A temporal blockchain: A formal analysis

Richard Dennis
School of Computing
University of Portsmouth
United Kingdom
Richard.dennis@port.ac.uk

Gareth Owenson
School of Computing
University of Portsmouth
United Kingdom
Gareth.owenson@port.ac.uk

Benjamin Aziz
School of Computing
University of Portsmouth
United Kingdom
Benjamin Aziz@port.ac.uk

***Abstract*—This paper presents a possible solution to a fundamental limitation facing all blockchain-based systems; scalability. We propose a temporal "rolling" blockchain which solves the problem of its current exponential growth, instead replacing it with a constant fixed-size blockchain. We conduct a thorough analysis of related work and present a formal analysis of the new rolling blockchain, comparing the results to a traditional blockchain model to demonstrate that the deletion of data from the blockchain does not impact on the security of the proposed blockchain model before concluding our work and presenting future work to be conducted.**

Keywords: Cryptocurrencies, blockchain, peer-to-peer, cryptography, distributed systems, distributed storage

I. INTRODUCTION

The blockchain was first described by a developer using the pseudonym of Satoshi Nakamoto in 2008 in his paper describing the Bitcoin protocol [23]. The blockchain, originally implemented for the virtual cryptocurrency, Bitcoin, is a novel peer-to-peer approach which links a sequence of transactions or events together in a way that makes them immutable.

Bitcoin is the most successful blockchain-based network; it 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 blockchain is a public ledger of all transactions that have ever been completed since the first "genesis" block. Each transaction from the Bitcoin protocol is broadcast to all nodes in the network which are maintaining the blockchain, known as miners.

Blockchain-based networks have not properly addressed the issue of scalability; this causes the original decentralized nature of the blockchain to become increasingly centralized, as only the highest-resourced users are able participate in the network. This is because each node on the network is required to store the entire blockchain, which stores every transaction since its deployment and consequently low-resourced users – such as mobile users – are excluded from the network.

To solve the issue of blockchain size that is facing not only the Bitcoin network, but all blockchain-based networks, we have created a new method, which can reduce the size of the Bitcoin blockchain from its current – and increasing – size of 71.8GB to a constant size of just 4.5GB. This new blockchain would remain this size irrespective of network size, or the length of time that the network has been deployed. As such, it not only reduces the cost of entry for low-resourced users, but also increases the network's security.

Despite extensive development over many years and having demonstrated significant benefits, 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 a formal analysis of the newly proposed temporal "rolling" blockchain using the B language. It will examine the security principles of the proposed model and conduct an in-depth analysis of the results, which will be compared to the security principles of traditional blockchain networks. Our aim is to demonstrate that our proposed model is a possible replacement the traditional blockchain, and is capable of solving the scalability issue without introducing any additional security issues into the core data structure.

The structure of this paper is as follows; first we examine related work conducted in the field of the blockchain and also formal methods. We then propose a solution to blockchain's scalability problem – a temporal blockchain – and present the formal analysis of this new model using the B language, conducting an in-depth evaluation against the traditional blockchain models. We then propose future work for this area of research, before summarizing our findings and concluding the paper.

II. RELATED WORK

A. 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, along with First Order Predicate Calculus [15].

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 states 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.

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] summarizes Z's focus as being the formalization of requirements rather than the correct executable implementation of the specification. This summary is expanded on by Kaur et al., who explain that Z is a high level abstract model of the system requirements, which only provides a base to design and test the system [14], while Diller and Docherty add that there is no method to develop the abstract model into 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 Z. Given B's origins, it is perhaps unsurprising that B notations at an abstract level are almost identical to Z's [17]. It has been claimed 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.

B. 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 that links 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 and movement of digital assets [21], and the success of Bitcoin has been attributed solely to the blockchain [25].

Drainville 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, 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 wide-spread 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 summarize 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 arguably 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.

