Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

# Deconstructing and Reconstructing Blockchain Analyses

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**ΜΑΝΤΑΛΟΣ ΜΑΡΙΟΣ**

**Επιβλέπων :** Αριστείδης Παγουρτζής
Καθηγητής ΣΗΜΜΥ Ε.Μ.Π.

Αθήνα, Οκτώβριος 2024

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών

# Deconstructing and Reconstructing Blockchain Analyses

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**ΜΑΝΤΑΛΟΣ ΜΑΡΙΟΣ**

**Επιβλέπων :** Αριστείδης Παγουρτζής
Καθηγητής ΣΗΜΜΥ Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 22η Οκτωβρίου 2024.

| | | |
|---|---|---|
| Αριστείδης Παγουρτζής<br>Καθηγητής ΣΗΜΜΥ Ε.Μ.Π. | Νίκος Λεονάρδος<br>Καθηγητής ΣΗΜΜΥ Ε.Μ.Π. | Βασίλης Ζήκας<br>Καθηγητής Gerorgia Tech |

Αθήνα, Οκτώβριος 2024

**Μάνταλος Μάριος**

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

# Περίληψη

Στην παρούσα διπλωματική μελετάμε τις μεθόδους ανάλυσης του blockchain, μιας κρυπτογραφικής κατασκευής η οποία στοχεύει στη δημιουργία ενός αποκεντρωμένου καταλόγου συναλλαγών. Το blockchain, ως σχετικά πρόσφατη τεχνολογία έχει γνωρίσει πολύ μεγάλη ανάπτυξη τα τελευταία χρόνια τόσο σε θεωρητικό όσο και σε τεχνικό επίπεδο. Για αυτό το λόγω θεωρούμε πως αξίζει να εμβαθύνουμε στους τρόπους με τους οποίους αναλύεται.

Αρχίζουμε παραθέτοντας τα μαθηματικά εργαλεία που μας επιτρέπουν να κατασκευάσουμε το blockchain και να αναλύσουμε τις ιδιότητές του. Έπειτα εξερευνούμε τρεις προσπάθειες ανάλυσης του, κάθε μία από τις οποίες χρησιμοποιεί διαφορετικές μεθόδους για να αποδείξει επιθυμητές ιδιότητες. Υπογραμμίζουμε τις διαφορές τους αλλά και τον τρόπο με τον οποίο έχουν αλληλεπιδράσει και επηρεάσει η μία την άλλη.

Τέλος, δημιουργούμε ένα δικό μας, απλοποιημένο μοντέλο του blockchain, το λεγόμενο Μοντέλο των Σφραγίδων, το οποίο χρησιμοποιούμε για να αναπαράξουμε την ανάλυση που μελετήσαμε αποδεικνύοντας τις ιδιότητές του. Στην πορεία τονίζουμε τις διαφορές που προκύπτουν στην ανάλυση εξαιτίας της απλοποίησης αλλά και τις μεθόδους που χρησιμοποιούνται για να καλύψουν αυτό το κενό.

## Λέξεις κλειδιά

κρυπτογραφία, blockchain, bitcoin, ανάλυση blockchain, αποκεντρωμένο, απόδειξη εργασίας

# Abstract

In this diploma thesis we study the methods of analyzing the blockchain, a cryptographic construction with the goal of creating a decentralized ledger of transactions. Blockchain, as a relatively recent technology has had a great amount of development in the last year both in a theoretical and a technical level. For this reason we believe it is worthwhile to delve into the ways it is analyzed.

We begin by stating the mathematical tool that enable us to construct the blockchain and analyze its properties. Then we explore three attempts to analyze it, each using different methods to prove our desired properties. We underline their differences as well as the way in which they have interacted and influenced each other.

Finally, we create our own, simplified blockchain model, called the Stamp Model, which we use to reproduce the analysis we have studied, proving its properties. In the course of this we highlight the differences that occur in the analysis due to the simplification as well as the methods used to bridge this gap.

## Key words

blockchain, blockchain analysis, cryptography, consensus, bitcoin, decentralized, proof of work

# Ευχαριστίες

Αυτή η σελίδα παρέμεινε σκοπίμως κενή μέχρι τη στιγμή όπου το τέλος ήταν πλέον βέβαιο. Η διεκπόνηση αυτής της διπλωματικής ήταν, και είναι όσο γράφω αυτό το κείμενο εξαιρετικά ψυχοφθόρα, πράγμα που μπορεί κανείς να καταλάβει και από τη χρονική της διάρκεια, εύτασε όμως επιτέλους στο τέλος της.

Αρχικά θέλω ευχαριστήσω τους καθηγητές Αριστείδη Παγουρτζή και Βασίλη Ζήκα για την ευκαιρία που μου έδωσαν να εργαστώ μαζί τους σε αυτή τη διπλωματική. Έπειτα να υπερευχαριστήσω την Pourandokht Behrouz για την συνεχή της υποστήριξη, κινητοποίηση και ενδιαφέρον χωρίς τα οποία δε θα ήταν ποτέ δυνατό να φτάσω εδώ. Θα ήθελα τέλος να ευχαριστήσω όλους εκείνους, παρόντες και απόντες, που στάθηκαν κοντά μου καθ'όλη την περίοδο της φοίτησής μου στη σχολή. Θα ήταν πραγματικά δύσκολο να τους βάλω σε σειρά και να περιγράψω την απεριόριστη προσφορά τους και τη συμβολή τους σε αυτό που είμαι σήμερα.

Με την σχολή πίσω μου κοιτάω προς το μέλλον και είμαι έτοιμος για τα επόμενα βήματα.

Μάνταλος Μάριος,

Αθήνα, 22η Οκτωβρίου 2024

# Περιεχόμενα

# Κατάλογος σχημάτων

**Σχήματα στο αγγλικό κείμενο**

# Εκτεταμένη Ελληνική Περίληψη

## 0.1 Blockchain

### 0.1.1 Εισαγωγή

Η διπλωματική αυτή αφορά τη μελέτη του Blockchain, ενός πεδίου της Κρυπτογραφίας που άνθησε με τη δημοσίευση του Bitcoin paper από τον Satoshi Nakamoto [1]. Από τότε το blockchain έχει αναπτυχθεί τόσο σε θεωρητικό όσο και σε εφαρμοσμένο επίπεδο. Το πρώτο αφορά τον τρόπο με τον οποίο αυτή η τεχνολογία μπορεί να λύσει το πρόβλημα της Βυζαντινής Συναίνεσης (Byzantine Consensus). Αυτό μας καλεί να δημιουργήσουμε ένα τρόπο έτσι ώστε $n$ ανεξάρτητες οντότητες, η κάθε μία με τη δική της άποψη, να καταφέρουν να φτάσουν σε συναίνεση (να αποφασίσουν όλες σε μία από τις απόψεις τους) όταν ανάμεσά τους βρίσκονται κάποιες "βυζαντινές" οντότητες οι οποίες προσπαθούν να αποτρέψουν τη συναίνεση αυτή. Το δεύτερο (πρακτικό) πεδίο, το οποίο είναι και αυτό που δημιουργήθηκε πρώτο, αφορά στην κατασκευή ενός κατανεμημένου καταλόγου συναλλαγών ο οποίος μπορεί να χρησιμεύσει σαν εναλλακτική σε ένα κεντρικό χρηματικό σύστημα (τράπεζα). Ουσιαστικά αποτελεί την πρακτική εφαρμογή της Βυζαντινής Συναίνεσης όπου οι οντότητες είναι οι συμμετέχοντες στο σύστημα και η συναίνεση στην οποία προσπαθούν να καταλήξουν είναι τα χρηματικά ποσά που διαθέτει ο καθένας και η σειρά των συναλλαγών που έχουν πραγματοποιηθεί.

Το blockchain, όπως φαναιρώνει και το όνομά του είναι μία αλυσίδα από blocks. Κάθε block περιέχει έναν αριθμό από συναλλαγές μεταξύ μελών του συστήματος. Για να συνδεθούν μεταξύ τους τα blocks περιέχουν το καθένα μία περίληψη του προηγούμενου η οποία πληρή χαρακτηριστικά που την καθιστούν υπολογιστικά δύσκολη να βρεθεί. Η διαδικασία με την οποία ένας χρήστης καταφέρνει να δημιουργήσει ένα block ονομάζεται εξόρυξη (mining) και ουσιαστικά απαιτεί τη δοκιμή ενός τεράστιου ποσού τυχαίων αριθμών μέχρι να βρεθεί κάποιος ο οποίος θα οδηγήσει σε ένα αποδεκτό block.

### 0.1.2 Εργαλεία

Για να καταφέρουμε να μελετήσουμε τη λειτουργία και τις ιδιότητες του blockchain χρησιμοποιούμε αρκετά μαθηματικά εργαλεία:

**Πιθανοτικοί Πολυωνυμικού-Χρόνου Αλγόριθμοι:**

**Definition 0.1.1** (*Πιθανοτικοί Πολυωνυμικού-Χρόνου Αλγόριθμοι* [2])**.** Ένας αλγόριθμος $A$ λέγεται πιθανοτικός πολυωνυμικού χρόνου (PPT) αν:

1. Υπάρχει πολυώνυμο $pol(\cdot)$ τέτοιο ώστε, για κάθε είσοδο $x \in \{0,1\}^*$, ο υπολογισμός $A(x)$ τελειώνει μέσα σε το πολύ $pol(\|x\|)$ βήματα.

2. Έχει πρόσβαση σε μία πηγή τυχαιότητας η οποία παράγει τυχαία bits τα οποία είναι ανεξάρτητα 1 με πιθανότητα $p$ και 0 με πιθανότητα $1-p$.

Ένα πιθανοτικός αλγόριθμος με $p = \frac{1}{2}$ λέγεται ομοιόμορφος. Εμείς θα χρησιμοποιήσουμε κυρίως μη-ομοιόμορφους αλγορίθμους για κάποιο $p \in (0,1)$. Στόχος μας είναι να έχουμε ένα μέτρο για την υπολογιστική δύναμη των παικτών αλλά κυρίως του Αντιπάλου.

**Αμελητέες Συναρτήσεις:** Για να είμαστε σίγουροι πως κάτι είναι απίθανο να συμβεί δεδομένου πολυωνυμικού χρόνου χρειαζόμαστε τις εξείς συναρτήσεις:

**Definition 0.1.2** (*Αμελητέες Συναρτήσεις*)**.** Μία συνάρτηση $\epsilon(\cdot)$ λέγεται αμελητέα αν για κάθε πολυώνυμο $p(\cdot)$, υπάρχει κάποιο $\kappa_0$ τέτοιο ώστε:

$$\epsilon(\kappa) \leq \frac{1}{p(\kappa)}$$

για όλα τα$\kappa \geq \kappa_0$.

**Definition 0.1.3** (*Strongly Negligible Functions*)**.** A function $\epsilon(\cdot)$ is said to be strongly negligible if there exist constants $c_0 > 0, c_1$ such that:

$$\epsilon(\kappa) \leq e^{-c_0\kappa + c_1}$$

for all $\kappa$.

**Συναρτήσεις Κατακερματισμού:** Όπως αναφέραμε νωρίτερα, για τη δημιουργία ενός block απαιτείται να βρεθεί μία περίληψη το προηγούμενου block, με τρόπο που είναι δύσκολο να αντιστραφεί. Για αυτό χρησιμοποιούνται οι "Συναρτήσεις Κατακερματισμού", για τις οποίες θα βασιστούμε στο [2].

**Definition 0.1.4.** (*Συνάρτηση Κατακερματισμού*) Μία συνάρτηση κατακερματισμού είναι ένα ζεύγος πιθανοτικών πολυωνυμικού-χρόνου αλγορίθμων $(Gen, H)$ που ικανοποιούν τα ακόλουθα:

- Το $Gen$ είναι ένας πιθανοτικός αλγόριθμος ο οποίος παίρνει σαν είσοδο την παράμετρο ασφαλείας $1^\kappa$ και επιστρέφει κλειδί $s$. Υποθέτουμε πως το $1^\kappa$ περιλαμβάνεται στο $s$.
- Υπάρχει πολυωνυμικό $l$ τέτοιο ώστε ο $H$ να είναι ένας (ντετερμινιστικός) πολυωνυμικού-χρόνου αλγόριθμος που παίρνει σαν είσοδο κλειδί $s$ και καποια ακολουθία $x \in \{0,1\}^*$, και επιστρέφει ακολουθία $H^s(x) \in \{0,1\}^{l(\kappa)}$.

Αν για κάθε $\kappa$ και $s$, το $H^s$ ορίζεται μόνο για εισόδους μήκους $l'(\kappa) > l(\kappa)$, τότε λέμε πως το $(Gen, H)$ είναι μία συγκεκριμένου-μήκους συνάρτηση κατακερματισμού με μήκος παραμέτρου $l'$.

**Chernoff Bound:**

**Theorem 0.1.5** (*Multiplicative Chernoff Bound* [3])**.** *Έστω $X_1, ..., X_n$ ανεξάρτητες Boolean τυχαίες μεταβλητές, τέτοιες ώστε για όλα τα $i$, $Pr[X_i = 1] = p$. Επιπλέον, έστω $X$ το άθροισμα αυτών των μεταβλητών, και $\mu = \mathcal{E}[X] = np$ η αναμενόμενη τιμή του αθροίσματος.*
*Τότε, για κάθε $\delta \in (0,1]$:*

$$Pr[X < (1-\delta)\mu] < e^{-\Omega(\delta^2\mu)}$$
$$Pr[X > (1+\delta)\mu] < e^{-\Omega(\delta^2\mu)}$$

**Union Bound:**

**Theorem 0.1.6** (*Union Bound*). *Για ένα πεπερασμένο σύνολο γεγονότων $A_1, A_2, ..., A_n$:*

$$Pr\left[\bigcup_{i=1}^{n} A_i\right] \leq \sum_{i=1}^{n} Pr[A_i]$$

## 0.2 Ανάλυση

Στρέφουμε το βλέμα μας έπειτα σε κάποια από τα κύρια παραδείγματα ανάλυσης του blockchain. Για να το κάνουμε αυτό πρέπει να διατυπώσουμε πρώτα τους στόχους της όποιας τέτοιας ανάλυσης οι οποίοι είναι δύο. Αρχικά πρέπει να εξασφαλίσουμε οτι οι συναλλαγές που δημοσιεύονται απο έντιμους παίκτες καταλήγουν να αποτελούν μέρος των αλυσιδών των έντιμων παικτών, να είναι σίγουρο δηλαδή πως το σύστημα είναι ζωντανό και νέες συναλλαγές εισάγονται σε αυτό. Έπειτα θέλουμε να βεβαιωθούμε πως κάθε συναλλαγή που γίνεται θα αποτελεί για πάντα μέρος του καταλόγου μας, εάν υπήρχε η πιθανότητα κάποια συναλλαγή να "εξαφανιστεί" ξαφνικά από το κατάλογο προφανώς το χρηματικό αυτό σύστημα δε μπορεί να λειτουργήσει. Η ύπαρξη του Αντιπάλου δημιουργεί δυσκολίες στη συμπλήρωση αυτών των απαιτήσεων. Εξαιτίας αυτής της δύναμης του Αντιπάλου αναγκαζόμαστε να εισάγουμε μία καινούρια παράμετρο ασφαλείας, το $T$, η οποία θα αφορά το βάθος στο οποίο πρέπει να βρίσκεται ένα block στην αλυσίδα προκειμένου να μπορεί να πληρεί τις απαιτήσεις μας. Διατυπώνουμε έτσι τους στόχους μας ως εξής:

1. **Ζωντάνια:** Οποιαδήποτε συναλλαγή δημοσιευτεί από έντιμο παίκτη του συστήματος θα φτάσει σε βάθος τουλάχιστον $T$ στην αλυσίδα κάποιου έγκυρου παίκτη.

2. **Επιμονή:** Οποιαδήποτε συναλλαγεί έχει φτάσει σε βάθος τουλάχιστον $T$ στο blockchain θα αποτελεί για πάντα μέρος των αλυσίδων κάθε έντιμου παίκτη.

### 0.2.1 Η πρώτη ανάλυση

Η πρώτη ανάλυση την οποία εξερευνούμε είναι το paper [4] των Garay, Kiagias και Leonardος.

Για αρχή οι συγγραφείς ορίζουν τη κεντρική παραδοχή του paper, η οποία εμφανίζεται και σε κάποια μορφή σε όλες τις αναλύσεις.

**Definition 0.2.1.** (*Παραδοχή Έντιμης Πλειοψηφίας*) $t$ από τους $n$ παίκτες είναι διεφθαρμένοι με:

$$t \leq (1 - \delta)(n - t),$$

όπου $3f + 3\epsilon < \delta \leq 1$.

Έπειτα γίνεται μία κατηγοριοποίση των δυνατών γύρων:

- $X_i = 1$ αν στο γύρο $i$ εξορύχθηκε ένα block από έντιμο παίκτη.
- $Y_i = 1$ αν στο γύρο $i$ εξορύχθηκε ακριβώς ένα block από έντιμο παίκτη.
- $Z_i = m$ αν στο γύρο $i$ εξορύχθηκαν $m$ blocks από διευθαρμένους παίκτες.

