, proving that our model of the blockchain, after completion of the operations, never put the invariant into an invalid state, thus proving that our model of the blockchain is mathematically correct. The experiments conducted below attempt to show how an attacker would attempt to subvert the protocol to gain an advantage on the network, such as the ability to remove blocks from the blockchain or add blocks of data which have previously been included in the blockchain.

```
block_duplication(b,bh,ph) =
PRE
  b : BLOCKS &
  bh : BLOCK_HASH &
  b |-> bh : confirmation &
  ph : PREVIOUS_HASH &
  bh |-> ph : cryptographic_link
THEN
  confirmation := confirmation \/ {b |-> bh}||
  cryptographic_link(bh) := ph
END;
```

*Figure 2: Operation attempting to duplicate a block in the blockchain*

The first operation we created set out to prove the requirement that no data can be duplicated within the blockchain. As can be seen in Figure 2, an operation was added to the base proof model which attempted to duplicate a block already stored within the blockchain.

During testing, when the operation was run, it caused the invariant to become invalid. The partial injection functions linking the block hash to the previous block hash, and the linking of the block to the block hash became invalidated. In addition, the number of confirmations failed to match the number of block IDs. To check this result was not an anomaly, the ProB validator tool ran this operation using 1000 case studies and each time the operation was run, the same invalidation of the invariant was produced. This operation caused the invariant to not hold true, clearly showing that the data (the block) cannot be duplicated and thus satisfying the requirement of no duplicated data being able to enter into the blockchain.
This confirmed an important feature of the blockchain, as this will prevent an attacker trying to claim that a transaction occurred multiple times by replaying the transaction.

Next, we challenged the blockchain’s immutable and integrity properties to evaluate whether the specification that once a piece of data has been added to the blockchain it can never be altered holds true.

```
data_deletion(b) =
PRE
  b : BLOCKS
THEN
  confirmation := {b} <<| confirmation
END;
```

*Figure 3: Operation attempting to remove a block from the blockchain*

Figure 3 shows an operation which attempted to test the immutability property of blockchain; we focused the experiment to see if data stored within the chain can be deleted. We first attempted a naïve method of deleting a block, by simply removing the block from the chain; this invalidated the invariant of the number of elements in the confirmation set and the number of block IDs was then not equal. In addition, the previous hash and current hash no longer collated. Once again, the results were verified using the ProB validation tool kit and this operation was run in 1000 use cases. In each case the invariant was invalided.

However, this operation can be improved by deleting the block entirely from the blockchain and recreating the cryptographic link between the previous block and the following block. This newly modified operation was created and tested, and ensured that the invariant remained valid. While this demonstrates that it is possible to delete a block of data from the blockchain, this model did not take into account the amount of resources this attack would require.

For this operation to succeed in the live deployed network, an attacker would need to re-mine every block from the deleted block onwards to ensure that the hashes linking to the previous block were correct and valid. Since this would require a majority of network hashing power and would be exponentially harder the further back the block was deleted from the blockchain, we consider this operation to be outside the attack model of this paper, and thus can conclude that, without a majority of hashing power, an attacker cannot remove a block from the blockchain.

### 3.2 Summary of the experiments conducted on our formal model of the blockchain

Having created the first formal specifications for the blockchain and, using these requirements as a base, created the first formal model of the blockchain using the B language, we then used this model as a base to conduct a series of experiments to see if the blockchain behaves in the manner which has been assumed by the blockchain community.

In addition, the experiments set out to see if the blockchain conformed to several key security properties – Confidentiality, Integrity and Authenticity – with a focus on data integrity within the blockchain. We are the first to use formal methods to determine whether data integrity is always maintained in the blockchain.

The formal specifications were met, and the assumed security properties of the blockchain held true. Perhaps this is not surprising given that these properties have been shown to be correct in other experiments, but this is the first time that these properties have been proven correct using formal models.