III. ROLLING BLOCKCHAIN OVERVIEW

With the increase in adaptation of blockchain-based networks, such as Bitcoin, the fundamental limitations of all blockchain-based networks are now being realized. Currently, a

major limitation of the adaptation 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.

Low-resourced users, for example users on mobile devices, can no longer become a full node due to the requirement to download the complete blockchain. Although a mobile user can participate in the network, for example to make a transaction, they do not contribute any resources to the network.

Low-resourced users will increase as the blockchain networks increase in size. If a blockchain-based network were to go mainstream, it would be unreasonable to ask each home user to donate hundreds of gigabytes of their hard drive just to participate in the network. Consequently, the size of the blockchain is a key barrier in the mass adaptation of a blockchain-based network that aims to move beyond the hobbyist user which currently use the network.

Users are clearly unwilling to be a full node, as evidenced by the formation of "mining pools". Here, the nodes participating in the network do not need to download the blockchain, and instead just donate hashing power to the pool. Although this method has many advantages for the nodes, such as less resources being required in order to participate in the network, it leads to centralization of the network. As an example, the mining pool GHASH.IO currently controls over 50% of the Bitcoin network.

Excluding low-resourced users, such as casual home users and mobile users, also has a detrimental impact on the security of the blockchain. With lower-resourced users being able to participate in the network as a full node, the donation of the CPU power would increase the total amount of hashing power held by the network. As a result, more resources would be required to conduct a 51% attack and the attack would be less likely to succeed. It would also reduce the probability of any partitioning attacks from succeeding. These reasons clearly show that the exclusion of low-resourced users cannot be taken lightly, and every effort should be made to enable users of all levels to participate in the network.

Blockchain technology has applications beyond virtual currency and, with research into temporary, distributed data storage and reputation systems on the blockchain already being conducted, it is evident that the blockchain no longer needs to store all data since the creation of the network.

Reducing the blockchain is currently possible using the pruning method. This involves the download of the entire blockchain and the client then manually searching through the blockchain to remove any "spent transactions". However, this method requires the user to be able to first download the entire blockchain. Mobile users cannot do this due to storage limitations, and this problem will only increase as the network grows. 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.

The rolling blockchain implements fully decentralized and trustless checkpoints on a blockchain network, and thus created a self-deleting and self-managing blockchain.

Unlike previous solutions to the scalability issue, miners are not required to download the entire consensus, and delete "spent blocks" manually. Our solution aims to be globally accepted, and will no longer require any node to store the blockchain history from the first "genesis" block.

At a set point every day, our method requires the miners who are mining the current block to add a checkpoint to this block, in which all blocks older than 30 days can be safely removed. Since this occurs every 24 hours, only 24 hours of data (144MB) would be removed each day.

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.

Our proposed 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 71.8GB to just 4.36GB.

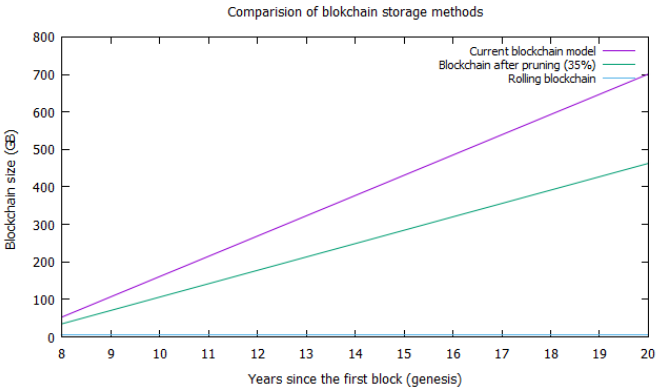

Figure 1: Comparison of blockchain scalability methods

Figure 1 shows the current Bitcoin blockchain, both before and after pruning, as well as 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 1 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 extent than a network using a traditional blockchain.

Simulation of the rolling blockchain demonstrated that when the blockchain was kept at a constant size of 4.5GB, the amount of network traffic caused by nodes joining the network was reduced significantly – 50.34TB daily, based on the average of 15.89% turnover of nodes on the network – as observed on the live Bitcoin network over a period of 30 days.