Η ανάλυση αυτή βασίζεται στην έννοια της "Τυπικής Εκτέλεσης" (Typical Execution)

**Definition 0.2.2.** (*Τυπική Εκτέλεση*) Μία εκτέλεση είναι $(\lambda, \epsilon)$-τυπική για $\epsilon \in (0, 1)$ και ακέραιο $\lambda \geq 2/f$, αν, για κάθε σύνολο $S$ από τουλάχιστον $\lambda$ διαδοχικούς γύρους, ισχύει:

(a) $(1-\epsilon)\mathbb{E}[X(S)] < X(S) < (1+\epsilon)\mathbb{E}[X(S)]$ και $(1-\epsilon)\mathbb{E}[Y(S)] < Y(S)$.

(b) $Z(S) < \mathbb{E}[Z(S)] + \epsilon\mathbb{E}[X(S)]$.

(c) Καμία εισαγωγή, αντιγραφή ή πρόβλεψη.

Όπου $X(S), Y(S), Z(S)$ είναι τα σύνολα των αντιστοίχων τυχαίων μεταβλητών στην περίοδο $S$.

Η Τυπική Εκτέλεση συμβαίνει με συντριπτική πιθανότητα:

$$1 - e^{-\Omega(\epsilon^2\lambda f + \kappa - \log(L))}$$

Για να αποδείξουν τις ιδιότητες της Ζωντάνιας και της Επιμονής ορίζουν τις ακόλουθες, εξαιρετικά χρήσιμες ενδιάμεσες ιδιότητες:

**Ανάπτυξη Αλυσίδας:** Για κάθε έντιμο παίκτη $P$ με αλυσίδα $\mathcal{C}$, ισχύει πως μετά από κάθε $s$ διαδοχικούς γύρους υιοθετεί μία αλυσίδα που έχει τουλάχιστον $\tau \cdot s$ blocks περισσότερα από $\mathcal{C}$ με παραμέτρους:

$$\tau = (1-\epsilon)f$$
$$s \geq \lambda$$

**Ποιότητα Αλυσίδας:** Για κάθε έντιμο παίκτη $P$ με αλυσίδα $\mathcal{C}$, ισχύει πως για κάθε $l$ συνεχόμενα blocks της $\mathcal{C}$ ο λόγος των έντιμων blocks είναι τουλάχιστον $\mu$ με παραμέτρους:

$$l \geq 2\lambda f$$
$$\mu = 1 - \left(1 + \frac{\delta}{2}\right) \cdot \frac{t}{n-t} - \frac{\epsilon}{1-\epsilon} > 1 - \left(1 + \frac{\delta}{2}\right) \cdot \frac{t}{n-t} - \frac{\delta}{2}.$$

**Κοινό Πρόθεμα:** Για κάθε ζεύγος έντιμων παικτών $P_1$ and $P_2$ με αλυσίδες $\mathcal{C}_1$ και $\mathcal{C}_2$ στους γύρους $r_1 \geq r_2$, ισχύει πως $\mathcal{C}_1[-T:] \preceq \mathcal{C}_2$ με παράμετρο:

$$T \geq 2\lambda f$$

### 0.2.2 Εισαγωγή στις καθυστερήσεις

Φτάνουμε έπειτα στην κύρια πηγή αυτής της διπλωματικής το paper [3] των Pass, Seeman και Shelat.

Η ανάλυση παραμετροποιείται από δύο σταθερές που αντιπροσωπεύουν τις δυνάμεις των έντιμων και αντιπάλων παικτών:

$\alpha \rightarrow$ Η πιθανότητα πως τουλάχιστον ένας έντιμος παίκτης εξορύσει ένα block σε ένα γύρο.

$\beta \rightarrow$ Ο αναμενόμενος αριθμός αντιπάλων block που εξορύσονται κάθε γύρο.

Εκτός από αυτό χρειάζεται να υπάρχει ένα μέτρο για την εξορυκτική δύναμη των έντιμων παικτών λαμβάνοντας υπόψη την Καθυστέρηση:

$$\gamma = \frac{\alpha}{1 + \Delta\alpha}$$

Οι συγγραφείς ορίζουν δύο Περιβάλλοντα πάνω στα οποία θα ποδείξουν τις επιθυμητές ιδιότητες:

- Το πρώτο: $\Gamma_0$ το οποίο δεν έχει περιορισμούς εκτός από την εγκυρότητα.

- Το δεύτερο: $\Gamma_\lambda^p$ όπου ο συσχετισμός δυνάμεων του Αντιπάλου και των έντιμων παικτών είναι:

  $$\alpha(1 - 2(\Delta + 1)\alpha) \geq \lambda\beta$$

  Το οποίο είναι το αντίστοιχο της Παραδοχής Έντιμης Πλειοψηφίας στο [4].

Για κάθε ιδιότητα που θέλουν αν αποδείξουν, οι συγγραφείς δημιουργούν ένα κατηγόρημα το οποίο ισχύει μόνο εάν ισχύει και η ιδιότητα για την αλυσίδα και έπειτα προσπαθούν να αποδείξουν πως αυτά τα κατηγορήματα ισχύουν με συντριπτική πιθανότητα

**Ανάπτυξη Αλυσίδας** Αρχικά ορίζεται ένας τρόπος να μετρηθεί η αύξηση του μήκους της αλυσίδας ανάμεσα σε γύρους.

$$\text{min-chain-increase}_{r,t}(\text{view}) = \min_{i,j} \left\{ |C_j^{r+t}(\text{view})| - |C_i^r(\text{view})| \right\}$$

Με αυτό δημιουργούμε το κατηγόρημα growth$^t$(view, $\Delta, T$),
growth$^t$(view, $\Delta, T$) = 1 iff:

- **(Συνεπές Μήκος)** για κάθε ζεύγος γύρων $r \geq |\text{view}| - \Delta, r' \leq |\text{view}|, r' \leq r + \Delta$, για κάθε ζεύγος παικτών $i, j$ τέτοιο ώστε στο view ο $i$ είναι έντιμος στο γύρο $r$ και ο $j$ είναι έντιμος στο γύρο $r'$, έχουμε πως:

  $$|\mathcal{C}_j^{r'}(\text{view})| \geq |\mathcal{C}_i^r(\text{view})|$$

- **(Ανάπτυξη Αλυσίδας)** για κάθε γύρο $t \leq |\text{view}| - r$, έχουμε:

  $$\text{min-chain-increase}_{r,t}(\text{view}) \geq T$$

Με αυτό το κατηγόρημα ορίζεται η ιδιότητα Ανάπτυξης Αλυσίδας για ένα γενικευμένο πρωτόκολλο blockchain.

**Definition 0.2.3.** Ένα blockchain πρωτόκολλο $(\Pi, \mathcal{C})$ έχει ρυθμό Ανάπτυξης Αλυσίδας $g(\cdot,\cdot,\cdot,\cdot)$ σε Γ-περιβάλλοντα αν για κάθε Γ-αποδεκτό $(n(\cdot), \rho, \Delta(\cdot), A, Z)$, υπάρχει κάποια σταθερά $c$ και αμελητέες συναρτήσεις $\epsilon_1, \epsilon_2$ τέτοιες ώστε για κάθε $\kappa \in \mathbb{N}, T \geq c\log(\kappa)$, και $t \geq \frac{T}{g(n(\kappa),\rho,\Delta(\kappa))}$, το ακόλουθο ισχύει:

$$Pr\left[\text{view} \leftarrow \text{EXEC}^{(\Pi^V,\mathcal{C})}(A, Z, \kappa) : \text{growth}^t(\text{view}, \Delta(\kappa), T) = 1\right] \geq 1 - \epsilon_1(\kappa) - \epsilon_2(T)$$

Επιπλέον, αν $\epsilon_1 = 0$ λέμε πως το $(\Pi, \mathcal{C})$ έχει errorless Ανάπτυξη Αλυσίδας $g$ σε Γ περιβάλλοντα.

**Theorem 0.2.4.** *Για κάθε $\delta > 0$, κάθε $p(\cdot)$, $\left(\Pi^p_{Nak}, \Pi^p_{Nak}\right)$ έχει ρυθμό Ανάπτυξης Αλυσίδας:*

$$g^p_\delta(\kappa, n, \rho, \Delta) = (1-\delta)\gamma$$

*σε $\Gamma_0$ περιβάλλοντα.*

**Ποιότητα Αλυσίδας**

**Definition 0.2.5.** $quality^T(\text{view}, \mu) = 1$ ανν για κάθε γύρο $r$ και κάθε παίκτη $i$ ο οποίος είναι έντιμος στο view$^r$, για κάθε ακολουθία από $T$ διαδοχικά blocks στο $C(\text{view})^r_i$, το κλάσμα των blocks $m$ που είναι έντιμα ως προς view$^r$, είναι τουλάχιστον $\mu$.

**Definition 0.2.6.** Ένα blockchain πρωτόκολλο $(\Pi, \mathcal{C})$ έχει Ποιότητα Αλυσίδας $\mu(\cdot,\cdot,\cdot,\cdot)$ σε Γ-περιβάλλοντα, αν για όλα τα Γ-αποδεκτά $(n(\cdot),\ \rho,\ \Delta,\ A,\ Z)$, υπάρχει κάποια σταθερά $c$ και αμελητέες συναρτήσεις $\epsilon_1,\ \epsilon_2$ τέτοιες ώστε για κάθε $\kappa \in \mathbb{N}, T > c\log(\kappa)$ το ακόλουθο ισχύει:

$$Pr\left[\text{view} \leftarrow \text{EXEC}^{(\Pi^V,\mathcal{C})}(A, Z, \kappa) : quality^T(\text{view}, \mu(\kappa, n(\kappa), \rho, \Delta(\kappa))) = 1\right] \geq 1 - \epsilon_1(\kappa) - \epsilon_2(T)$$

Επιπλέον, αν $\epsilon_1 = 0$ λέμε πως το $(\Pi, \mathcal{C})$ έχει errorless Ποιότητα Αλυσίδας $\mu$ σε Γ περιβάλλοντα.

**Theorem 0.2.7.** *Για κάθε $\delta > 0$, κάθε $p(\cdot)$, $\left(\Pi^p_{Nak}, \Pi^p_{Nak}\right)$ έχει Ποιότητα Αλυσίδας:*

$$\mu^p_\delta(\kappa, n, \rho, \Delta) = 1 - (1+\delta)\frac{\beta}{\gamma}$$

*σε $\Gamma_0$ περιβάλλοντα.*

**Συνέχεια** Η Συνέχεια έχει έναν ενδιαφέρον τρόπο να αντιμετωπίζει την Καθυστέρηση, χρησιμοποιώντας της λεγόμενες "Ευκαιρίες Σύγκλισης", στιγμές κατά τις οποίες οι έντιμοι παίκτες έχουν την ευκαιρία να συγχρονίσουν τις αλυσίδες τους. Μία τέτοια ευκαιρία προκύπτει σε κάθε γύρο που έχει εξορυχθεί ένα μοναδικό έντιμο block και έχει $\Delta$ γύρους σιγής πριν και μετά.

**Definition 0.2.8.** Έστω $consistent^T(\text{view}) = 1$ ανν για όλους τους γύρους $r_1 \leq r_2$ και όλους τους παίκτες $i, j$ τέτοιους ώστε ο $i$ είναι έντιμος στο view$^{r_1}$ και ο $j$ είναι έντιμος στο view$^{r_2}$, έχουμε πως $C^{r_1}_i[-T] \leq C^{r_2}_j$

**Definition 0.2.9.** Ένα blockchain πρωτόκολλο $(\Pi, \mathcal{C})$ ικανοποιεί Συνέχεια σε $\Gamma$ περιβάλλοντα, αν για όλα τα $\Gamma$-αποδεκτά $(n(\cdot), \rho, A, Z)$, υπάρχει κάποια σταθερά $c$ και αμελητέες συναρτήσεις $\epsilon_1, \epsilon_2$ τέτοιες ώστε για κάθε $\kappa \in \mathbb{N}, T > c\log(\kappa)$, το ακόλουθω ισχύει:

$$Pr\left[\text{view} \leftarrow \text{EXEC}^{(\Pi^V, \mathcal{C})}(A, Z, \kappa) : consistent^T(\text{view}) = 1\right] \geq 1 - \epsilon_1(\kappa) - \epsilon_2(T)$$

Επιπλεόν, αν $\epsilon_1 = 0$, λέμε πως $(\Pi, \mathcal{C})$ έχει errorless Συνέχεια σε $\Gamma$-περιβάλλοντα.

**Theorem 0.2.10.** *Για κάθε $\lambda > 1$, κάθε $p(\cdot)$, $\left(\Pi^p_{Nak}, \Pi^p_{Nak}\right)$ ικανοποιεί Συνέχεια σε $\Gamma^p_\lambda$ περιβάλλοντα.*

**Ζωντάνια και Επιμονή** Τέλος οι συγγραφές στρέφουν την προσοχή τους στους αρχικούς στόχους, τη Ζωντάνια και την Επιμονή:
**Ζωντάνια** Έστω $live(\text{view}, t) = 1$ ανν για κάθε $t$ διαδοχικούς γύρους $r, ..., r + t$ στο view υπάρχει κάποις γύρος $r' \in [r, r + t]$ και παίκτης $i$ τέτοιος ώστε: στοview:

1. ο $i$ είναι έντιμος στο γύρο $r'$,
2. ο $i$ έλαβε ένα μήνυμα $m$ ως είσοδο στο γύρο $r'$ και
3. για κάθε παίτκη $j$ ο οποίος είναι έντιμος στον γύρο $r + t$ στο view, $m \in \mathcal{L}^{r+t}_j(\text{view})$.

**Definition 0.2.11** (*Ζωντάνια*)**.** Λέμε πως ένας δημόσιος κατάλογος $(\Pi, \mathcal{L})$ είναι ζωντανός με χρόνο αναμονής $w$ σε $\Gamma$ περιβάλλοντα αν για όλα τα $\Gamma$-αποδεκτά $(n(\cdot), \rho, \Delta(\cdot), A, Z)$, υπάρχει αμελητέα συνάρτηση $\epsilon$ στην παράμετρο ασφαλείας $\kappa \in \mathbb{N}$, τέτοια ώστε:

$$Pr\left[\text{view} \leftarrow \text{EXEC}^{(\Pi, \mathcal{L})}(A, Z, \kappa) : live\left(\text{view}w\left(\kappa, n(\kappa), \rho, \Delta(\kappa)\right)\right) = 1\right] \geq 1 - \epsilon(\kappa)$$

**Επιμονή** Έστω $persist_\Delta(\text{view}) = 1$ ανν για κάθε γύρο $r \leq |\text{view}| - \Delta$, κάθε παίκτης $i$ που είναι έντιμος στο view$^r$ και κάθε θέση $pos \leq |\mathcal{L}^\nabla_\rangle(\text{view})|$, if if $\mathcal{L}^r_i(\text{view})$ περιέχει το μήνυμα $m$ στο $pos$, τότε για κάθε γύρο $r'$ τέτοιο ώστε $r + \Delta \leq r'$ και κάθε έντιμο παίκτη $j$ έχουμε πως το $m$ είναι επίσης στο $pos$ στο $\mathcal{L}^{r'}_j(\text{view})$.

**Definition 0.2.12** (*Επιμονή*)**.** Λέμε πως ένας δημόσιος κατάλογος $(\Pi, \mathcal{L})$ είναι επίμονος σε $\Gamma$ περιβάλλοντα αν για όλα τα $\Gamma$-αποδεκτά $(n(\cdot), \rho, \Delta(\cdot), A, Z)$, υπάρχει μία αμελητέα συνάρτηση $\epsilon$ στη παράμετρο ασφαλείας $\kappa \in (N)$ τέτοια ώστε:

$$Pr\left[\text{view} \leftarrow \text{EXEC}^{(\Pi, \mathcal{L})}(A, Z, \kappa) : persist_{\Delta(\kappa)}(\text{view}) = 1\right] \geq 1 - \epsilon(\kappa)$$

### 0.2.3 Επιστροφή στην πρώτη ανάλυση

Έχοντας δει πως το [3] εισήγαγε την έννοια της Περιορισμένης Καθυστέρησης, ας δούμε πως το [5] την συμπεριέλαβε στο δικό του τρόπο ανάλυσης.

Αρχικά προσαρμόζουν την κεντρική τους παραδοχή.

**Definition 0.2.13.** (*Παραδοχή Έντιμης Πλειοψηφίας- Περιορισμένη Καθυστέρηση*) $t$ από τους $n$ παίκτες είναι διεφθαρμένοι με:

$$t \leq (1-\delta)(n-t),$$

όπου $\epsilon + 2\Delta f + \frac{2\Delta}{\lambda} \leq \frac{\delta}{2}$.