The operations were thoroughly tested using a large number of use cases and using the built-in ProB validation tools to meet the safety integrity level 4 standard. By working to this standard, it gives us confidence that the results obtained are accurate and are a reliable model of how the live, deployed system would behave if attacked. In addition, the experiments conducted show that the security properties of Integrity and Authenticity are always adhered to; during the experiments, these properties were never compromised. This is a positive result which shows that the blockchain protocol is a very good protocol for storing data and ensuring the integrity of the data can never be compromised.

Overall, the experiments did not result in any surprises, and reinforced the previously assumed properties of the blockchain. They have provided a strong basis for future development and research into the blockchain and will be further developed as research into this topic increases.

## 4 Rolling blockchain overview

With the increase in adaption of blockchain-based networks such as Bitcoin, the fundamental limitations of all blockchain-based networks are now being realised. Currently, a major limitation of the adaption of blockchain-based networks is the amount of resources – specifically the hard drive space required to store the blockchain – that a user must donate to the network in order to participate in it.

Bitcoin’s blockchain is currently 71.8GB in size and is increasing at a rate of 1GB every 7 days. The peer-to-peer nature of the blockchain means that each client on the network is required to download, store and keep the complete blockchain up-to-date. This prevents low-resourced and mobile users from participating in the network and is a problem that will get bigger the longer the Bitcoin network is deployed. In turn, this will lead to a system that is increasingly centralised, defeating the aim of all blockchain-based systems as set out by Nakamoto. The lack of participation from low-resourced users also negatively impacts the security of the blockchain as, if these users were able to participate in the network, they would donate hashing power to the network and would therefore make it more expensive for an attacker to successfully conduct the 51% attack.

With some applications of blockchain technology, such as reputation data being stored on a blockchain, the requirement to store all data since the creation of the network is not needed.

Currently there is only one method to reduce the size of the blockchain; pruning. This involves each node downloading the entire blockchain, and manually searching through the blockchain to remove any “spent transactions”. However, this method clearly has many disadvantages; such as needing to first download the entire blockchain, and then having to use computational resources to manually remove spent transactions. There is no global consensus on what is the smallest required blockchain and, while the blockchain can currently be reduced by 35%, this is still far from ideal. This method also focuses on transaction-based blockchain systems, and does not take into account storage-based blockchains, where after a set period of time the stored data becomes obsolete.

We propose an innovative new method of solving the scalability issue, a rolling blockchain. In this blockchain, only data stored for a pre-set period will be included in the blockchain; any data older than this period is removed automatically.

In the example of a reputation system, to prevent a user and their score from being deleted in the event that they have not gained any reputation in the past thirty days, upon each deletion, the network would be able to populate a special "history" section of the block, which would average the user's reputation score from the data and add it to this section, thus ensuring that no user is ever forgotten.

This rolling method would not be a separate action or require any additional resources; instead, it would be merged with the mining process for a new block at a set point daily, for example at midnight GMT.This method allows for a consistent size blockchain that would be significantly smaller than the current Bitcoin blockchain, for example, enforcing a period of 30 days would reduce Bitcoin's blockchain from the current size of 77GB to just 4.36GB.