The reduced network traffic would impact on the propagation of new transactions over the network, reducing the propagation delay. This issue which is likely to become critical as the network grows and the block size increases. It also reduces the cost of entry to the network, as with the reduced network traffic and download being required, users in areas with slow and limited internet connections are able to participate, something which on current blockchain-based networks has to date not been possible.

IV. Our formal analysis method

No previous formal analysis has been conducted on any blockchain-based networks. In the following sections we conduct the first formal analysis of a blockchain-based network and analyze whether our temporal "rolling" blockchain model adheres to the three guiding principles in information security of Confidentiality, Integrity and Authenticity [30].

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 behavior correctly.

Since formal requirements have never been defined for blockchain-based systems, we defined a set of requirements for each of the core principles in information security using assumed goals of the traditional blockchain-based system, although these were altered slightly for our temporal blockchain.

- Confidentiality – Whether data contained within the blockchain be copied, removed or falsely created by a defined type of attack. Will focus on ensuring the model of the temporal "rolling" blockchain is mathematically secure, and blocks cannot be duplicated. We do not consider attacks such as the client being compromised in our model.
- Integrity – Once a block is confirmed and added to the blockchain, it cannot be modified or removed unless it is scheduled to be removed because of the "roll". This is the section where we focus much of the research in this paper.
- Availability – Ensure no authorized users are denied service. Due to the peer-to-peer nature and the fact that each node has a full copy of the blockchain, as long as a single node is online, the network would continue to function correctly

The modelling conducted in the B language will focus on the integrity property of the temporal blockchain. The specifications for the integrity property of the temporal blockchain are:

- Data can only be inserted into the blockchain if valid (The data has a matching hash to the contents contained within the block – we assume the block contents are valid in this model)
- A block must be unique and not a duplication of a previous block.
- A block must have a unique identifier, as well as a cryptographic link to a previous valid block
- After a pre-defined time, (30 days) data will be removed from the blockchain

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:

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

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 behavior but is true as long as no malicious actions have been 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 tool, 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.

A. *Our experiments*

We modelled our temporal blockchain using the B language as can be seen in Figure 2.

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. A comparison against the tradition blockchain security principles will be carried out after each operation to allow us to accurately evaluate if the temporal blockchain achieves the same security principles as the traditional blockchain.

The invariant defined in Figure 2 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 that 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.

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

VARIABLES
  accounts, transactions,
  cryptographic_link, confirmation,
  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, confirmation,
  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 2: Temporal blockchain in B

The conduct_roll 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 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. To ensure consistency with the results all operations where conducted on the same machine with the same amount of resources provided to them.

The 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 temporal blockchain is mathematically correct.

It also shows that it is possible to implement a rolling blockchain and disproves the 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 operations demonstrated in the rest of this paper attempt to subvert the main protocol of the temporal rolling blockchain to invalidate the formal specifications.

Figure 3 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

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

Figure 3: operation attempting pre-emptive deletion of a data block

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 temporal 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 whether 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 temporal rolling blockchain maintains the integrity and authenticity of data.

B. Summary of the experiments conducted on our formal model of the temporal rolling blockchain

The temporal blockchain proposed in this paper reduces the resources required to be donated to network by the nodes participating in the network, reducing the scalability issue currently facing all traditional blockchain-based networks.

Before the temporal blockchain can be further developed to be considered a serious replacement of the traditional blockchain, it was vital that the security properties of the temporal blockchain were evaluated and compared to those of the traditional blockchain model.

We created a formal model of the temporal blockchain, focusing on the integrity of the data contained within it, as shown in the previous section. The operations conducted on the formal model attempted to subvert the protocol into adding invalid data to the blockchain.

The results obtained from the operations conducted on the formal model demonstrate that the temporal blockchain is a suitable replacement for the traditional blockchain model, and achieved all the formal requirements.