Έπειτα, νέες τυχαίες μεταβλητές πρέπει να οριστούν για να περιγράψουν επιπλέον είδη γύρων:

- $X_i' = 1$ αν στο γύρο $i$ εξορύχθηκε ένα έντιμο block και δεν έχει εξορυχθεί άλλο έντιμο block μέχρι και $\Delta$ γύρους πριν το $i$. Αυτό το ονομάζουμε $\Delta$-απομονομένο μοναδικά επιτυχημένο γύρο.
- $Y_i' = 1$ αν στο γύρο $i$ εξορύχθηκε καριβώς ένα έντιμο block και δεν έχει εξορυχθεί άλλο έντιμο block μέσα σε $\Delta$ γύρους. Αυτό το ονομάζουμαι απομονομένο πετυχημένο γύρο.

Με την εισαγωγή της καθυστέρησης, νοιαζόμαστε πλέον και για το αν τα έντιμα blocks που εξορύσσονται είναι μόνα τους σε μία περίοδο γύρων ίση με την Καθυστέρηση.

**Definition 0.2.14.** (*Τυπική Εκτέλεση - Περιορισμένη Καθυστέρηση*)

(a) $(1-\epsilon)\mathbb{E}[X'(S)] < X'(S) < (1+\epsilon)\mathbb{E}[X(S)]$ και $(1-\epsilon)\mathbb{E}[Y'(S)] < Y'(S)$.

(b) $Z(S) < \mathbb{E}[Z(S)] + \epsilon\mathbb{E}[X'(S)]$.

(c) Καμία εισαγωγή, αντιγραφή ή πρόβλεψη.

**Ανάπτυξη Αλυσίδας** ισχύει με παραμέτρους:

$$\tau = (1-\epsilon)f(1-f)^{\Delta-1}$$
$$s \geq \lambda$$

**Ποιότητα Αλυσίδας** ισχύει με παραμέτρους:

$$l \geq 2\lambda f + 2\Delta f$$
$$\mu = 1 - \frac{1}{(1-\epsilon)(1-f)^{\Delta}} \cdot \frac{t}{n-t} - \frac{\epsilon}{1-\epsilon}\left(1 + \frac{\Delta}{\lambda}\right)$$

**Κοινό Πρόθεμα** ισχύει με παράμετρο:

$$T \geq 2\lambda f + 2\Delta$$

Η ιδιότητες λοιπόν παραμένουν ουσιαστικά οι ίδιες, προσαρμοσμένες για να λαμβάνουν υπόψη την Καθυστέρηση.

### 0.2.4 Μια επιπλέον οπτική

Για να κλείσουμε την εξερεύνηση διαφορετικών προσεγγίσεων της ανάλυσης του blockchain, βλέπουμε συνοπτικά το paper [6] του Ren.

Σε αντίθεση με τα paper που είδαμε, ο Ren δεν επιχειρεί την απόδειξη των τριων ενδιάμεσων ιδιοτήτων αλλά προσπαθεί κατευθείαν να αποδείξει τα:

1. **Ζωντάνια:** Κάθε συναλλαγή καταλήγει στην αλυσίδα όλων των έντιμων παικτών.
2. **Ασφάλεια:** Έντιμοι παίκτες δεν βάζουν διαφορετικές συναλλαγές στο ίδιο ύψος της αλυσίδας. Προφανώς αυτό είναι μία διαφορετική έκφραση της ιδιότητας της Επιμονής.

Όπως και το [6] συμβολίζουμε τις δυνάμεις των έντιμων και αντιπάλων παικτών ως $\alpha$ και $\beta$ αντίστοιχα.

**Definition 0.2.15.** Έστω $g = e^{-\alpha\Delta}$. Έστω $\delta$ μία θετική σταθερά. Η συναίνεση Nakamoto με $T$-επιβεβαίωση εγκυάται Ασφάλεια και Ζωντάνια εκτός με πιθανότητας $e^{-\Omega(\delta^2 g^2 T)}$ αν:

$$g^2\alpha > (1+\delta)\beta$$

Η δική τους κατηγοριοποίηση των γύρων είναι ως εξής:

**Μη-ουραγός:** Ένα έντιμο block χωρίς άλλο έντιμο block εξορυγμένο στους προηγούμενους $\Delta$ γύρους.

**Μοναχικός:** Ένα έντιμο block χωρίς άλλο έντιμο block εξορυγμένο στους προηγούμενους ή επόμενους $\Delta$ γύρους.

Με αυτά τα εργαλεία ο Ren αποδεικνύει τις ιδιότητες:

**Theorem 0.2.16** (*Ζωντάνια*)**.** *Έστω $g\alpha > (1+\delta)\beta$. Στον χρόνο $t$, εκτός με πιθανότητα $e^{-\Omega(\delta^2 g\alpha t)}$, κάθε έντιμος παίκτης βάζει τουλάχιστον $\frac{\delta}{6}g\alpha t - T - 1$ έντιμα blocks στην αλυσίδα του.*

**Theorem 0.2.17** (*Ασφάλεια*)**.** *Έστω $g\alpha > (1+\delta)\beta$. Θεωρούμε κάθε χρόνο $t$ και κάθε block $B$ το οποίο θεωρείται τοποθετημένο από κάποιο έντιμο παίκτη στο χρόνο $t$. Εκτός με πιθανότητα $e^{-\Omega(\delta^2 g^2 \alpha t)}$, για κάθε χρόνο $t' \geq t$, κανένας έντιμος παίκτης δε βάζει block $B' \neq B$ στο ίδιο ύψος με το $B$.*

## 0.3 Το Μοντέλο των Σφραγίδων

Έχοντας θέσει τις βάσεις στις οποίες στηρίζεται το blockchain και έχοντας δει κάποιες από τις κύριες αναλύσεις του προχωράμε στη δημιουργία ενός δικού μας, απλοποιημένου, μοντέλου προκειμένου να μπορέσουμε να τις εφαρμόσουμε στην πράξη. Ο στόχος μας είναι διπλός, από τη μία να κατασκευάσουμε ένα μοντέλο το οποίο θα μπορεί να λειτουργήσει ως βάση για την ανάλυση του blockchain προσφέροντας ένα απλοποιημένο σύστημα πάνω στο οποίο να μπορούν να δοκιμαστούν νέες τεχνικές και ιδέες χωρίς το βάρος όλης της προϋπάρχουσας μαθηματικής ανάλυσης. από την άλλη να παράξουμε ένα εργαλείο διδασκαλίας και κατανόησης του blockchain

μέσα από την πρακτική εφαρμογή της ανάλυσης σε ένα απλοποιημένο μοντέλο και το σταδιακό χτίσιμο πάνω σε αυτή μέχρι την ανακατασκευή της πλήρους ανάλυσης.

Οδηγούμαστε έτσι στο Μοντέλο των Σφραγίδων. Για να καταφέρουμε να δημιουργήσουμε μία αφηρημένη μορφή του blockchain είναι απαραίτητο να απλοποιήσουμε τη διαδικασία της εξόρυξης. Η λύση ενός υπολογιστικά απαιτητικού προβλήματος αντικαθύσταται από τη "Σφραγίδα" (Stamp). Ένα block είναι έγκυρο και μπορεί να τοποθετηθεί στην αλυσίδα μόνο εάν διαθέτει μία Σφραγγίδα για τον γύρο στον οποίο εκδόθηκε. Η Δημιουργία Σφραγγίδων γίνεται από το "Μαντείο" (Oracle), το οποίο είναι επιφορτισμένο με το να δίνει μία σφραγγίδα κάθε γύρο σε κάποιο έντιμο ή αντίπαλο παίκτη.

### 0.3.1 Παραλλαγές

Η απλότητα του μοντέλου μας επιτρέπει να θεωρήσουμε διάφορες παραλλαγές τους ανάλογα με την κατεύθυνση προς την οποία θέλουμε να εμβαθύνουμε:

- **Πολλαπλές Σφραγίδες:** Αλλάζοντας τη λειτουργία του Μαντείου μπορούμε να επιτρέψουμε την δημιουργία περισσότερων από μίας Σφραγίδας σε κάθε γύρο.
- **Παρακράτηση Block:** Μπορούμε να δώσουμε την δυνατότητα στον Αντίπαλο να μην αποκαλείπτει κατευθείαν τα blocks τα οποία έχει σφραγίσει αλλά να τα κρατάει μεχρίς ότου θεωρεί πως είναι βέλτιστο.
- **Καθυστέρηση:** Φυσικά μπορούμε να εισάγουμε και την κεντρική έννοια του [3], αυτή της Καθυστέρησης η οποία δραστικά αυξάνει την ισχύ του Αντιπάλου και επηρεάζει την ανάλυση του μοντέλου.

Με βάση αυτό το απλό μοντέλο, προχωράμε στην αναδόμηση της ανάλυσης με πρότυπό μας το [3].

### 0.3.2 Ανάπτυξη Αλυσίδας

**Theorem 0.3.1.** *Για κάθε $\delta > 0$, $\left(\Pi^p_{Stamp}, \mathcal{C}^p_{Stamp}\right)$ έχει (errorless) ρυθμό Ανάπτυξης Αλυσίδας $g^p_\delta(\kappa, n, \rho, \Delta) = (1-\delta)\alpha$.*

Αρχίζουμε καλύπτοντας τα δύο μέρη του growth κατηγορήματος που που ορίστηκε στο [3].

**Lemma 0.3.2.** *(Συνεπές Μήκος) Αν στο view, ο $i$ είναι έντιμος στο γύρο $r$ και ο $j$ είναι έντιμος στο γύρο $r+t$, τότε $|\mathcal{C}_j^{r+t}(view)| \geq |\mathcal{C}_i^r(view)|$, για κάθε $t \geq 1$.*

Αυτή η ιδιότητα δεν επηρεάζεται από των αριθμό των σφραγίδων σε κάθε γύρο, επηρεάζεται όμως από την Καθυστέρηση καθώς θα πρέπει να περιμένουμε περισσότερους ($D$) γύρους για να είμαστε βέβαιοι πως οι αλυσίδα του ενός παίκτη έχει γίνει γνωστή και στον άλλο.

Ονομάζουμε $l^r$ την μακρύτερη αλυσίδα ενός έντιμου παίκτη στο γύρο $r$.

**Lemma 0.3.3.** *Για κάθε $r, t \geq 0$ και για κάθε $\delta > 0$,*

$$Pr\left[l^{r+t}(EXEC) < l^r(EXEC) + (1+\delta)\alpha t\right] < e^{-\Omega(\delta^2 \alpha t)}$$

Έδω αποκλίνουμε σημαντικά από την αρχική απόδειξη του [3]. Η Καθυστέρηση αναγκάζει τους συγγραφείς να ορίσουν ένα "Υβριδικό Πείραμα", μία εκτέλεση του πρωτοκόλλου στην οποία το μόνο που κάνει ο Αντίπαλος από ένα σημείο και μετά είναι να καθυστερεί τα μηνύματα των έντιμων παικτών για το μέγιστον δυνατό χρόνο ($D$). Μέσα από αυτή την απόδειξη η παράμετρος $\gamma = \frac{\alpha}{1+\Delta\alpha}$ δημιουργείται για να περιγράψει την ισχύ των έντιμων παικτών προσαρμοσμένη με βάση την Καθυστέρηση. Οι συγγραφείς αποδεικνύουν πως οι αλυσίδες των έντιμων παικτών στην πραγματική εκτέλεση είναι τουλάχιστον τόσο μεγάλες όσο αυτές των έντιμων παικτών στην υβριδική εκτέλεση. Έχοντας πετύχει αυτή τη σύνδεση μπορούν να μεταφέρουν τα όρια που απέδειξαν στην υβριδική εκτέλεση στην πραγματική. Οι αποδείξεις μας εδώ συγκλίνουν ξανά.

**Lemma 0.3.4.** *Για κάθε $r, t \geq 0$ και κάθε $\delta > 0$,*

$$Pr\left[\textit{min-chain-increase}_{r,t}(\textit{EXEC}) < (1-\delta)\alpha t\right] < e^{-\Omega(\delta^2 \alpha t)}$$

### 0.3.3 Ποιότητα Αλυσίδας

Αρχικά αποδεικνύουμε δύο άνω όρια για τον αριθμό από block που σφραγίζονται σε κάποια περίοδο γύρων.

**Lemma 0.3.5.** *(Άνω Όριο Blocks.) Έστω $Q_t(view)$ να είναι ο μέγιστος αριθμός από blocks σφραγισμένα σε κάποιο παράθυρο $t$ γύρων στο view. Για κάθε $t \geq 0$ και κάθε $\delta$,*

$$Pr\left[Q_t(\textit{EXEC}) > (1+\delta)(\alpha+\beta)t\right] < e^{-\Omega[\delta^2(\alpha+\beta)t]}$$

**Lemma 0.3.6.** *(Άνω Όριο Αντιπάλων Blocks.) Έστω $A_t(view)$ να είναι ο μέγιστος αριθμός από ανίπαλα blocks σφραγισμένα σε κάποιο παράθυρο $t$ γύρων στο view. Για κάθε $t \geq 0$ και $\delta > 0$,*

$$Pr\left[A_t(\textit{EXEC}) > (1+\delta)\beta t\right] < e^{-\Omega(\delta^2 \beta t)}$$

Συνδυάζοντας τα δύο άνω όρια αποδεικνύουμε την επιθυμητή ιδιότητα.

**Theorem 0.3.7.** *Για όλα τα $\delta > 0$, κάθε $p(\cdot)$, $\left(\Pi^p_{stamp}, \mathcal{C}^p_{stamp}\right)$ έχει (errorless) Ποιότητα Αλυσίδας $\mu = 1 - (1+\delta)\frac{\beta}{\alpha}$.*

Τόσο η ιδιότητα αυτό όσο και η απόδειξή της δε διαφέρουν σημαντικά από αυτή του [3]. Ο λόγος για αυτό είναι πως όλες οι διαφορές οι οποίες οφείλονται στην Καθυστέρηση βρίσκονται πλέον πίσω από το $\alpha$.

### 0.3.4 Συνέχεια

Η απόδειξη της Συνέχειας επίσης διαφέρει αρκετά από αυτή του [3]

Είτε ο Αντίπαλος έχει τη δυνατότητα παρακράτησης blocks είτε όχι, είναι χρήσιμο να ξέρουμε πως αντίπαλα blocks τα οποία δε τοποθετούνται κοντά στην κορυφή της αλυσίδας χάνουν την ισχύ τους και μάλλον δε θα βρεθούν στις αλυσίδες έντιμων παικτών.

**Lemma 0.3.8.** *Αν $\alpha \geq (1+\delta)\beta$ για κάποιο $0 < \delta < 1$ τότε, για κάθε σταθερά $0 < \omega < 1$ υπάρχει αμελητέα συνάρτηση $\epsilon(\cdot)$ τέτοια ώστε:*

$$Pr\left[\textit{view} \leftarrow \textit{EXEC} : \textit{withholding-time}(\textit{view}) \geq \omega t\right] \leq \epsilon(\beta t)$$

*Όπου withholding-time(view) είναι ο μεγαλύτερος αριθμός γύρων $t$ τέτοιος ώστε, στο view ο Αντίπαλος σφραγίζει ένα block $b$ στο γύρο $r$ και υπάρχει κάποιος έντιμος παίκτης $i$ τέτοιος ώστε το $b$ να εμφανίστηκε για πρώτη φορά στην αλυσίδα του $i$ στο γύρο $r + t$.*

**Απόκλιση:** Αρχικά σημειώνουμε τι σημαίνει για δύο παίκτες να αποκλίνουν σε ένα συγκεκριμένο γύρο. Δύο αλυσίδες $C_1$ και $C_2$ αποκλίνουν στο γύρο $r$ στο view αν το τελευταίο block που έχουν κοινό σφραγίστηκε πριν το γύρο $r$.

**Ευκαιρίες Σύγκλισης:** Next we need to see how these divergences are resolved.

**Definition 0.3.9.** (*Ευκαιρία Σύγκλισης*) Μία Ευκαιρία Σύγκλισης συμβαίνει κάθε φορά που οι έντιμοι παίκτες έχουν ευκαιρία να αποκτήσουν όλοι την ίδια αλυσίδα.

Στη δική μας περίπτωση μία τέτοια ευκαιρία συμβαίνει κάθε φορά που σφραγίζεται ένα έντιμο block. Στην περίπτωση της περιορισμένης καθυστέρησης παίρνει τη μορφή:

1. $\Delta$ **γύροι σιωπής.**
2. **Ένα μόνο νέο block εξορύσσεται.**
3. $\Delta$ **γύροι σιωπής.**

**Lemma 0.3.10.** *Έστω πως υπάρχει $0 < \lambda < 1$ τέτοιο ώστε $a \geq (1+\delta)\beta$. Εκτός με πιθανότητα $e^{-\Omega(\beta t)}$ στο view $\leftarrow$ EXEC, δεν υπάρχουν γύροι $r \geq r'$ και παίκτες $i, j$ τέτοιοι ώστε ο $i$ να είναι έντιμος στο γύρο $r$ και ο $j$ να είναι έντιμος στο γύρο $r'$ και $\mathcal{C}_i^r(\textit{view})$ και $\mathcal{C}_j^{r'}(\textit{view})$ αποκλίνουν στο γύρο $s = r - t$.*

**Theorem 0.3.11.** *Για κάθε $\lambda > 0$, κάθε $p(\cdot)$, $\left(\Pi_{stamp}^p, \mathcal{C}_{stamp}^p\right)$ ικανοποιεί (errorless) συνέχεια.*

# Κείμενο στα αγγλικά

# Chapter 1

# Introduction

## 1.1 Outline

Lets begin by providing a short plan for the chapters that will follow:

In this chapter we will focus on the very basic concepts of the blockchain. It will contain a summary of its history, structure and of course the goals and ideas that led to its creation in the first place.