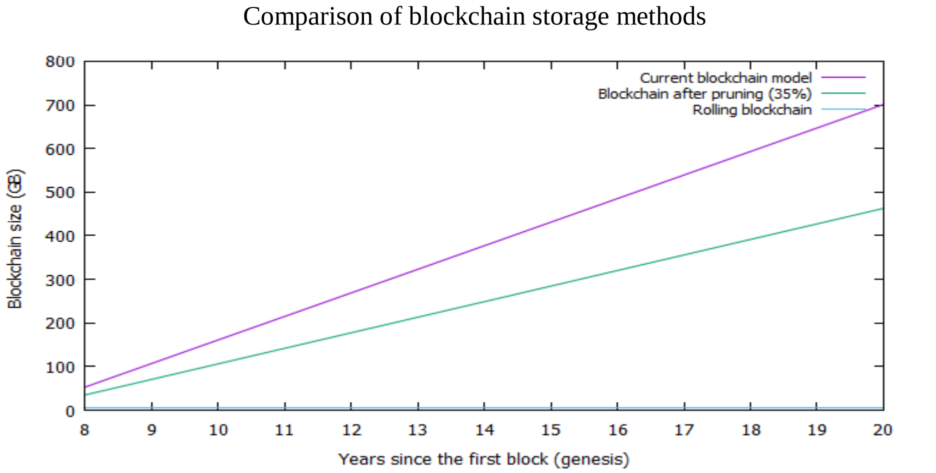


*Figure 4: Graph comparing blockchain size compared to network deployment time*

Figure 4 shows a comparison of the current reduction in storage methods applicable for blockchain. It shows the current blockchain, a blockchain after pruning, and our proposed model, all modelled up to twenty years from the creation of the network, based on the assumption that the current network growth is maintained.

Figure 4 clearly demonstrates how a network with the rolling blockchain implemented would be able to include lower-resourced users, whereas even with pruning enabled, the resources required for traditional blockchain-based networks would exclude all bar the most highly-resourced users. This will allow a rolling blockchain-based network to scale to a greater scale than the traditional blockchain network.

A rolling blockchain not only removes the requirement for any node to maintain a full blockchain dating back to the creation of the network, but as a new user to the network is no longer required to download the entire blockchain from the first or "genesis" block, resources are reduced not only in terms of storage, but also in terms of the bandwidth required to download the blockchain, thus reducing the load on the network.

### 4.1 Modelling of the "Rolling Blockchain" in B

This section will focus on the modelling of the rolling blockchain in the B language; it will continue from the modelling of the traditional blockchain, using the same methodology and procedures. The refinement of the blockchain model to evaluate the rolling blockchain, while the preferred method, was not possible since two specification requirements had been removed due to the nature of the rolling blockchain, these were:

- Once valid, no data can be removed
- The blockchain should be able to be traced back to the first or "genesis" block

These requirements, while valid for the traditional blockchain model, can no longer be specified for the rolling blockchain. With the deletion of data now a requirement, the model would not be able to satisfy these requirements,
The requirements for the rolling blockchain can be seen below and once again focus on the integrity of the data contained within the rolling blockchain.

- Data can only be inserted into the blockchain if valid
- No data can be duplicated
- A block can only be inserted if the data contained within it is valid, not a duplicate of a previous block, and successfully completed the required proof of work
- After a pre-defined time, data will be removed from the blockchain

This section investigates the impact that the deletion of data from a blockchain has on the core security principles of Confidentiality, Integrity, and Authenticity, with a focus on Integrity. The results obtained in this section will be compared to the results obtained in the traditional blockchain section to see whether the deletion of data creates additional security vulnerabilities.

We will focus on the above specification during our experiments, as these are the critical and most important specifications which the rolling blockchain must adhere to in order for it to be considered a viable alternative to the traditional blockchain as implemented in the Bitcoin protocol.

### 4.2 Our experiments

We created a modified version of the traditional blockchain's formal model, to enable the new blockchain to conduct the self-deletion of data. We opted for the roll to be conducted after 30 days, although this would be dependent on the application of the rolling blockchain and the requirements of the network. The experiments were conducted under the same testing environment as the traditional blockchain model in order to ensure results can be compared fairly. This model, as can be seen in Figure 5, is the basis for all our experiments in this section and models the rolling blockchain as accurately as possible.