In the traditional blockchain model, data duplication is impossible, and data modification is also impossible unless an

attacker controls a majority of the network, and the difficultly of conducting a "51" attack grows exponentially the further in the past the block requiring modification is. The traditional model also requires a block to only be inserted if the block has the correct hash for the block, is cryptographically linked to a previous block, and contains valid data. These are the criteria our proposed model must fulfil if it is to be considered successful.

The operations conducted on the formal model shows the temporal blockchain achieves all the same principles as the traditional blockchain.

The temporal blockchain demonstrated with the add_block operation that only valid block data with the correct block hash and the correct cryptographic link could be added to the blockchain. This operation was tested with incorrect data, and 1000 use cases were run to ensure that this operation did not add any incorrect data on to the blockchain, which it did not. This shows that this operation behaves in the same way as a traditional blockchain model would, and shows the temporal model prevents invalid data and duplicate data from being added to the temporal blockchain.

The requirement of no data modification occurring within the temporal blockchain, and thus the assurance of complete data integrity, was also challenged in our model via two data_deletion operations, a naïve operation and a more complex operation. The naïve operation, which simply tried to remove the block, resulted in the invariant becoming invalid when the operation was ran. The more complex operation, meanwhile, attempted to change the cryptographic link, and was successful. This however does not mean that our protocol is weaker than the traditional blockchain models. To conduct this attack on a live network, an attacker would need a majority of the hashing power on the network; our adversary model focuses on the attacker having less control than this, and also this result is the same as if an attacker would have conducted this on the traditional blockchain network with 51% of the network hash. To reinforce this claim, a model of the traditional blockchain was created and this same operation was run; the results obtained showed that both the temporal and traditional blockchain provide the same level of protection against an attacker trying to remove a valid block from the blockchain.

This result demonstrates that the temporal blockchain, which enables data to be deleted at a specific point in time, does not allow blocks to be deleted pre-emptively – an important feature if this blockchain is to replace the traditional blockchain.

These results show that the added function of deletion of data from the blockchain at a pre-determined time does not add any additional vulnerabilities to the structure of the blockchain and the temporal model achieves the same security principles as the traditional blockchain that is currently implemented in Bitcoin.

Overall, the temporal 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.

V. Conclusion

In this paper we have provided a detailed discussion of formal methods and their advantages to software development, and have applied these to an improved blockchain, known as a temporal "rolling" blockchain.

The temporal blockchain was proposed in detail in this paper, in which core formal requirements were set out before being modelled using the B language. Several experiments were conducted to test whether periodically deleting data from the blockchain would affect the security properties of the rolling blockchain, and to see whether the underlying model is able to be subverted by an attacker. Through the modeling of the temporal blockchain, we conducted several operations which tested the security principles of the temporal blockchain, with a focus on the integrity of the data contained within the blockchain. We ere then able to compare these results with those of the traditional blockchain model.

The results of our experiments confirmed that the temporal rolling blockchain maintains the key security principles and 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. In addition, it solves key fundamental issues, such as the scalability of the blockchain, and enables low-resourced users to be included in the network.

Overall, this paper aimed to propose a solution to the problems faced by the traditional blockchain. This solution is the rolling blockchain, which we have shown to be able to maintain the same security properties as the traditional blockchain. However, this is just the foundation of this idea and there is scope for a lot more research to be conducted in various areas to ensure the rolling blockchain is capable of replacing all blockchain-based systems in the real world.

VI. Future work

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 controls 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 that 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 analyze and, with more research, we believe that this project could lead to the next generation of blockchain systems.

VII. 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 Proceed-ings Software Engineering (144,2), pp. 119-133. IET, (1997)

[4] Bowen, J., Hinchey, M.: Seven More Myths of Formal Methods: Dispelling Industrial Prej-udices, 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, Spring-er-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 Descrip-tion, 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 Data-base, 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 Compu-ting. In: Okada, M., Pierce, B., Scedrov, A., Tokuda, H., Yonezawa, A. (eds.) Software Se-curity – 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