In the second chapter we will go through some of the basic mathematical tools that are necessary to create and analyse the blockchain. We will also explain notation that will be useful to for the rest of the thesis.

Next we will dedicate one chapter to defining blockchain protocols in a more formal way. Moreover we will present an example proof that will play a significant part in proving the properties of our own model later on.

The third chapter consists of summaries of some main attempts at analysing the blockchain. We begin with the classic work of Garay, Kiagias and Leonardos: ”The Bitcoin Backbone Protocol: Analysis and Applications” [4]. This will be the introduction to some of the main properties that the rest of the summarized papers and this thesis is based on. We will also explain some of the changes this paper has undergone through the years and how they reflect the history of the blockchain. Next we will discuss the main inspiration of this thesis, the paper by Pass, Seeman and Shelat: ”Analysis of the Blockchain Protocol in Asynchronous Networks” [3]. We will go through its main concepts and the way it proves what it sets out to paying closer attention to the framework it creates as it is the one we are going to use in our proof in the last chapter. Lastly we briefly mention the work of Ling Ren: ”Analysis of Nakamoto Consensus” [6], which will also give us some insight into blockchain analysis and will complement the work of [3]. For each of these papers we will compare and explain the different ways in which they approach the analysis.

Next we will proceed to define a simplified blockchain protocol and then go through proving its properties using the methods of [3]. We will discuss how our choices and omissions regarding this protocol influence the course of the proof and the steps that separate it from the original work of Pass, Seeman and Shelat.

As a final chapter we will briefly discuss what this small journey through blockchain analysis has helped us learn and will mention some avenues for further exploration.

## 1.2 The Blockchain

### 1.2.1 History

There are two parts to the origin of the blockchain as a concept, a theoretical and a practical one. Let us start with the theoretical one, reaching consensus in a distributed setting with no trusted parties, also known as Byzantine Agreement ([7]–[9]). We assume that there is a set of players that have the capability of exchanging messages with each other. Each player $i$ holds some initial value $v_i$ and our goal is to ensure that the players can all end up with the same value. The problem arises when we consider that there might be a subset of players that have the adversarial goal of making sure the rest of the players don't reach a consensus on a single value.

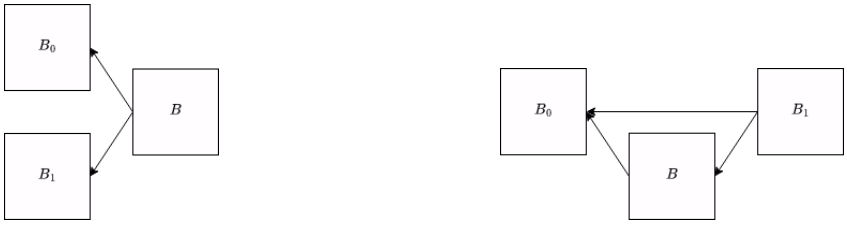
The practical origin of the blockchain is the attempt to create a decentralized ledger for financial transaction. This is what led to the creation of the first blockchain which became known as Bitcoin [1] and created a new field in cryptography and the realm of cryptocurrencies, both of which have gone through rapid evolution in the years previous to the writing of this thesis. Financial systems up to that point required some trusted third party (a bank) in order to ensure their proper operation.

This practical challenge is what sparked the publishing of [1] by Satoshi Nakamoto and the creation of Bitcoin. In it the author describes the basic premise of the blockchain, its structure and the reason it achieves its goal. From there, using the idea of decentralised financial systems as the main motive and appeal, the field grew rapidly and has been producing many examples both of theoretical [3], [4] and applied work [10]–[13] the latter of which is characterised by the great number of blockchain protocols that have been created ([1], [14]–[16]).

### 1.2.2 Structure

**Blocks:** The basic unit of the blockchain is, of course, the block. A block at its core is a collection of records, such are transactions. These are collected by the player that created the block simply through listening to the transactions that are broadcast to the network.

**Linking the Chain:** In order for a set of blocks to become a chain there needs to be something linking them. In the case of the blockchain, each block contains a summary (a hash as we will see later) of the previous block, this summary can be viewed as a pointer to the block that precedes it. In order for this linking to be unique we don't want different blocks to be able to have the same hashes. That way we can't have blocks that can be linked to more than one ancestor. Moreover we want to avoid being able to control what the hash of a block will be so that we can avoid blocks being inserted into the chain between two already existing blocks.




**(a)** Blocks $B_0$ and $B_1$ have the same hash so it is possible for a block $B$ to point to both of them.

**(b)** If a player is able to control the hash of $B$ they could insert it between $B_0$ and $B_1$.

**Figure 1.1:** Unwanted hash behaviors.

Achieving this unique linking is done by making the process of finding the correct link for the previous block computationally hard. This is the process of ”mining” which has the users wanting to publish a new block search through a huge amount of random numbers (and therefore use a huge amount of computational power) in order to find one that is valid for the blocks at hand. Mining can in this way also limit the rate at which blocks are produced bringing the growth of the chain to a manageable amount. We will go through the mechanism of hashing in the next chapter in order to understand how it functions in more detail.

**Forks:** Nothing stops multiple people from successfully mining blocks on the same round, these blocks moreover may have the same ancestor.

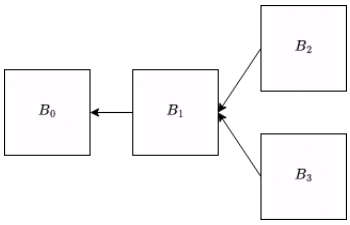
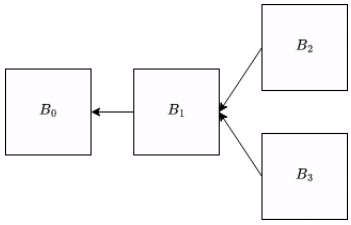
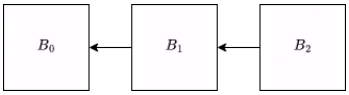



**Figure 1.2:** Two players aware of the chain $B_0, B_1$ both succeed in extending $B_1$.

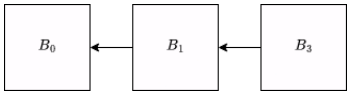
This leads to the creation of so called ”forks” in the chain, different paths one might go down from the genesis block to the most recent block of the chain. The chain has become more of a tree so how does a user decide which path in this tree is the one to be followed? These paths might well contain different or even contradictory transactions so this decision is crucial to the function of the blockchain. We could imagine a scenario where network errors or malicious actors could convince or trick players into each believing a different path of the chain to be the longest, breaking the core promise of the distributed ledger. This type of an attack can in fact be dangerously effective and requires dealing with [17], [18].




**(a)** The chain as it exists.




**(b)** The longest chain player 1 holds.




**(c)** The longest chain player 2 holds.

**Figure 1.3:** Players holding different longest chains.

Nakamoto tells us that each honest player should consider the longest available chain to be the valid one. As we will see in later chapters, this is easier said than done and requires a few compromises in order to make sure that every honest player refers to the same path when viewing the chain. In fact it requires an constraining the protocol with a separate security parameter dedicated to making sure that the players only consider chains that won't be overtaken.

As a last not regarding the complexity of forks as a concept it has to be said that completely eliminating them like some protocols attempt to do [16], [19] might not be a clear cut optimal solution [20]. There are protocols [21], [22] that forgo the chain structure for a seemingly more chaotic graph structure in order to achieve much greater speeds.

# Chapter 2

# Mathematical Background

## 2.1 Probabilistic Polynomial-Time Algorithms

For the sake of completeness, we need to include a few things about the complexity of the algorithms considered in cryptographic settings.

**Definition 2.1.1** (*Probabilistic Polynomial Time Algorithms* [2])**.** An algorithm $A$ is said to be probabilistic polynomial time (PPT) if:

1. There exists a polynomial $pol(\cdot)$ such that, for every input $x \in \{0,1\}^*$, the computation of $A(x)$ terminates within at most $pol(\|x\|)$ steps.
2. It has access to a source of randomness that yields random bits that are each independently equal to 1 with probability $p$ and 0 with probability $1-p$.

An probabilistic algorithm with $p = \frac{1}{2}$ is called uniform. Most of the time we will consider non-uniform algorithms for some $p \in (0,1)$.

This type of algorithms helps us quantify things like the strength of the Adversary in our protocols. At the core of a blockchain protocol is the probabilistic nature of block generation, this is the reason we use probabilistic algorithms. We will also consider our blockchain protocol execution to happen over a polynomial number of rounds and for this reason our algorithms need to be polynomial time.

## 2.2 Negligible Functions

By their construction, blockchain protocols work in a probabilistic fashion, as each user only has a chance of creating a valid block each round. This naturally introduces an amount of uncertainty to the process. How do we ensure then that chance doesn't happen to favor adversarial players for a period of time? To relieve our protocols of such fears we rely on negligible functions.

**Definition 2.2.1** (*Negligible Functions*)**.** A function $\epsilon(\cdot)$ is said to be negligible if for every polynomial $p(\cdot)$, there exists some $\kappa_0$ such that:

$$\epsilon(\kappa) \leq \frac{1}{p(\kappa)}$$

for all $\kappa \geq \kappa_0$.

**Definition 2.2.2** (*Strongly Negligible Functions*)**.** A function $\epsilon(\cdot)$ is said to be strongly negligible if there exist constants $c_0 > 0, c_1$ such that:

$$\epsilon(\kappa) \leq e^{-c_0\kappa+c_1}$$

for all $\kappa$.

These definitions are the ones used in [3] and therefore the ones most useful to our analysis, specifically the second one. As we have already mentioned we will mainly consider algorithms that run in polynomial time, with that we can understand that a strongly negligible function completely overshadows the power of such algorithms, making it ideal to guarantee security.

## 2.3 Hash Functions

In the introduction we spoke about including a summary of the previous block in each block, the mechanism that links what we call the blockchain. This mechanism should be able to get a large string (the size of a block), and return a much smaller string (one that can be included in a another block). This is something a hash function can achieve.

We will use [2] for our small exploration of hash functions.

**Definition 2.3.1.** (*Hash Function - syntax*) A hash function is a pair of probabilistic polynomial-time algorithms $(Gen, H)$ fulfilling the following:

- $Gen$ is a probabilistic algorithm which takes as input a security parameter $1^\kappa$ and outputs a key $s$. We assume that $1^\kappa$ is included in $s$ (though we will let this be implicit).

- There exists a polynomial $l$ such that $H$ is a (deterministic) polynomial-time algorithm that takes as input a key $s$ and any string $x \in \{0,1\}^*$, and outputs a string $H^s(x) \in \{0,1\}^{l(\kappa)}$.

If for every $\kappa$ and $s$, $H^s$ is defined only over inputs of length $l'(\kappa) > l(\kappa)$, then we say that $(Gen, H)$ is a fixed-length hash function with length parameter $l'$.

We have also mentioned how a block should only be able to be put in a single spot, there shouldn't be a way for a block to be "slid" between two blocks that already exist and there also shouldn't be a way for a block to be a continuation of more than one other blocks. For this we will briefly mention collision resistant hash functions.

**Algorithm 1:** Hash-coll$_{A,\Pi}(\kappa)$ [2]

$s \leftarrow Gen(1^\kappa)$
$(x, x') \leftarrow A(s)$
**if** $x \neq x'$ ;
& $H^s(x) = H^s(x')$ **then**
    **return** 1
    // The output of the experiment is 1 iff $x \neq x'$ and $H^s(x) = H^s(x')$.
    // Adversary $A$ has found a collision!
**else**
    **return** 0
    // No collision
**end**

**Definition 2.3.2.** (Collision resistant Hash Function) A hash function $\Pi = (Gen, H)$ is collision resistant if for all probabilistic polynomial-time adversaries $A$ there exists a negligible function $neg$ such that:

$$Pr\left[\text{Hash-coll}_{A,\Pi}(\kappa) = 1\right] \leq neg(\kappa)$$

## 2.4 Chernoff Bound

The probabilistic analysis of blockchain protocols requires the creation of some bounds that will limit the probability of some, usually unwanted, event from occurring. For this we will heavily utilise the multiplicative Chernoff Bound:

**Theorem 2.4.1** (*Multiplicative Chernoff Bound* [3])**.** *Let $X_1, ..., X_n$ be independent Boolean random variables, such that for all $i$, $Pr[X_i = 1] = p$. Furthermore, let $X$ be the sum of these variables, and $\mu = \mathcal{E}[X] = np$ be the expectation of the sum.*
*Then, for any $\delta \in (0, 1]$:*

$$Pr[X < (1 - \delta)\mu] < e^{-\Omega(\delta^2\mu)}$$
$$Pr[X > (1 + \delta)\mu] < e^{-\Omega(\delta^2\mu)}$$

We note here the right hand side of both of those inequalities does not need to specify the exact exponent only an $\Omega$, a lower bounding limit. This in turn creates an upper bound for the exponent and therefore the probability. By combining the bounds with the negligible functions mentioned above we can reach the following conclusion: The event $X$ will deviate from its expected value $\mu$ only with negligible probability, the higher the deviation $\delta$, the less likely it will happen and since the probability is negligible we can safely assume it will not occur.

## 2.5 Union Bound

Let us consider an event $A_i$ that can happen in a period $i$ of given a length (measured in time or blocks) in the blockchain. In our analysis, as in all blockchain analysis, we will often need

to bound the probability of such an event occurring in any period $i$. We are therefore forced to bound the union of all $A_i$ events occurring and for this purpose we will use a staple of probability theory, the Union Bound.

**Theorem 2.5.1** (*Union Bound*)**.** *For a finite set of events* $A_1, A_2, ..., A_n$*:*

$$Pr\left[\bigcup_{i=1}^{n} A_i\right] \leq \sum_{i=1}^{n} Pr[A_i]$$

# Chapter 3

# Construction

In the previous chapter we presented some basic mathematical tools that are necessary for the function and analysis of a blockchain protocol. In this chapter we will build upon these to explain the following:

- A basic framework that will allow us to talk more formally about blockchain protocols. We will also use this to bridge the gap between the different sources that will be presented in the next chapter.

- A kind of proof that is used extensively our main source and will therefore also be utilized by the analysis of our own model.

## 3.1 Framework

While we will present multiple papers, [3] is our main source. For this reason we will try to translate all other notation and variables into one most resembling its version to keep a sense of continuity.

**Security Parameters:** The first and most fundamental security parameter that is introduced is $\kappa$. This is the security parameter upon which our hash function depends and refers to length in bits.

This parameter is so fundamental in fact that there will be many cases where it will be omitted for convenience since it included in most functions and algorithms. In the next chapter we will also introduce the second security parameter $T$. In short, this parameter operates on a higher level than $\kappa$ in order to ensure the smooth function of the blockchain as a protocol instead of its basic cryptographic consistency. More on that when the need for a second parameter becomes unavoidable.

**Protocol:** Each player uses two basic algorithms $(\Pi, \mathcal{C})$.
Firstly $\Pi$ is the algorithm that keeps the $state$ the player is in. It uses the security parameter $\kappa$ and starts off with an input of $1^\kappa$ and an empty $state$.

Then comes $\mathcal{C}(\kappa, state)$, which translates the $state$ of the player into the actual chain they can consult. It takes $\kappa$ and the $state$ that is kept by $\Pi$ and outputs the records that are contained

within the chain in an ordered sequence. Obviously in most cases it is transactions and the ordered sequence that is produced by $\mathcal{C}$ is the ledger of all transactions that have happened in our system.

**Environment:** The environment that directs the protocol is denoted $Z(1^\kappa)$ and is a non-uniform probabilistic polynomial-time algorithm. It is what activates the $n$ players that take part in the protocol and denotes some of them as adversarial. The honest parties execute the protocol $\Pi$ while the adversarial parties are controlled by the Adversary $A$. The environment also serves as a blanket for the network the protocol uses, and delivers all messages the players send each other.

**Validity:** There needs to be a way to symbolize the fact that not all chains are in fact acceptable. We can't accept chains that contain, in a financial example, transactions that spend more money than a player has. The validity of the chain according to the rules that the protocol has set for its function is checked by the predicate $V(\cdot)$. For this reason a protocol $\Pi$ that uses a predicate $V$ for its validity will be symbolized with $\Pi^V$.

**Adversary:** Many things can go wrong during the execution of a blockchain protocol, so it should be noted that the Adversary doesn't just represent malicious actions. In order to achieve maximum security though when analysing a protocol we can assign everything that can go wrong to an adversarial party. From network delays, to players not operating their nodes correctly to multiple real malicious parties that are trying to cause harm to the protocol for some personal gain or other reason, all of these possibilities get wrapped under the blanket of the Adversary. Moreover, when there is delay or some other adversarial interfering with players' messages we assume that the Adversary is the one that delivers these messages to the players' receiving queue.