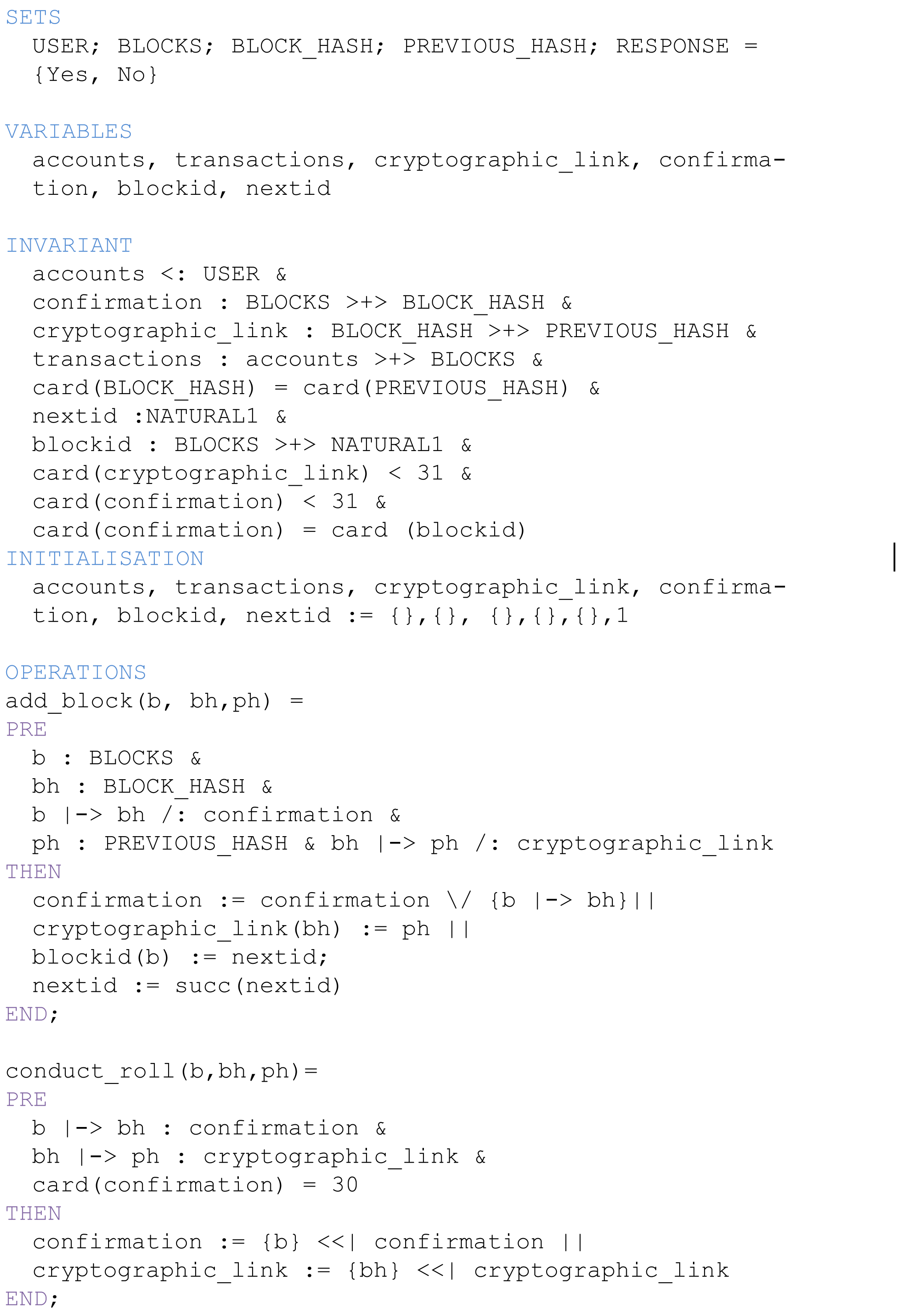
The model in Figure 5 modelled the storing and addition of data to the blockchain, as well as the deletion of data after 30 days as mentioned above. With the exception of this addition, and the inclusion of the conduct_roll operation, the invariant in this model is the same as that of the blockchain modelled in the previous section.

This operation ensures the block to be deleted is the correct block, preventing any pre-emptive deletion of a block as can been seen in the prerequisites for the operations. The prerequisites ensure the block to be deleted is the correct block in the chain, and the block has previously been confirmed in the blockchain.

The operation then removes the block and cryptographic link from the blockchain, thus ensuring the invariant never becomes invalidated, unlike in the original model when a block was deleted.