The Adversary is a non-uniform probabilistic polynomial-time algorithm and by communicating with the Environment can corrupt players turning them adversarial up to a certain limit. If we assume that our protocol can have up to $n$ players, the Adversary can have a maximum of $\rho n, \rho < \frac{1}{2}$ players under their control. All corrupted / adversarial parties are controlled by the Adversary and can therefore cooperate, share information and use their power collectively.

**Views:** As we discussed in the introduction even though our final goal is for all honest players to have the same image of the blockchain that might not be the case at all points in time. For this reason we need to establish a way to talk about what the chain looks like to each player at any point in time. First we define the view of the blockchain that a specific party $P$ has:

$$\text{view}_P^{(\Pi^V, \mathcal{C})}(A, Z, \kappa)$$

Next we will get the ensemble of these variables.
We also care about the longest chain a party has in its possession in a specific round, this will be written as:

$$\mathcal{C}_P^r(\text{view})$$

Next we have the joint view of the blockchain over all players:

$$\text{EXEC}^{(\Pi^V, \mathcal{C})}(A, Z, \kappa)$$

Again we will generally skip the many parameters and only use EXEC.

**Chain Notation** Now that we know how to specify for what player and round a chain appears at we can also show a way to talk about the blocks in that chain. For this we can use what is referred to as "python notation" because of its resemblance to the way Python indexes its lists, this has been used in various works such as [23] and [24]. To address a specific block in the chain $\mathcal{C}$ we will use the format $\mathcal{C}[\cdot]$. The first block of the chain is $\mathcal{C}[0]$ or $Gen$, the genesis block, while $\mathcal{C}[i], i > 0$ will refer to the $i^{th}$ block counting from the genesis block.
On the other hand, $\mathcal{C}[-i], i > 0$ will refer to the $i^{th}$ block counting from the tip of the chain (i.e. $\mathcal{C}[-1]$ is the last block of the chain).
Finally, $\mathcal{C}[i:j]$ will denote the sequence of blocks from $i$ (inclusive) to $j$ (exclusive). Skipping one of the two numbers will mean our sequence extends to the respective end of the chain.

## 3.2 Example Proof

Now that we have set up all of these tools let's see how they are used in our main source: [3] to prove properties about blockchain protocols.
We define a measure $A_t$ over any period of $t$ consecutive rounds in the blockchain with an expected value of:

$$\mathbb{E}[A_t] = \mu(t)$$

This method of proof works exactly the same for both versions of the Chernoff bound, for the sake of the example we will use the following:

$$Pr[A_t < (1-\delta)\mu(t)] < e^{-\Omega(\delta^2\mu(t))}$$

for any $\delta > 0$.
We have then a bound for the probability that his measure differs significantly from its expected value $\mu(t)$ in that period of $t$ rounds. Next we will bound the probability that this happens in ANY period of $t$ rounds during the execution of the protocol.

$$Pr\left[\bigcup_{i=1} Pr\left[A_t^i\right]\right] \leq \sum_{i=1} Pr\left[A_t^i\right] < poly(r) \cdot e^{-\Omega(\delta^2\mu(t))}$$

The number of rounds in the execution are bounded by $poly(\kappa)$, where $\kappa$ is a security parameter. Furthermore the entire exponent will be substituted by another security parameter, $T$ as such: $e^{-\Omega(T)}$. Lastly, [3] considers its results for $T > c\log(\kappa)$ for a large enough $c$. Putting all these together we can ignore the polynomial factor $poly(\kappa)$ as [3] does for all its proofs. We arrive at last at the following bound:

$$e^{-\Omega(T)}$$

This is a negligible function meaning we can safely ignore the probability of our measure differing significantly from its mean value during the execution of the blockchain.

# Chapter 4

# Analysis

Before we move to the examples of blockchain analysis we should discuss what we actually hope to achieve through it. As mentioned in the introduction blockchain protocols were first conceived as a means of creating a decentralized ledger for transactions. What would that require in order to work properly?

First of all we would like a guarantee that any transaction broadcast by a user will eventually be carried out. Therefore any such transaction must find its way into a block and finally into the blockchain itself. No financial system would be considered competent if there was any significant chance that a transaction will simply never be completed.

Furthermore, any transaction that is completed should also stay completed. It shouldn't be possible for a transaction that has been considered successful to be reverted or cancelled. If that were to happen all trust in such a system would evaporate.

As we have explained in the introduction though, the blockchain is a volatile construction and is really more of a tree, especially when considering the latest blocks added. With forks been a common occurrence how do we ensure that the above guarantees hold with overwhelming probability? To make this possible we have to make a compromise; it is not possible to expect that a block just appended to the longest chain will always be a part of it since another fork may overtake it temporarily or even permanently. For this reason, as promised in the previous chapter, we will create a depth parameter referred to as $T$ in order to follow the notation of our main source, [3]. With this our above desired properties become:

> **Liveness:** Any transaction that is broadcast by an honest player will reach a depth of at least $T$ in an honest player's blockchain.
>
> **Persistence:** Any transaction that has reached a depth of at least $T$ in the blockchain will always be part of every honest player's blockchain.

We want these properties to hold with overwhelming probability. Any block (and therefore also any transaction) that is buried at least $T$ deep in the chain can be referred to as "committed" as it has become a permanent part of the ledger. There now exist two security parameters we care about: $\kappa$ and $T$ with the former being the fundamental parameter that determine the cryptographic protocols the blockchain uses and the latter the way the players decide on the longest chains and blocks get permanently embedded in the chain.

## 4.1 The Backbone

The first stop of our exploration is [4], the paper by Garay, Kiagias and Leonardos. This paper has gone through many iterations, each adding and improving the previous one, we can briefly go through some important changes:

1. [25] is the original version.
2. [26] was created to integrate the concept of Typical Executions that was introduced by [5] while also addind bounded delay to the model after the release of [3].
3. [4] is the current latest version on which we will base the sections dedicated to this paper.

To begin with, the authors present the core assumption of the paper. An assumption such as this exists in all papers we will examine for a very simple reason: No matter how sophisticated a blockchain protocol network is, there will always be an amount of power that the adversarial parties should not possess because it will throw the whole system out of balance. this first assumption is now only based on the raw number of adversarial and honest parties, after the introduction of delay it will have to be adapted to deal with it.

**Definition 4.1.1.** (*Honest Majority Assumption*) $t$ out of $n$ parties are corrupted with

$$t \leq (1-\delta)(n-t),$$

where $3f + 3\epsilon < \delta \leq 1$.

Here we can present a categorization of rounds that is crucial to this analysis but is also present in the rest of the papers we will go through and is overall a useful way to distinguish between rounds.

- $X_i = 1$ if on round $i$ there was a block mined by an honest party.
- $Y_i = 1$ if on round $i$ there was exactly one block mined by an honest party.
- $Z_i = m$ if on round $i$ there were $m$ blocks mined by adversarial parties.

Moving on, this analysis is centered around the concept of a "Typical execution", this will use the core assumption about the power relation of the honest and adversarial parties to prove the building blocks with which the rest of the paper will be based on. This concept was first introduced in [5] and was then brought into the Backbone with the release of [26].

**Definition 4.1.2.** (*Typical Execution*) An execution is $(\lambda, \epsilon)$-typical for $\epsilon \in (0,1)$ and integer $\lambda \geq 2/f$, if, for any set $S$ of at least $\lambda$ consecutive rounds, it holds:

(a) $(1-\epsilon)\mathbb{E}[X(S)] < X(S) < (1+\epsilon)\mathbb{E}[X(S)]$ and $(1-\epsilon)\mathbb{E}[Y(S)] < Y(S)$.

(b) $Z(S) < \mathbb{E}[Z(S)] + \epsilon\mathbb{E}[X(S)]$.

(c) No insertions, copies or predictions.

Where $X(S), Y(S), Z(S)$ are sums of the corresponding random variables over the period of rounds $S$.

We can intuitively explain what the notion of the typical execution entails:

Its last part suggests that our assumptions about hash functions should hold. Furthermore, both the adversarial and honest parties should mine blocks at a rate that corresponds to their power. We especially care about the number of unique successful rounds been as expected as these are the rounds that are the most crucial for the proper function of the blockchain.

Having defined what an execution of the protocol should look like, the authors prove that this is true with the overwhelmingly high probability of:

$$1 - e^{-\Omega(\epsilon^2 \lambda f + \kappa - \log(L))}$$

As we have explained there are two main properties we would like for a blockchain protocol to have in order to be effective: Liveness and Persistence. These are not always easy to prove directly, the authors of these paper first begin by proving three other, extremely useful properties that can then be used to form our original requirements as well as other properties that we might desire. For typical executions and therefore with overwhelming probability, the following properties are then proven:

**Chain Growth:** For any honest party $P$ that has chain $\mathcal{C}$ in view, it holds that after any $s$ consecutive rounds it adopts a chain that is at least $\tau \cdot s$ blocks longer than $\mathcal{C}$ with parameters:

$$\tau = (1 - \epsilon) f$$
$$s \geq \lambda$$

For a blockchain to function it should be alive, it should be able to grow and increase in size continually. To make sure of this we have to find a bound for the number of blocks that will be produced in any period of rounds.

**Chain Quality:** For any honest party $P$ with chain $\mathcal{C}$ in view, it holds that for any $l$ consecutive blocks of $\mathcal{C}$ the ratio of honest blocks is at least $\mu$ with parameters:

$$l \geq 2\lambda f$$
$$\mu = 1 - \left(1 + \frac{\delta}{2}\right) \cdot \frac{t}{n-t} - \frac{\epsilon}{1-\epsilon} > 1 - \left(1 + \frac{\delta}{2}\right) \cdot \frac{t}{n-t} - \frac{\delta}{2}.$$

By the very definition of the adversary it is understood that we would desire them to have as little power and influence over the chain as possible. Therefore in any series of blocks added to the chain the fraction of blocks that were contributed by the adversary should be bounded by a, preferably low and surely known measure.

**Common Prefix:** For any pair of honest players $P_1$ and $P_2$ with chains $\mathcal{C}_1$ and $\mathcal{C}_2$ at rounds $r_1 \geq r_2$ in view it holds that $\mathcal{C}_1[-T:] \preceq \mathcal{C}_2$ with parameter:

$$T \geq 2\lambda f$$

This is perhaps the most crucial property of a blockchain and the one most dependant on adversarial power and action as we will see in [3]. Each party taking part in the execution of the protocol needs to have a common view of the chain. This can be intuited quite easily as these protocols are supposed to be public ledgers that record various events such as transactions and should therefore have only one interpretation. We therefore desire that blocks that are at least $T$ deep in the chain will never be discarded through a fork and are therefore "carved" into this public ledger forever for all parties to see.

## 4.2 Introduction to Delays

Next we will visit the paper that provides the main basis for this thesis, [3] by Pass, Seeman and Shelat.

The analysis is parameterized using two constants representing the powers of the honest and adversarial parties:

$\alpha \rightarrow$ The probability that at least one honest party mines a block in a round.

$\beta \rightarrow$ The expected number of adversarial blocks that are mined each round.

The difference in the two measures is that the adversarial parties can cooperate with each other and can therefore use their entire power at once, while the honest players can work separately and might contribute to the blockchain in different directions, creating forks.

In addition to that, there needs to be a measure for the mining power of the honest parties that has been adjusted for delay:

$$\gamma = \frac{\alpha}{1 + \Delta\alpha}$$

This in fact arises naturally during the Chain Growth proof and is then used in the rest of the paper.

To begin with authors define two different environments in which they will prove their desired properties.

- The first one is $\Gamma_0$ which has no restrictions apart from validity.
  This means that any property proven in these environment is independent even of the adversary. As we will see the Chain Quality and Chain Growth properties are proven in this environment and are therefore true regardless of adversarial power.

- In the second environment, $\Gamma^p_\lambda$ the adversary is actually taken into account with their power being subject to:

$$\alpha(1 - 2(\Delta + 1)\alpha) \geq \lambda\beta$$

This is the equivalent of the Honest Majority Assumption in a bounded delay setting defined in [4]. A major difference here is that this assumption is not considered for the entire proof but rather only a part of it. In particular the only part of the analysis that is dependant on the adversary's power in this way is the Consistency property.

We move now to examine the properties the authors prove for their model. These are in essence the same as the ones introduced by [4] but do contain some changes.

For each of the properties the authors want to prove, they define a predicate that is true if and only if that property holds for the chain. Then they try to prove that these predicates hold with overwhelming probability.

### 4.2.1 Chain Growth

This first predicate is also the most complicated one in its definition and therefore the best example for the whole process. To begin with, the authors define way to measure the increase of the chain between rounds.

$$\text{min-chain-increase}_{r,t}(\text{view}) = \min_{i,j} \left\{ |C_j^{r+t}(\text{view})| - |C_i^r(\text{view})| \right\}$$

Next we arrive at the actual predicate that checks the property of Chain Growth. The authors explain that we want this property to have to characteristics:

- Players have similar chain length so that no player is left behind.
- The chains increase continually by at least some known amount.

The predicate $\text{growth}^t(\text{view}, \Delta, T)$ combines these desired characteristics,
$\text{growth}^t(\text{view}, \Delta, T) = 1$ iff:

- **(consistent length)** for all rounds $r \geq |\text{view}| - \Delta, r' \leq |\text{view}|, r' \leq r + \Delta$, for every two players $i, j$ such that in view $i$ is honest at round $r$ and $j$ is honest at round $r'$, we have that:

$$|\mathcal{C}_j^{r'}(\text{view})| \geq |\mathcal{C}_i^r(\text{view})|$$

- **(chain growth)** for every round $t \leq |\text{view}| - r$, we have:

$$\text{min-chain-increase}_{r,t}(\text{view}) \geq T$$

The two parts of the predicate correspond to the two requirements we expressed for our model, with the first one in particular allowing players a period $\Delta$ (equal to the delay of the system) to synchronize their chain lengths.
With this predicate the framework defines the Chain Growth property for a generalized blockchain protocol.

**Definition 4.2.1.** A blockchain protocol $(\Pi, \mathcal{C})$ has chain growth rate $g(\cdot,\cdot,\cdot,\cdot)$ in $\Gamma$-environments if for all $\Gamma$-admissible $(n(\cdot), \rho, \Delta(\cdot), A, Z)$, there exists some constant $c$ and negligible functions $\epsilon_1$, $\epsilon_2$ such that for every $\kappa \in \mathbb{N}, T \geq c\log(\kappa)$, and $t \geq \frac{T}{g(n(\kappa),\rho,\Delta(\kappa))}$, the following holds:

$$Pr\left[\text{view} \leftarrow \text{EXEC}^{(\Pi^V,\mathcal{C})}(A, Z, \kappa) : \text{growth}^t(\text{view}, \Delta(\kappa), T) = 1\right] \geq 1 - \epsilon_1(\kappa) - \epsilon_2(T)$$

Additionally, if $\epsilon_1 = 0$ we say that $(\Pi, \mathcal{C})$ has errorless chain growth $g$ in $\Gamma$ environments.

**Theorem 4.2.2.** *For any $\delta > 0$, any $p(\cdot)$, $\left(\Pi^p_{Nak}, \Pi^p_{Nak}\right)$ has chain growth rate*

$$g^p_\delta(\kappa, n, \rho, \Delta) = (1-\delta)\gamma$$

*in $\Gamma_0$ environments.*

### 4.2.2 Chain Quality

**Definition 4.2.3.** $quality^T(\text{view}, \mu) = 1$ iff for every round $r$ and every player $i$ that is honest in view$^r$, for any consecutive sequence of $T$ blocks in $C(\text{view})^r_i$, the fraction of blocks $m$ that are honest w.r.t. view$^r$, is at least $\mu$.

**Definition 4.2.4.** A blockchain protocol ($\Pi$, $\mathcal{C}$) has chain quality $\mu(\cdot,\cdot,\cdot,\cdot)$ in $\Gamma$, if for all $\Gamma$-admissible ($n(\cdot)$, $\rho$, $\Delta$, $A$, $Z$), there exists some constant $c$ and negligible functions $\epsilon_1$, $\epsilon_2$ such that for every $\kappa \in \mathbb{N}, T > c\log(\kappa)$ the following holds:

$$Pr\left[\text{view} \leftarrow \text{EXEC}^{(\Pi^V,\mathcal{C})}(A, Z, \kappa) : quality^T(\text{view}, \mu(\kappa, n(\kappa), \rho, \Delta(\kappa))) = 1\right] \geq 1 - \epsilon_1(\kappa) - \epsilon_2(T)$$

Additionally, if $\epsilon_1 = 0$ we say that $(\Pi, \mathcal{C})$ has errorless chain quality $\mu$ in $\Gamma$ environments.

**Theorem 4.2.5.** *For all $\delta > 0$, any $p(\cdot)$, $\left(\Pi^p_{Nak}, \Pi^p_{Nak}\right)$ has chain quality*

$$\mu^p_\delta(\kappa, n, \rho, \Delta) = 1 - (1+\delta)\frac{\beta}{\gamma}$$

*in $\Gamma_0$ environments.*

### 4.2.3 Consistency

The Consistency property also has an interesting way of dealing with delay. The authors take it into account by defining "Convergence Opportunities", moments in time that the honest player have a chance to sync with each other and all obtain the same longest chain. We will see how these work in our own proof but we can briefly mention that a Convergence Opportunity is a uniquely successful honest round preceded and followed by at least $\Delta$ rounds of silence.

**Definition 4.2.6.** Let $consistent^T(\text{view}) = 1$ iff for all rounds $r_1 \leq r_2$ and all players $i, j$ such that $i$ is honest at view$^{r_1}$ and $j$ is honest at view$^{r_2}$, we have that $C_i^{r_1}[-T] \leq C_j^{r_2}$

**Definition 4.2.7.** A blockchain protocol $(\Pi, \mathcal{C})$ satisfies consistency in $\Gamma$ environments, if for all $\Gamma$-admissible $(n(\cdot), \rho, A, Z)$, there exist some constant $c$ and negligible functions $\epsilon_1, \epsilon_2$ such that for every $\kappa \in \mathbb{N}, T > c\log(\kappa)$, the following holds:

$$Pr\left[\text{view} \leftarrow \text{EXEC}^{(\Pi^V, \mathcal{C})}(A, Z, \kappa) : consistent^T(\text{view}) = 1\right] \geq 1 - \epsilon_1(\kappa) - \epsilon_2(T)$$

Additionally, if $\epsilon_1 = 0$, we say that $(\Pi, \mathcal{C})$ has errorless consistency in $\Gamma$-environments.

**Theorem 4.2.8.** *For any $\lambda > 1$, any $p(\cdot)$, $\left(\Pi^p_{Nak}, \Pi^p_{Nak}\right)$ satisfies consistency in $\Gamma^p_\lambda$ environments.*

As an aside, the authors also prove an upper bound for the Chain Growth Property. They argue this can be a useful measure to further understand the blockchain but don't use it for the rest of their analysis and for this reason we will omit it for now.

### 4.2.4 Liveness and Persistence

To end the analysis, [3] turns to proving how a blockchain that has the properties they just defined and proved for the Nakamoto protocol can be used to create a public ledger. That is they prove the properties of Liveness and Persistence just like [4] did before them. They define them in a way that follows their own notation:
**Liveness** Let $live(\text{view}, t) = 1$ iff for any $t$ concecutive rounds $r, ..., r + t$ in view there exists some round $r' \in [r, r + t]$ and player $i$ such that in view:

1. $i$ is honest at round $r'$,
2. $i$ received a message $m$ as input $r'$ and
3. for every player $j$ that is honest at $r + t$ in view, $m \in \mathcal{L}_j^{r+t}(\text{view})$.

**Definition 4.2.9** (*Liveness*)**.** We say that a public ledger $(\Pi, \mathcal{L})$ if live with wait-time $w$ in $\Gamma$ environments if for all $\Gamma$-admissible $(n(\cdot), \rho, \Delta(\cdot), A, Z)$, there exists a negligible function $\epsilon$ in the security parameter $\kappa \in \mathbb{N}$, such that:

$$Pr\left[\text{view} \leftarrow \text{EXEC}^{(\Pi,\mathcal{L})}(A, Z, \kappa) : live\left(\text{view}w\left(\kappa, n(\kappa), \rho, \Delta(\kappa)\right)\right) = 1\right] \geq 1 - \epsilon(\kappa)$$

**Persistence** Let $persist_\Delta(\text{view}) = 1$ iff for every round $r \leq |\text{view}| - \Delta$, every player $i$ that is honest at view$^r$ and every position $pos \leq |\mathcal{L}_{\rangle}^{\nabla}(\text{view})|$, if if $\mathcal{L}_i^r(\text{view})$ contains the message $m$ at $pos$, then for every round $r'$ such that $r + \Delta \leq r'$ and every honest player $j$ we have that $m$ is also at $pos$ in $\mathcal{L}_j^{r'}(\text{view})$.

**Definition 4.2.10** (*Persistence*)**.** We say that a public ledger $(\Pi, \mathcal{L})$ is persistent in $\Gamma$ environments if for all $\Gamma$-admissible $(n(\cdot), \rho, \Delta(\cdot), A, Z)$, there exists a negligible function $\epsilon$ in the security parameter $\kappa \in (N)$ such that:

$$Pr\left[\text{view} \leftarrow \text{EXEC}^{(\Pi,\mathcal{L})}(A, Z, \kappa) : persist_{\Delta(\kappa)}(\text{view}) = 1\right] \geq 1 - \epsilon(\kappa)$$

## 4.3 The Backbone with delay

Now that we have seen how [3] introduced the concept of bounded delay let's see how [5] brought it into their own way of analysis.

Like [3], the addition of delay necessitates the re-balancing of adversarial to honest power ratio given this new parameter. For this purpose the authors alter their Honest Majority Assumption:

**Definition 4.3.1.** (*Honest Majority Assumption - Bounded Delay*) $t$ out of $n$ parties are corrupted with

$$t \leq (1-\delta)(n-t),$$

where $\epsilon + 2\Delta f + \frac{2\Delta}{\lambda} \leq \frac{\delta}{2}$.

Now that delay has been introduced to the system, there need to be some extra types of rounds taken into account. For this, new random variables similar to the ones mentioned above are defined:

- $X'_i = 1$ if on round $i$ there was an honest block mined and there were no other honest blocks mined up to $\Delta$ rounds before $i$. We call this a $\Delta$-isolated uniquely successful round.
- $Y'_i = 1$ if on round $i$ there was exactly one honest block mined and there is no other honest block mined within $\Delta$ rounds. We call that a isolated successful round.

So, when delay is introduced we start caring not only for the honest blocks mined and whether or not there was only one mined in each given round but for whether or not they were alone in a period of rounds equal to the delay. That is because, any honest blocks mined within $\Delta$ rounds of each other still have the potential to create a fork in the chain as the miner of the second block mined might not be aware of the first one due to the delay.

**Definition 4.3.2.** (*Typical Execution - Bounded Delay*)

(a) $(1-\epsilon)\mathbb{E}[X'(S)] < X'(S) < (1+\epsilon)\mathbb{E}[X(S)]$ and $(1-\epsilon)\mathbb{E}[Y'(S)] < Y'(S)$.

(b) $Z(S) < \mathbb{E}[Z(S)] + \epsilon\mathbb{E}[X'(S)]$.

(c) No insertions, copies or predictions.

This is in essence the same as the previous definition with a few subtle differences. In this version we value isolated [uniquely] successful rounds instead of simple [uniquely] successful rounds. As we explained before this is because the introduction of delay hinders the cooperation of honest players.

We also can begin to see a pattern emerging, both [4] and [3] have turned their attention to blocks that have a safe buffer of silence around them the former using isolated rounds and the latter using convergence opportunities.

Having altered the main assumption to accommodate delay, the properties mentioned above are proven again although with delay dependent parameters:

**Chain Growth** holds with parameters:

$$\tau = (1-\epsilon)f(1-f)^{\Delta-1}$$
$$s \geq \lambda$$

**Chain Quality** holds with parameter:

$$l \geq 2\lambda f + 2\Delta f$$
$$\mu = 1 - \frac{1}{(1-\epsilon)(1-f)^{\Delta}} \cdot \frac{t}{n-t} - \frac{\epsilon}{1-\epsilon}\left(1 + \frac{\Delta}{\lambda}\right)$$

**Common Prefix** holds with parameter:

$$T \geq 2\lambda f + 2\Delta$$

The properties remain the same in essence but have been adjusted for the delay that was introduced.

## 4.4 Further Exploration

To conclude our journey through different approaches of analysis blockchains we will briefly talk about Ling Ren's [6].

In contrast to the two papers we went through before, the author, Ling Ren does not really bother with proving the three properties but instead goes straight for the ones that are directly essential to the function of the blockchain as a transaction ledger.
They are concerned with:

1. **Liveness:** Every transaction is eventually committed by all honest players. This definition is the same as the one we have already gone over.

2. **Safety:** Honest players don't commit different blocks at the same height. Here is a different expression of the property of Persistence. Persistence essentially tells us that all honest players should have the same blocks at the same positions (except maybe for the last $T$ blocks). Therefore them having different blocks at the same position (height) would be it's negation.

Like the previous paper [6] denotes the mining rates of the honest and adversarial players as $\alpha$ and $\beta$ respectively.

**Definition 4.4.1.** Let $g = e^{-\alpha\Delta}$. Let $\delta$ be any positive constant. Nakamoto consensus with the $T$-confirmation commit rule guarantees safety and liveness except for $e^{-\Omega(\delta^2 g^2 T)}$ probability if

$$g^2\alpha > (1+\delta)\beta$$

Ling Ren's approach of taking delay into account is most similar to that of [3]. They also care about the moments in time when the honest players can sync their chains. For this reason they define different categories of blocks.

> **Non-tailgater:** An honest block with no other honest block mined in the previous $\Delta$ rounds.
>
> **Loner:** An honest block with no other honest blocks mined in the previous or next $\Delta$ rounds.

This paper continues the pattern of establishing ways to talk about different kinds of blocks and specifically isolated ones. We can recall the Convergence Opportunities mentioned in [3] and notice that the Loner blocks defined here are exactly the same. This paper builds its construction on this and the additional category of Non-tailgater blocks. In fact, as the author points out, the bound for Convergence Opportunities in this paper is better than the one in [3].

With these tools Ren proves the properties:

**Theorem 4.4.2** (*Liveness*)**.** *Suppose $g\alpha > (1+\delta)\beta$. At time $t$, except with probability $e^{-\Omega(\delta^2 g\alpha t)}$, every honest player commits at least $\frac{\delta}{6}g\alpha t - T - 1$ honest blocks.*

**Theorem 4.4.3** (*Safety*)**.** *Suppose $g\alpha > (1+\delta)\beta$. Consider any time $t$ and any block $B$ that is considered committed by some honest player at time $t$. Except with probability $e^{-\Omega(\delta^2 g^2 \alpha t)}$, for all time $t' \geq t$, no honest player commits a block $B' \neq B$ at the height of $B$.*

# Chapter 5

# The Stamp Model

## 5.1 Definition

We have now seen examples of blockchain analysis, each dealing with a fully realized protocol. Let us explore what this analysis contains in greater detail by creating our own, much simplified blockchain protocol. This ”Stamp Model” is what will be analyzed in the rest of this thesis.

**Blocks:** Much like in the original case, a block is just a collection of records that is broadcast by a player to the whole network. The goal is of course for this block to be included in the chain in a permanent way. Each block contains the following:

- A Stamp (for validation)
- The transactions published through it
- The Stamp of the previous block (for ordering)

Honest players will publish blocks as a continuation of their longest chain, by including the Stamp of its latest block. One the other hand, the Adversary can publish their blocks as continuations of any already published block by including the corresponding Stamp, creating forks.

**Stamps:** For a player to publish a block they normally need to solve a computational puzzle through the process called mining. To abstract this we will define Stamps. A Stamp is a token that a player can obtain which allows them to publish a single block in the round that they received the Stamp. That block will then be taken into account by the rest of the players because it has a Stamp that was given out the round it was published in. The Stamps have the following properties:

- Only the Oracle is able to create Stamps
- Each Stamp is unique

Because they are unique, there can be no insertions in the chain.

**Oracle:** The Stamp Oracle is the mechanism that gives out the Stamps each round and serves as the abstraction for the hashing puzzle in a normal mining process. The mining process, as

we discussed before, is the random process through which players compete with each other in order to discover the hash that will allow them to create a new block. By creating the Oracle we simplify and abstract this process, hiding the complex mathematical mechanisms (hashing) and the competition of computing power (mining) without losing their core purpose of allowing a random player to publish a block. We will call this the $\mathcal{F}_{Stamp}$ functionality and will assume that each player can petition it once per round in order to obtain a Stamp for the round. For now we will assume that the Oracle only hands out at most one Stamp in each round, this leaves exactly three possible scenarios for each round:

1. The Adversary receives the Stamp.
2. Exactly one honest player receives the Stamp.
3. Nobody receives the Stamp.

This approach removes the possibility of multiple honest players creating a block on the same turn or the Adversary creating a block on the same turn with some honest players. We are then left with three very distinct scenarios that can take place. If we turn back to the analysis in [4] that would mean that the $X$ and $Y$ random variables are now the exact same for any round.

Let us explain the experiment that simulates this process: In each round, the Oracle flips a coin for each player that takes part in our protocol. These flips come up heads (the event $H$ happens) with probability $p$. Then, after all the flips the Oracle goes through the results and decides who to hand out the Stamp to in the following way:

1. If any adversarial player has gotten the event $H$, the Adversary gets the Stamp.
2. Else if exactly one honest player has gotten the event $H$ they receive the Stamp.
3. Else nobody receives the Stamp.

In our simulation we can clearly see that the probability $p$ is the equivalent of the success probability of a hashing query in the mining process. Our blockchain protocol will therefore be denoted:

$$\left(\Pi^{p}_{Stamp}, \mathcal{C}^{p}_{Stamp}\right)$$

Our three scenarios mean that anytime more than two honest players get an $H$ as their result, these results cancel each other out. Moreover, anytime the Adversary gets an $H$ they overpower any honest player that also gets one.

**Stamping Power:** Next we want to see the power that this way of handing out Stamps gives the participants. For this we will follow the initial steps of [3] and define the following probabilistic quantities, assuming that there are $n$ total players and at most $\rho n$ players controlled by the Adversary.

The probability that an honest player gets the Stamp:

$$\alpha = Pr[HonestStamp] = (1-\rho)np(1-p)^{n-1}$$

The probability that an adversarial player gets the Stamp:

$$\beta = Pr[AdversarialStamp] = 1 - Pr[No\ adversarial\ player\ gets\ an\ H] = 1 - (1-p)^{\rho n}$$

For $p$ small enough compared to $\frac{1}{n}$, $\beta$ can be approximated by the easier to use:

$$\beta \approx \rho np$$

## 5.2 Variations

Since we have simplified our model down to a bare minimum there are a lot of changes and alterations we can make to it. We will briefly go through them here and then see what some of them actually change in the course of the proofs.

**Multiple Stamps:** One of the things that stand out in the above definition as a significant simplification is the fact that at most one Stamp is ever given out at each round. One of the alternative scenarios that will be featured in the following sections is the one where there is no limitation to the number of Stamp each round can wield.

To achieve that with our imagined experiment we only have to omit the final selection process. This way any player who succeeded in their coin flip will be awarded a Stamp for that round. We still assume these Stamps can only be used for the round they were awarded. In this version the quantities defined above change as such:

$$\alpha' = Pr[HonestStamp] = 1 - (1-p)^{(1-\rho)n}$$

$$\beta' = E[Adversarial\ Stamps\ in\ a\ round] = \rho np$$

Similar to what we did with $\beta$ in the original iteration of the model, for a $p$ small enough compared to $\frac{1}{n}$, $\alpha'$ can be approximated by the easier to use:

$$\alpha' \approx (1-\rho)np$$

Two main differences can be noted here:

1. While $\beta$ denotes the probability of the Adversary gaining the unique Stamp at a particular round, $\beta'$ is the expected number of Stamps that the adversarial players will gain in total in a single round. The reason for this change in perspective is that because the Adversary controls all its players it can cooperate and effectively utilize all of its gained Stamps in a single round.

2. In contrast to the Adversary, honest players gaining multiple stamps in a single round doesn't contribute to their power since they cannot cooperate the same way as the players controlled by the Adversary. For this reason both $\alpha$ and $\alpha'$ denote the probability of an honest player gaining a stamp although in the first case that stamp will be unique while in the second it might be one among many honest and / or adversarial stamps.

Despite these differences, $\alpha'$ and $\beta'$ will still play the same roles as their previous counterparts.

**Stamp Withholding:** As we have explained above any player, including ones controlled by the Adversary has an obligation to publish a stamped block any round they receive the Stamp. The alternative is of course to give the ability to the adversarial players (the honest ones have no use for this) to keep the Stamps they get to themselves until they decide to publish the blocks. This means that the Adversary could prepare an entire chain in secret and publish it all at once. Obviously we would have to restrict the Adversary in only connecting the blocks they have stamped and kept private to block in the chain that existed when they got that Stamp.

**Delay:** One of the main features of [3] is the concept of Adversary induced Delay. The Adversary in that setting has the power to delay any message sent by any users from being received by any other user by up to $\Delta$ rounds. It is interesting to note that this measure encompasses system induced delays too, that is delays caused by the network the model would be built on. Attributing all delay to the Adversary simplifies the model and creates a stronger Adversary and therefore stronger security results. As we will discuss in the following sections this delay significantly effects the analysis, requiring in multiple cases large portions of proofs dedicated to taking it into account while also introducing a new measure $\gamma$ that more accurately represents the power of the honest players.

## 5.3 Chain Growth

**Theorem 5.3.1.** *For any $\delta > 0$, $\left(\Pi^p_{Stamp}, \mathcal{C}^p_{Stamp}\right)$ has (errorless) chain growth rate $g^p_\delta(\kappa, n, \rho, \Delta) = (1-\delta)\alpha$.*

We begin covering the two parts of the growth predicate (defined by [3]), beginning with the first; we should make sure our protocol doesn't allow players to drift apart in chain length. In our case this is trivially easy due to the simplifications made in the definition.