The model was once again evaluated using the same methods as the previous model to ensure consistent results which can be fairly compared with each other. The ProB validation suite was used to ensure the state model never became invalidated and, if it were to be invalidated, it would display what operation caused the invalidation. Using 1000 use cases the model in Figure 5 always maintained a valid state.

This result shows that the basic model of the rolling blockchain is correct and correctly implements all mathematical standards.



```
SETS
  USER; BLOCKS; BLOCK_HASH; PREVIOUS_HASH; RESPONSE =
  {Yes, No}

VARIABLES
  accounts, transactions, cryptographic_link, confirma-
  tion, blockid, nextid

INVARIANT
  accounts <: USER &
  confirmation : BLOCKS >+> BLOCK_HASH &
  cryptographic_link : BLOCK_HASH >+> PREVIOUS_HASH &
  transactions : accounts >+> BLOCKS &
  card(BLOCK_HASH) = card(PREVIOUS_HASH) &
  nextid :NATURAL1 &
  blockid : BLOCKS >+> NATURAL1 &
  card(cryptographic_link) < 31 &
  card(confirmation) < 31 &
  card(confirmation) = card (blockid)
INITIALISATION
  accounts, transactions, cryptographic_link, confirma-
  tion, blockid, nextid := {},{}, {},{},{},1

OPERATIONS
add_block(b, bh,ph) =
PRE
  b : BLOCKS &
  bh : BLOCK_HASH &
  b |-> bh /: confirmation &
  ph : PREVIOUS_HASH & bh |-> ph /: cryptographic_link
THEN
  confirmation := confirmation \/ {b |-> bh}||
  cryptographic_link(bh) := ph ||
  blockid(b) := nextid;
  nextid := succ(nextid)
END;

conduct_roll(b,bh,ph)=
PRE
  b |-> bh : confirmation &
  bh |-> ph : cryptographic_link &
  card(confirmation) = 30
THEN
  confirmation := {b} <<| confirmation ||
  cryptographic_link := {bh} <<| cryptographic_link
END;
```

*Figure 5: Formal model of the rolling blockchain*

It also shows how it is possible for a rolling blockchain to be implemented, and disproves many criticisms that there is no solution to the scalability issue and that it is impossible to delete data contained within a blockchain. However, this result does not show whether the rolling blockchain is able to maintain integrity of data contained within the blockchain when an adversary is attacking the blockchain. The adversary model for the rolling blockchain is the same model as presented for the traditional blockchain model.

```
data_deletion(b) =
PRE
  b : BLOCKS
THEN
  confirmation := {b} <<| confirmation
END;
```

*Figure 6: Operation attempting pre-emptive data deletion from the blockchain*

The operations demonstrated in the rest of this paper attempt to subvert the main protocol of the rolling blockchain to invalidate the formal specifications.

Figure 6 shows a pre-emptive deletion of a data block before the applicable time to delete blocks. The operation attempts to delete a block at a random point in the blockchain. This would mimic an attacker attempting to remove a block of data before the correct period.

This operation was a naïve attempt to remove the block from the blockchain, where an attacker simply tried to remove the block without considering the cryptographic links between the previous and next blocks. Due to the method of block deletion, during the testing of this operation, it was shown that this operation invalidated the invariant. This result was confirmed during the use case validation testing, where this operation caused the invariant to fail 100% of the time.

However, an improved method of attack would be remove the cryptographic link between the previous and following block. An operation to test this theory was implemented and evaluated and, perhaps surprisingly, this method kept the state machine valid at all times, and passed all 1000 use cases. On the surface, this shows that the blockchain is susceptible to this attack, however for this attack to be successfully conducted on a live deployed network, the attacker would require over half of the total hashing power of the network, which is beyond the attacker model used in this paper. It should also be noted that this is the same attack that was able to be conducted on the traditional blockchain, as discussed in Section 3.1. This result demonstrates that linking together with the hash of the previous block, as per Bitcoin's blockchain, is effective against this attack and, since the attack was before the correct deletion point, the attack failed. This shows that integrity of data against deletion is achieved in this system and, when compared with the results obtained during the integrity against deletion test conducted on the traditional blockchain, it shows that the rolling blockchain offers the same integrity against deletion of data as the traditional blockchain; this is critical if this model is to be considered a viable solution to the scalability issue currently faced by all blockchain-based networks.