**Lemma 5.3.2.** *(Consistent Length) If in view, $i$ is honest at round $r$ and $j$ is honest at round $r+t$, then $|\mathcal{C}_j^{r+t}(view)| \geq |\mathcal{C}_i^r(view)|$, for any $t \geq 1$.*

*Proof.* As we have explained in our model definition, the messages sent by players are delivered within one round, sidestepping the need to take delay into account.
Therefore, since $i$ is honest at $r$ they will broadcast (or have already broadcast in the past) their chain of length $|\mathcal{C}_i^r(\text{view})|$. Since this chain will be delivered at the very next round $r+1$ from that round on any player $j$ that is honest at round $r+t$ will have a chain of length $|\mathcal{C}_j^{r+t}(\text{view})|$ that is at least equal to $|\mathcal{C}_i^r(\text{view})|$. □

This first property is not affected by the amount of stamps that are given out during a single round. It is though affected by the introduction of delay. In that case, the honest player $j$ at round $r+t$ is only guaranteed to have a chain as long as that of $i$ at round $r$ for $t \geq \Delta$. This is because only then can we be sure that all messages from round $r$ have reached $j$ and $j$ has the chain $i$ had at round $r$.

To continue with the second property (bounded minimum chain increase) it is first useful to show a bound for the increase of the longest chain. We will denote the longest chain of an honest player at round $r$ as $l^r$

**Lemma 5.3.3.** *For any $r, t \geq 0$ and for any $\delta > 0$,*

$$Pr\left[l^{r+t}(EXEC) < l^r(EXEC) + (1+\delta)\alpha t\right] < e^{-\Omega(\delta^2 \alpha t)}$$

*Proof.* Fix some $r$ and $t$. Each time an honest player stamps a block they increase the length of the honest chain by one. Therefore, in $t$ rounds, the longest chain $l^r$ will be increased by at least as many honest blocks were mined in the period $[r, r+t]$.
Since the probability of an honest block being stamped in any given round is $\alpha$ we get, for the number of honest blocks stamped in a period of $t$ rounds ($X_t$):

$$E\left[X_t\right] = \alpha t$$

By using 3.2 we can arrive at the desired result. □

Here we have already diverged from the original proof of [3]. The reason for this is again delay, in a much more major way this time. When a player in a system with delay gains a Stamp, they cannot be sure that the block they publish will actually contribute to the increase of the chain's length. This is because there might be blocks already added to the same spot they intend to add theirs that simply have not reached them due to the delay. Through this issue the parameter $\gamma$ is created to more accurately represent the honest player power in a bounded delay setting.

**Hybrid Experiment:** Since the Adversary can choose to delay different messages for different amounts in an arbitrary way the authors of [3] have to employ a very interesting method to navigate delay. They define a "Hybrid Experiment", and execution of the protocol that diverges from the "Real" by having its Adversary replaced by a much simpler, predictable one. A new Adversary that just delays all messages as much as possible ($\Delta$ rounds) and makes sure no honest player can mine while a message is being delayed. This achieves a predictable behavior that can be bounded more easily and for this hybrid experiment the authors prove the above bound but for $\gamma = \frac{\alpha}{1+\Delta\alpha}$ instead of $\alpha$.
Even with that they still need a way to transfer the bound over to the Real execution. For this they successfully prove that any honest player's chain in the Real experiment is at least as large as the chain of the same player in the Hybrid experiment at the same point in time. They achieve it by using the fact that the two executions differ only in their Adversary and that everything else happens in the exact same way. After this they transfer the bound they have proven for the Hybrid experiment to the Real execution through simply applying the relationship shown for the two.
Our proofs now converge again, having proven the bound for the increase of the longest chain in a period of rounds, into showing a bound for the minimum chain increase.

**Lemma 5.3.4.** *For any $r, t \geq 0$ and any $\delta > 0$,*

$$Pr\left[\textit{min-chain-increase}_{r,t}(EXEC) < (1-\delta)\alpha t\right] < e^{-\Omega(\delta^2 \alpha t)}$$

*Proof.* We begin by observing that the minimum chain increase between two rounds is going to be the difference between the shortest chain of the later rounds and the longest chain of the earlier round:

$$\text{min-chain-increase} = \min_{i,j}\left\{|C_j^{r+t}(\text{view})| - |C_i^r(\text{view})\right\} = \min_j |C_j^{r+t}(\text{view})| - \max_i |C_i^r + (\text{view})|$$

Because of the Consistent Length property we proved earlier we have:

$$\min_j |C_j^{r+t}(\text{view})| \geq \max_j |C_j^{r+t-1} + (\text{view})|$$

Therefore:

$$\text{min-chain-increase} \geq \max_j |C_j^{r+t-1}(\text{view})| - \max_i |C_i^r + (\text{view})| = l^{r+t-1}(\text{view}) - l^r(\text{view})$$

By 5.3.3:

$$Pr\left[l^{r+t-1}(\text{EXEC}) < l^r(\text{EXEC}) + (1+\delta)\alpha(t-1)\right] < e^{-\Omega(\delta^2\alpha(t-1))}$$

Which means:

$$Pr\left[\min_{i,j}\left\{|C_j^{r+t}(\text{EXEC})| - |C_i^r(\text{EXEC})\right\} < (1+\delta)\alpha(t-1)\right] < e^{-\Omega(\delta^2\alpha(t-1))}$$

□

Now that we have both the consistent length and the minimum chain increase parts of the growth predicate we can prove it for our model by using a Union Bound over rounds $r$ (3.2).

The divergence in proofs demonstrates the difference in complexity that is created by adding delay to the model showing that it stems from the choices that the Adversary gains (in the selection of messages to delay and rounds to delay them by) rather than the simple fact that messages are being delayed.

## 5.4 Chain Quality

To begin with, this proof will require finding bounds on the rate the blocks are created and added to the blockchain both from the honest players and the Adversary. These bounds help us connect the number of blocks created to the number of rounds that have passed, enabling us to move from the domain of blocks to that of rounds and vice versa.
We start by showing a bound for the total number of blocks stamped in a window of time.

**Lemma 5.4.1.** *(Upperbound on Blocks.) Let $Q_t(view)$ be the maximum number of blocks stamped in any window of $t$ rounds in view. For any $t \geq 0$ and any $\delta$,*

$$Pr\left[Q_t(EXEC) > (1+\delta)(\alpha+\beta)t\right] < e^{-\Omega[\delta^2(\alpha+\beta)t]}$$

*Proof.* Each round at most one of the two can be true: either an honest player gets the stamp or the adversary gets the stamp. The first event happens with probability $\alpha$ and the second with probability $\beta$, therefore the probability that a block is stamped in any given round is $\alpha + \beta$.
We can therefore expect, for the total number of blocks stamped in a period of $t$ rounds ($\Sigma_t$):

$$E[\Sigma_t] = (\alpha + \beta)t$$

We can use 3.2 in order to arrive to our desired bound. □

We will then prove a bound for the number of adversarial blocks stamped in a window of time.

**Lemma 5.4.2.** *(Upperbound on Adversarial Blocks.) Let $A_t(view)$ be the maximum number of adversarial blocks stamped in any window of $t$ rounds in view. For any $t \geq 0$ and $\delta > 0$,*

$$Pr[A_t(EXEC) > (1+\delta)\beta t] < e^{-\Omega(\delta^2 \beta t)}$$

*Proof.* Similarly to 5.4.1, each round the Adversary can stamp at most one block with probability $\beta$.
Much like before this gives us, for the number of adversarial blocks stamped in a period of $t$ rounds ($Y_t$):

$$E[Y_t] = \beta t$$

Again, we use 3.2 to reach our desired bound. □

By combining the bounds on the total number of blocks created in a window and the number of adversarial blocks, we can ensure that the fraction of adversarial blocks in any sequence is limited. This directly leads to the Chain Quality property, where the proportion of adversarial blocks remains below a defined threshold.

**Theorem 5.4.3.** *For all $\delta > 0$, any $p(\cdot)$, $\left(\Pi^p_{stamp}, \mathcal{C}^p_{stamp}\right)$ has (errorless) chain quality $\mu = 1 - (1+\delta)\frac{\beta}{\alpha}$.*

*Proof.* Let us consider some round $r$ and a player $i$ that is honest at view$^r$. For the sake of simplicity we will define $C = \mathcal{C}^r_i(\text{view})$, in this we care about the blocks $b_0, ..., b_{|C|}$ that are contained in $C$. We will examine some sequence of $T$ consecutive blocks $C[j : j + T + 1] = b_j, ..., b_{j+T}$. A block $b_i$ will be considered adversarial if $b_i$ was stamped by and adversarial party. What we have to show is that the fraction of adversarial blocks in any such sequence is upper-bounded by $(1+\delta)\frac{\beta}{\alpha}$.


**Figure 5.1:** Sequence of $T$ consecutive blocks $C[j : j + T + 1] = b_j, ..., b_{j+T}$.

Now, we can safely assume that blocks $b_{j-1}$ and $b_{j+T+1}$ are not adversarial. If they are not we can simply increase the ratio of adversarial blocks by including them and making our sequence larger since we care about its maximum possible.

To reach the goal of proving the chain quality ratio we need to take a small detour, up to now we have considered a sequence of blocks, we will first translate this into a period of rounds.

For a sequence of blocks such as the above let:

- $r'$ be the round when block $b_{j-1}$ was stamped and sent out
- $r' + t$ be the round when block $b_{j+T}$ was stamped and sent out.

We know that all blocks in the sequence $C[j : j + T + 1]$ were stamped in the period of rounds between $r'$ and $r' + t$.

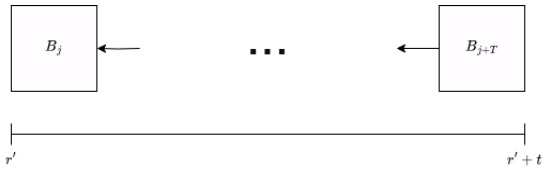



**Figure 5.2:** The sequence $C[j : j + T + 1]$ was stamped in the period of rounds between $r'$ and $r' + t$.

Now that we have a period of rounds to examine we will first define some events that should not happen and indeed don't happen except with negligible probability.

- We are currently considering a sequence of $T$ blocks, to proceed we need to translate this length to the number of rounds it was mined in. To achieve this we can use the first of the previous lemmas. Lemma 4.5.1 ensures that, for $t = \frac{T}{(1+\delta)(\alpha+\beta)}$:
  For every $0 < \delta < 1$ we have that $Q_{\frac{T}{(1+\delta)(\alpha+\beta)}} < T$ except with probability $e^{-\Omega(\delta^2 T)}$.
  Therefore, since our sequence is $T$ blocks long, we have that:

$$t > \frac{T}{(1+\delta)(\alpha+\beta)}$$

  except with probability $e^{-\Omega(\delta^2 T)}$.
  We define $bad1$ as the event that this does not hold.

- Since for every $\delta'$ the chain growth is bounded by $(1+\delta')\alpha$ we have that,

$$t \leq \frac{T}{(1-\delta')\alpha}$$

  except with probability $neg(T)$.
  We define $bad2$ as the event that this does not hold.

- By Lemma 5.4.2 we have that for every $\delta''$, the number of blocks stamped by the Adversary during this period is upper bounded by

$$(1+\delta'')\beta t \leq \frac{1+\delta''}{1-\delta'} \cdot T \cdot \frac{\beta}{\alpha}$$

We define $bad3$ as the event that this does not hold.
Since we assumed that $bad1$ does not happen:

$$\beta t = \rho npt > \rho np \frac{T}{(1+\delta)(\alpha+\beta)} > \rho np \frac{T}{(1+\delta)np}$$

Since $np > \alpha + \beta$:

$$\rho np \frac{T}{(1+\delta)(\alpha+\beta)} > \rho np \frac{T}{(1+\delta)np} = \frac{\rho T}{(1+\delta)}$$

This means that $bad3$ only happens with probability $neg(T)$.

To complete the proof we utilize the steps explained in 3.2 and apply a union bound while eliminating the polynomial factors to conclude that this upper bound holds with probability $neg(T)$. □

Comparing our proofs for the simplified stamp model with the proofs in [3], this one is the closest to its corresponding proof

Introducing delay into the analysis only forces us to replace the instances of $\alpha$ in it with the delay adjusted $\gamma$ which emerged from the Chain Growth proof. With that, the rest of the proof can proceed as normal since it only considers lengths of already mined, or in our case stamped, blocks.

Additionally, enabling multiple stamps to be given out each round would only change the numeric value of $\alpha$ (or $\gamma$ if we also add delay) and $\beta$. This wouldn't alter the proof in any meaningful way, so this variation too appears quite simple.

We can therefore draw the conclusion that the Chain Quality property is mostly impacted by the raw power of the honest and the adversarial party. Even in cases of adversarial interference such as the creation of delay, it is only honest power that is hindered while the core of the proof remains the same. This is in contrast to the Chain Growth property proof, which as we already discussed is heavily influenced by the actions of the adversarial parties. As we will see in the next section, the Consistency property and its proof is also deeply affected by the abilities of the adversary, leaving Chain Quality as the one outlier in the analysis.

## 5.5 Consistency

After having a short break with Chain Quality we once again arrive at a property that varies significantly from its original [3] version.

The Consistency property, simply put, states that no two honest players should hold chains that differ by more than the last $T$ blocks at any two points in time. This includes the same honest player seen in two different rounds.

From the very beginning of this proof attempt we notice a major difference:

Our model can't have two honest players in the same round holding two different chains. This happens simply because we have assumed that all messages are instantly delivered to everyone. So, as long as the honest players have a consistent way to pick between two chains of equal length, for example always considering the first one they obtained, they will always hold the same chain in each round. If this is the case, the only way for the Adversary to break the Consistency property is to make a single honest player (and therefore every honest player) hold chains that differ in more than the last $T$ blocks in two different points in time.

**Block Withholding:** To begin with, the authors of [3] prove that the Adversary doesn't benefit from withholding blocks for long. This means that if the adversarial parties collaborate to mine a large chain that they plan to release all at once to fork the chain Moreover, it means that if enough time has passed since an adversarial block was mined, whether or not it was published, the chances of it making it to the chain decrease. Therefore, the more time that passes without a fork overtaking the current longest chain, the less chance it has in doing so.

In our model, the Adversary, or any player for that matter, lacks the ability to keep the blocks they stamp a secret until it is convenient for them. Still, even if it is not exactly "withholding" of blocks, we should show that a block that was stamped and published by the Adversary a sufficiently long time ago won't end up as part of an honest player's chain.

**Lemma 5.5.1.** *If $\alpha \geq (1+\delta)\beta$ for some $0 < \delta < 1$ then, for every constant $0 < \omega < 1$ there exists a negligible function $\epsilon(\cdot)$ such that:*

$$Pr\left[\textit{view} \leftarrow \textit{EXEC} : \textit{withholding-time}(\textit{view}) \geq \omega t\right] \leq \epsilon(\beta t)$$

*Where withholding-time(view) is the longest number of rounds $t$ such that, in viewthe Adversary stamps a block $b$ at round $r$ and there exists some honest player $i$ such that $b$ first appeared as $i$'s chain at round $r + t$.*

*Proof.* Let us assume that this does not hold. For this reason we will consider the block $b_l$ that was stamped by the Adversary at round $r$. This block is then accepted by an honest party $i$ at round $s$, such that: $s - r \geq \omega t$. We also care about block $b_{l-k}$, that is the most recent (smallest $k$) non-adversarial block that is a prefix of block $b_l$. The block $b_{l-k}$ was mined at some round $r' \leq r$ therefore at least $\omega t$ rounds have passed between $r'$ and $s$.

Since block $b_l$ first makes to an honest player's chain at round $s$, every block that extends it and was stamped in the rounds between $r$ and $s$ has to have been stamped by the Adversary. It suffices to show that the Adversary couldn't have possibly mined a sufficient amount of blocks in that period.

As we did in the proof for the Chain Quality property, we will first define some events that we would like to not happen with overwhelming probability.

- At round $r'$ some honest player has a chain of length $l - k$ and at $s$ some honest player has a chain of length $l$. We already have a bound for the growth of a chain that is $(1 - \delta')\alpha t$ except with probability $neg(\alpha t)$ for any $\delta'$. With this we can say:

$$k \geq (1 - \delta')\alpha\omega t$$

The event that this doesn't hold will be $bad1$ and the probability of it happening is $neg(\beta t)$, since we have assumed that $\alpha \geq (1+\delta)\beta$.

We now know the number of rounds (or at least a bound for it) that have elapsed and can move on to finding a bound for the number of blocks that could have been stamped in that period.

- From the bound of adversarial blocks we know that for every $\delta''$, except with probability $e^{-\Omega(\beta t)}$ the number of adversarial blocks that could have been stamped in that period is at most:

$$(1+\delta'')\beta\omega t$$

The event that this doesn't hold will be $bad2$.