Preventing the duplication of data is another important requirement that needs to be met. This task is made harder in the rolling blockchain since blocks can be deleted, so unlike the traditional blockchain it is no longer as simple as searching the blockchain to see if the block has been included before. To prevent repetition, each block is given a unique identifier, and requires all the data to be valid before it is entered into the block, which prevents this attack. As such, the rolling blockchain maintains the integrity and authenticity of data.

Comparing this result with those obtained from the same experiment conducted on the traditional blockchain, shows that the same results were achieved. This once again shows that the rolling blockchain offers the same protection and security properties as the traditional blockchain.

### 4.3 Summary of the experiments conducted on our formal model of the rolling blockchain

We proposed a new method to reduce the resources required to be donated by each user on the network, which in turn solves the scalability issue currently facing all blockchain-based networks. For the proposed solution to be suitable for the scalability problem and to be able to replace the traditional blockchain, it needed to provide the same security properties to the data contained within the blockchain as the traditional blockchain, which is why this section conducted several experiments which aimed to show whether the rolling blockchain achieved this.

Overall, the results obtained from the experiments conducted on the formal model of the rolling blockchain protocol show that the rolling blockchain achieves all the formal requirements which were created.

The experiments conducted tested the formal model for the same properties as the traditional blockchain, therefore allowing the results to be easily compared. The results showed that even though the rolling blockchain deleted data from the blockchain, it provided no less security than the traditional blockchain, and achieved the same security principles – particularly data integrity – showing that the rolling blockchain is a viable solution to replace the traditional blockchain.

If a blockchain-based network were to replace the blockchain protocol with the rolling blockchain, it not only prevents the constant growth in the blockchain by being able to keep the blockchain at a constant size, but it also allows a greater number of users to participate in the network, as those who were previously unable to participate in the network due to its high cost of entry would then be able to participate. This not only allows for a scalable network, but also increases the security of the network due to the increase in hashing power that an attacker would be required to control if they were to successfully conduct a 51% attack.

Overall, the rolling blockchain is able to solve the scalability issue of storage-based blockchain systems, which currently affects reputation systems implemented on the blockchain, while maintaining the core security principles held by the traditional blockchain model.

## 5 Conclusion

In this paper we have provided a detailed discussion of formal methods and their advantages to software development, and have applied them for the first time to the blockchain, the underlying protocol of Bitcoin and other cryptocurrencies.

This paper created the first formal requirements for a blockchain-based network and, using the B language, we accurately modelled the blockchain. Using this model, we conducted a series of experiments on the formal model of the blockchain to test whether long standing assumptions about the security properties provided by blockchain are correct, with a specific focus on the integrity of the data stored in the blockchain.

The results of our experiments confirmed that the traditional blockchain provides the core security principles of Confidentiality, Integrity and Authenticity even when under attack from an attacker who controls less than a majority of the network.

This paper then examined a fundamental issue facing all blockchain-based networks; scalability, which results in users being excluded from participating due to the size of the blockchain. The paper then proposed a solution to this issue; a rolling blockchain. We then implemented a formal model of the rolling blockchain in the B language, and conducted several experiments to test whether periodically deleting data from the blockchain would affect the security properties of the rolling blockchain, and if the underlying model is able to be subverted by an attacker. The experiments conducted on the formal model of the rolling blockchain were the same as those conducted on the traditional blockchain. This showed that the rolling blockchain maintains the key security principles, provides the same security properties as the traditional blockchain, and does not introduce any additional vulnerabilities.

Our results suggest that the rolling blockchain is a viable alternative to the traditional blockchain, as it maintains the core security principles, especially data integrity, and is as secure against data manipulation and attack as the traditional blockchain, while also solving key fundamental issues, such as the scalability of the blockchain, and the inclusion of low-resourced users into the network.

Overall, this paper aimed to propose a solution to the rolling blockchain and demonstrate that it is able to maintain the same security properties as the traditional blockchain. However, this is just the foundation of the idea and there is scope for a lot more research to be conducted in the future in various areas to ensure the rolling blockchain is capable of replacing all blockchain-based systems in the real world.

## 6 Future work

This paper has shown that the underlying protocol to Bitcoin, the blockchain, provides the core security principles of Confidentiality, Integrity and Authenticity. We also demonstrated how a rolling blockchain can, for certain applications, solve the scalability issue currently facing the blockchain while maintaining the security principles, especially data integrity. However, this is just the beginning of development of this network, and there are still many avenues of research left to pursue in this area.

Arguably the most important piece of work to conduct in the future is to make this proposed network live. This will then let us examine in greater detail whether the assumptions in this paper hold true against a real world adversary, who control various percentages of the network.

The deployment onto a real world network would also allow us to see whether our solutions to known issues and limitations hold true, or if new issues surface. It would also allow more research into possible attack vectors, such as an offline chain attack, and the effect this would have on the integrity of the data, as well as possible ways to prevent this attack.

Finally, another key research area is implementing this network, for example on a distrusted reputation system, to accurately model how the roll of the blockchain should be performed: i.e. whether a simple time period is sufficient, or if a more sophisticated model is required.

These are just some of the interesting research areas that we have yet to fully analyse and, with more research, this project could lead to the next generation of blockchain systems.

## 7 References

1. Abrial, J., Schuman, S., Meyer, B.: A Specification Language. In. Macnaghten, A., McKeag, R.: On the Construction of Programs, Cambridge University Press, Cambridge (1980)
2. Barber, S., Boyen, X., Shi, E., Uzun, E.: Bitter to Better – How to Make Bitcoin a Better Currency, https://crypto.stanford.edu/~xb/fc12/bitcoin.pdf
3. Bicarregui, J., Clutterbuck, D., Finnie, G., Haughton, H., Lano, K., Lesan, H., Marsh, D., Matthews, B., Moulding, M., Newton, A., Ritchie, B., Rushton, T., Scharbach, P.: Formal Methods Into Practice: Case Studies In The Application Of The B Method. In: IEE Proceedings Software Engineering (144,2), pp. 119-133. IET, (1997)
4. Bowen, J., Hinchey, M.: Seven More Myths of Formal Methods: Dispelling Industrial Prejudices, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.18.2582&rep=rep1&type=pdf
5. Coin Market Cap, https://coinmarketcap.com/all/views/all/
6. CoinDesk, http://www.coindesk.com/data/bitcoin-daily-transactions/
7. CoinDesk, http://www.coindesk.com/price
8. Diller, A., Docherty, R.: Z and Abstract Machine Notation: A Comparison, http://www.cantab.net/users/antoni.diller/papers/s2.pdf
9. Drainville, D.: An Analysis of the Bitcoin Electronic Cash System, https://uwaterloo.ca/combinatorics-and-optimization/sites/ca.combinatorics-and-optimization/files/uploads/files/drainville_danielle.pdf
10. Dumas, J., Sygnet, P., Xuereb, V.: Bitcoin a Peer-to-Peer Payment Solution, https://www.semanticscholar.org/paper/Bitcoin-a-Peer-to-peer-Payment-Solution-security-Dumas-Joseph/7a1e2a9e0fa3b9e64d09c0587ce302dfe7a32ee3/pdf

11. Eyal, I., Gün Sirer, E.: Majority is not Enough: Bitcoin Mining is Vulnerable, https://www.cs.cornell.edu/~ie53/publications/btcProcFC.pdf
12. Hall, A.: Seven Myths of Formal Methods, http://www.cs.um.edu.mt/gordon.pace/Teaching/FormalMethods/2006Papers/SevenMyths.pdf
13. Hoare, C.: An Axiomatic Basis for Computer Programming, https://www.cs.cmu.edu/~crary/819-f09/Hoare69.pdf
14. Kaur, A., Gulati, S., Singh, S.: Analysis of Three Formal Methods – Z, B and VDM, http://www.ijert.org/view-pdf/297/analysis-of-three-formal-methods-z-b-and-vdm
15. Knight, J., DeJong, C., Gibble, M., Nakano, S.: Why are Formal Methods Not Used More Widely?, http://www.cs.virginia.edu/~jck/publications/lfm.97.pdf
16. Kossak, F., Mashkoor, A.: How To Select The Suitable Formal Method For An Industrial Application: A Survey. In: Butler, M., Schewe, K, Mashkoor, A., Biro, M. (eds.) Abstract State Machines, Alloy, B, TLA, VDM, and Z: 5th International Conference, ABZ 2016, Linz, Austria, May 23-27, 2016, Proceedings, pp. 213-228. Springer, (2016)
17. Kroll, J., Davey, I., Felten, E.: The Economics of Bitcoin Mining or Bitcoin in the Presence of Adversaries, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.364.5595&rep=rep1&type=pdf
18. Lano, K.: The B Language and Method: A Guide to Practical Formal Development, Springer-Verlag, London (2012)
19. Lecomte, T., Servat, T., Pouzancre, G.: Formal Methods in Safety-Critical Railway Systems, http://www.methode-b.com/wp-content/uploads/sites/7/dl/thierry_lecomte/Formal_methods_in_safety_critical_railway_systems.pdf
20. Leuschel, M., Butler, M.: The ProB Animator and Model Checker for B – A Tool Description, http://users.ecs.soton.ac.uk/mal/systems/prob_tooldescription.pdf
21. McConaghy, T., Marques, R., Müller, A., De Jonghe, D., McConaghy, T., McMullen, G., Henderson, R., Bellemare, S., Granzotto, A.: Bigchain DB: A Scalable Blockchain Database, https://www.bigchaindb.com/whitepaper/bigchaindb-whitepaper.pdf
22. Meadows, C.: Formal Methods For Cryptographic Protocol Analysis: Emerging Issues and Trends. In: IEEE Journal on Selected Areas in Communications (21, 1), pp. 44-54. IEEE, (2003)
23. Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System, https://bitcoin.org/bitcoin.pdf
24. Pandey, T., Srivastava, S.: Comparative Analysis of Formal Specification Languages Z, VDM and B, http://inpressco.com/wp-content/uploads/2015/06/Paper1082086-2091.pdf
25. Pilkington, M.: Blockchain Technology: Principles and Applications, http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2662660
26. Poon, J., Dryja, T.: The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments, https://lightning.network/lightning-network-paper.pdf
27. Rawson, M., Allen, P.: Synthesis – An Integrated, Object-Oriented Method and Tool for Requirements Specification in Z. In: Bryant, A., Semmens, L. (eds.) Proceedings of the Methods Integration Workshop, pp. 1-22, Springer (1996)
28. Smith, D.: Generating Programs plus Proofs by Refinement, http://vstte.ethz.ch/Files/smith.pdf
29. Sompolinsky, Y., Zohar, A.: Secure High-Rate Transaction Processing in Bitcoin, http://fc15.ifca.ai/preproceedings/paper_30.pdf
30. Stajano, F.: Security for Whom? The Shifting Security Assumptions of Pervasive Computing. In: Okada, M., Pierce, B., Scedrov, A., Tokuda, H., Yonezawa, A. (eds.) Software Security – Theories and Systems, pp. 16-27, Springer-Verlage, Berlin. (2003)

31. Voros, N., Mueller, W., Snook, C.: An Introduction to Formal Methods, https://www.hni.uni-paderborn.de/en/publications/publikationen/?tx_hnippview_pi1%5Bpublikation%5D=2052&tx_hnippview_pi1%5Bfelder%5D%5Blade%5D=972