As we have already said, all blocks from $b_{l-k}$ to $b_l$ are adversarial and therefore:

$$k \leq (1+\delta'')\beta\omega t$$

If the events we have defined don't happen we can combine our two inequalities for $k$ to obtain:

$$(1-\delta')\alpha\omega t \leq (1+\delta'')\beta\omega t$$

We conclude that:

$$\alpha \leq \frac{1+\delta''}{1-\delta'}\beta$$

Since this is true for all $\delta', \delta''$ we can configure them to be small enough that:

$$\alpha < (1+\delta)\beta$$

which of course contradicts our initial assumption. □

The Block Withholding lemma guarantees that if an adversarial block has not been included in the chain within a certain time frame, its chances of ever making it into an honest chain diminish

**Divergence:** First we note what it means for two players to diverge at a specific round. Two chains $C_1$ and $C_2$ diverge at round $r$ in view if the last block that they have in common was stamped before round $r$.

**Convergence Opportunities:** Next we need to see how these divergences are resolved.

**Definition 5.5.2.** (*Convergence Opportunities*) A Convergence Opportunity occurs any time the honest players of the protocol have a chance to all obtain the same chain.

This is intuitively a really important moment that we would like to happen frequently. Anytime such an opportunity succeeds we can be sure that all honest players have the same chain, which is the very goal of our system. Conversely, in order for the Adversary to hinder the operation of the system and ruin its Consistency they would like for these opportunities to fail.

We can observe that anytime an honest block is stamped, it is sent out to every honest player instantaneously and without fail. Because of this, a convergence opportunity occurs anytime an honest block is stamped. If the Adversary has managed to split the players between two chains of equal lengths and an honest player manages to stamp a block for one of the two the Adversary then needs to also stamp a block on the other chain to keep the players split between them.

For the bounded delay case, this gets quite more complicated. Firstly the block that is mined to create the convergence opportunity should be unique. Next, the existence of delay means that we need to wait in order to be sure that blocks mined have reached all honest players.

A convergence opportunity in this scenario therefore involves:

1. $\Delta$ **rounds of silence.**

   This is required to ensure that every honest player can obtain a chain of the same length. These chains might still be different among players.

2. **A single new block is mined.**

   Since every honest player has a chain of the same length, this means that the player that mined this block has the longest chain in the entire protocol.

3. $\Delta$ **rounds of silence.**

   This ensures that the single longest chain that was just mined successfully reaches every honest player in the network. Now each and every one of them has the same chain.

**Lemma 5.5.3.** *Assume there is an $0 < \lambda < 1$ such that $a \geq (1+\delta)\beta$. Except with probability $e^{-\Omega(\beta t)}$ over view $\leftarrow$ EXEC, there do not exist rounds $r \geq r'$ and players $i, j$ such that $i$ is honest at round $r$ and $j$ is honest at round $r'$ and $\mathcal{C}_i^r(view)$ and $\mathcal{C}_j^{r'}(view)$ diverge at round $s = r - t$.*

*Proof.* Here we see an example of standard induction over $r' - r$. For this we need to prove our lemma for:

1. $r' - r = 0$, the base case and

2. $r' - r = k + 1$ where the lemma is true for $r' - r = k$, the induction step.

Of course if the two chains don't diverge at round $r' > r$ they don't diverge at round $r' = r$ either. For this reason the induction step is reduced to proving the lemma for $r' = r + 1$. Adding the base case to this we need only prove the lemma for:

$$r \leq r' \leq r + 1$$

We are going to do this through convergence opportunities. How many of those will there be between rounds $s = r - t$ and $r$? Since in our simple model a convergence opportunity happens

whenever an honest block is stamped we get (see the Chain Growth section) that there exists $0 < \delta' < 1$ such that:

$$Pr\left[< (1-\delta')\alpha t\right] < e^{-\Omega(\delta'^2 \alpha t)}$$

So there are at least $(1-\delta')\alpha t$ convergence opportunities. The adversarial parties will then have to ruin all of these and therefore will have to mine at least:

$$T_{\delta'} = (1-\delta')\alpha t$$

blocks in that period that need to be part of an honest player's chain at round $s$ or later.
**Note:** Here is the point at which the authors of [3] make use of their block withholding lemma. They do this because the blocks that take part in creating the divergence at round $s$ might have been mined before that round and kept secret, meaning the adversarial parties might already have access to a chain longer than the one available to the rest of the players at that round. In our case though this is impossible, since blocks are published instantly. Therefore if the adversary had created some longer already at that point it would already be the longest chain of every player. For this reason we only care about the blocks that are stamped by the adversary between rounds $s$ and $r'$.
We already have an upper bound for the adversarial blocks mined in the period between $s$ and $r'$. This is:

$$(1+\omega')(t+1)\beta$$

blocks except with probability $e^{-\Omega(\beta t)}$ Because of the assumption we made at the beginning we know that $\alpha \geq (1+\delta)\beta$ and therefore the Adversary could have mined at most

$$(1+\omega')(t+1)\beta \leq \frac{1+\omega}{1+\delta}(t+1)\alpha$$

blocks. We can now pick our parameters $\omega$ and $\delta'$ to be such that this measure is less than $T_{\delta'}$. With this the chains could diverge at $s$ only with probability $e^{-\Omega(\beta t)}$. □

As we can recall, [3] considers a different power relation between the honest and adversarial parties, one that includes delay in it. With that the authors move to find a bound for the number of convergence opportunities that occur in each period of rounds.

**Theorem 5.5.4.** *For any $\lambda > 0$, any $p(\cdot)$, $\left(\Pi^p_{stamp}, \mathcal{C}^p_{stamp}\right)$ satisfies (errorless) consistency.*

*Proof.* We take a view view in EXEC and players $i, j$ such that, in view $i$ is honest at round $r$ and $j$ is honest at round $r_2 > r_1$. Let:

$$C_1 = \mathcal{C}_i^{r_1}(\text{view})$$
$$C_2 = \mathcal{C}_j^{r_2}(\text{view})$$

For every constant $0 < \delta < 1$, by Lemma 5.4.3 the probability that $C_1$ and $C_2$ diverge at round $s = r_1 - \frac{T}{\alpha(1+\delta)}$ is at most:

$$e^{-\Omega(T\frac{\beta}{\alpha})} = e^{-\Omega(T\frac{1}{\rho})} = e^{-\Omega(T)}$$

By Lemma 5.3.1, except with probability $e^{-\Omega(T)}$, the number of blocks stamped between rounds $s$ and $r_1$ is smaller than:

$$(1+\delta)\cdot\alpha\cdot\frac{T}{\alpha(1+\delta)}=T$$

By using a union bound we get that the chains $C_1$ and $C_2$ don't diverge except maybe in the last $T$ blocks. □

# Chapter 6

# Conclusion

## 6.1 Takeaways

What have we learned through this exploration?

The blockchain is a recent technology that has gone through rapid evolution and has gained a lot of popularity focused mostly on its practical uses. To begin with we went through the mathematical and cryptographical foundation upon which the blockchain was created identifying the main tools it requires. For our main chapters we explore some fundamental works of blockchain analysis. These were the papers that managed to create a framework for formally analysing the blockchain an proving its desired properties. Lastly we went through the process of defining a blockchain protocol and proving some of its properties. Using something as simple as the Stamp Model we managed to provide a showcase of how proofs on the blockchain work and how they can be built upon to increase the model's fidelity. We believe that this can serve two goals:

1. Serves as an educational tool for those unfamiliar with blockchain technology and its mathematical foundations. This is achieved by focusing on a small and simple model and increasing complexity by pointing out how and why each part of the analysis change in order to make it more complex and powerful.

2. Second, it offers a simplified framework to test and develop more complex concepts progressively We believe it can be useful to occasionally take a step back from full scale models and try to understand how one can get there step by step.

## 6.2 Further Work

While our analysis focused on Proof of Work-based blockchains, there are multiple directions for expanding this work, including exploring alternative consensus mechanisms and optimizing blockchain architecture.

**Proof of Stake:** Everything we have talked about so far only considers blockchains that use Proof of Work as their basis. This though is not the only option available.

We could perform the same type of exploration as we did in this thesis for the analysis of Proof of Stake protocols introduced by the Ouroboros protocol [27]–[29]. Just like Proof of Work selects users to publish blocks based on their ability to solve cryptographic puzzles (their

work), Proof of Stake selects them based on the amount of currency (their stake) they have in the blockchain. By doing this they require less computational power and therefore a smaller energy cost while adopting the intuition that those with the largest stake in the protocol are likely to be honest players. One specificaly interesting approach is the one by Algorand [16], [19] which fully eliminates forks by developing a Proof of Stake voting system that elects a unique block at each round.

After deconstructing the way a Proof of Stake system is analyzed by trying to recreate it with a simple model similar to the one in this work it would be interesting to see how one moves to that from a Proof of Work system. What changes exactly have to be made in each part of the analyses of the two in order to bridge their gap?

It should also be simple enough to create the equivalent Stamp Model for the Proof of Stake scenario and use that to slowly build up the analysis of a full model.

**Multiple Chains:** We talked about deconstructing the way people have proved properties about blockchain protocols but there are in fact ways to deconstruct the blockchain idea itself. This is explored in papers such as [30] and [31], the basic premise of both being the same: how can we improve the performance of blockchain protocols through the parallelization of their processes? Performance here refers to the throughput and the latency of the protocol - in contrast to what we have seen up to now - real world time is taken into account. This provides new avenues for exploration and simplification, as our Stamp model has to be adapted to thinking about the round duration, network capacity and delays.

On the sightly simpler side, [30] decides to do this by trying to increase the number of chains the players are working on. One of the interesting aspects of this paper is the way the work of the players is split between these parallel chains and how this set of chains is then compiled into a single unique ledger for all to follow.

Taking this idea further, [31] deconstructs the way the mining and committing process itself works. As we have already discussed blockchain protocols need to have the transactions organized and published to the network, have these transactions be ordered and finally decide which ones will make it into the final chain / ledger. For this reason the chain is split into three parts that each perform one of these functions:

- Transaction blocks that simply contain published transactions and wait to be included in the ledger.
- A proposal chain to keep proposal blocks (blocks that point to transaction blocks) in a rough order, split into levels each of which will have one ”leader” (a voted upon block).
- Multiple chains, in the style of [30] that are used for voting which proposal block is the leader of each level.

Both of these approaches achieve much better throughput for bitcoin-like blockchain protocols.

To begin with, we could apply the ideas of [30] and [3] to our own simple Stamp model. With this we could explore exactly how these two papers try to improve the blockchain protocol with a bare-bones system as the basis. The idea of deconstructing the blockchain models fits naturally

with deconstructing the analysis of such models so we believe it would be a worthwhile effort for the sake of better understanding and explaining the blockchain and the ways it can improve.

# Bibliography


[1] S. Nakamoto, *Bitcoin: A peer-to-peer electronic cash system*, https://bitcoin.org/bitcoin.pdf, 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf.

[2] J. Katz and Y. Lindell, *Introduction to Modern Cryptography* (Chapman & Hall/CRC Cryptography and Network Security Series). CRC Press, 2020, ISBN: 9781351133012. [Online]. Available: https://books.google.gr/books?id=RsoOEAAAQBAJ.

[3] R. Pass, L. Seeman, and A. Shelat, *Analysis of the blockchain protocol in asynchronous networks*, Cryptology ePrint Archive, Paper 2016/454, https://eprint.iacr.org/2016/454, 2016. [Online]. Available: https://eprint.iacr.org/2016/454.

[4] J. Garay, A. Kiayias, and N. Leonardos, *The bitcoin backbone protocol: Analysis and applications*, Cryptology ePrint Archive, Paper 2014/765, https://eprint.iacr.org/2014/765 [current version], 2014. [Online]. Available: https://eprint.iacr.org/2014/765.

[5] J. A. Garay, A. Kiayias, and N. Leonardos, *The bitcoin backbone protocol with chains of variable difficulty*, Cryptology ePrint Archive, Paper 2016/1048, https://eprint.iacr.org/2016/1048, 2016. [Online]. Available: https://eprint.iacr.org/2016/1048.

[6] L. Ren, *Analysis of nakamoto consensus*, Cryptology ePrint Archive, Paper 2019/943, https://eprint.iacr.org/2019/943, 2019. [Online]. Available: https://eprint.iacr.org/2019/943.

[7] M. Okun, "Agreement among unacquainted byzantine generals," in *Proceedings of the 19th International Conference on Distributed Computing*, ser. DISC'05, Cracow, Poland: Springer-Verlag, 2005, pp. 499–500, ISBN: 3540291636. DOI: 10.1007/11561927_40. [Online]. Available: https://doi.org/10.1007/11561927_40.

[8] M. Okun and A. Barak, "Efficient algorithms for anonymous byzantine agreement," *Theor. Comp. Sys.*, vol. 42, no. 2, pp. 222–238, Jan. 2008, ISSN: 1432-4350. DOI: 10.1007/s00224-007-9006-9. [Online]. Available: https://doi.org/10.1007/s00224-007-9006-9.

[9] A. K. Miller and J. J. Laviola, "Byzantine consensus from moderately-hard puzzles : A model for bitcoin," 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:14522813.

[10] R. Pass and abhi shelat, *Micropayments for decentralized currencies*, Cryptology ePrint Archive, Paper 2016/332, 2016. [Online]. Available: https://eprint.iacr.org/2016/332.

[11] A. Kiayias, N. Leonardos, and D. Zindros, *Mining in logarithmic space*, Cryptology ePrint Archive, Paper 2021/623, 2021. [Online]. Available: https://eprint.iacr.org/2021/623.

[12] E. N. Tas, D. Tse, L. Yang, and D. Zindros, *Light clients for lazy blockchains*, Cryptology ePrint Archive, Paper 2022/384, 2022. [Online]. Available: https://eprint.iacr.org/2022/384.

[13] A. Kiayias and G. Panagiotakos, *Speed-security tradeoffs in blockchain protocols*, Cryptology ePrint Archive, Paper 2015/1019, 2015. [Online]. Available: https://eprint.iacr.org/2015/1019.

[14] C. HOSKINSON, *Why we are building cardano*, https://whitepaper.io/document/581/cardano-whitepaper, 2017. [Online]. Available: https://whitepaper.io/document/581/cardano-whitepaper.

[15] V. Buterin, *Ethereum: A next-generation smart contract and decentralized application platform*, https://ethereum.org/content/whitepaper/whitepaper-pdf/Ethereum_Whitepaper_-_Buterin_2014.pdf, 2014. [Online]. Available: https://ethereum.org/content/whitepaper/whitepaper-pdf/Ethereum_Whitepaper_-_Buterin_2014.pdf.

[16] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, *Algorand: Scaling byzantine agreements for cryptocurrencies*, Cryptology ePrint Archive, Paper 2017/454, https://eprint.iacr.org/2017/454, 2017. [Online]. Available: https://eprint.iacr.org/2017/454.

[17] C. Natoli and V. Gramoli, *The balance attack against proof-of-work blockchains: The r3 testbed as an example*, 2016. arXiv: 1612.09426 [cs.DC]. [Online]. Available: https://arxiv.org/abs/1612.09426.

[18] C. Natoli and V. Gramoli, "The balance attack or why forkable blockchains are ill-suited for consortium," in *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2017, pp. 579–590. DOI: 10.1109/DSN.2017.44.

[19] J. Chen and S. Micali, *Algorand*, 2017. arXiv: 1607.01341 [cs.CR]. [Online]. Available: https://arxiv.org/abs/1607.01341.

[20] Y. Sompolinsky and A. Zohar, *Accelerating bitcoin's transaction processing. fast money grows on trees, not chains*, Cryptology ePrint Archive, Paper 2013/881, https://eprint.iacr.org/2013/881, 2013. [Online]. Available: https://eprint.iacr.org/2013/881.

[21] Y. Sompolinsky, Y. Lewenberg, and A. Zohar, *SPECTRE: A fast and scalable cryptocurrency protocol*, Cryptology ePrint Archive, Paper 2016/1159, 2016. [Online]. Available: https://eprint.iacr.org/2016/1159.

[22] Y. Sompolinsky, S. Wyborski, and A. Zohar, *PHANTOM and GHOSTDAG: A scalable generalization of nakamoto consensus*, Cryptology ePrint Archive, Paper 2018/104, 2018. [Online]. Available: https://eprint.iacr.org/2018/104.

[23] R. Pass and E. Shi, *FruitChains: A fair blockchain*, Cryptology ePrint Archive, Paper 2016/916, https://eprint.iacr.org/2016/916, 2016. [Online]. Available: https://eprint.iacr.org/2016/916.

[24] A. Kiayias, A. Miller, and D. Zindros, *Non-interactive proofs of proof-of-work*, Cryptology ePrint Archive, Paper 2017/963, https://eprint.iacr.org/2017/963, 2017. [Online]. Available: https://eprint.iacr.org/2017/963.

[25] J. Garay, A. Kiayias, and N. Leonardos, *The bitcoin backbone protocol: Analysis and applications*, Cryptology ePrint Archive, Paper 2014/765, https://eprint.iacr.org/archive/2014/765/20140930:123325 [original version], 2014. [Online]. Available: https://eprint.iacr.org/archive/2014/765/20140930:123325.

[26] J. Garay, A. Kiayias, and N. Leonardos, *The bitcoin backbone protocol: Analysis and applications*, Cryptology ePrint Archive, Paper 2014/765, https://eprint.iacr.org/archive/2014/765/20170214:030133 [intermediate version], 2014. [Online]. Available: https://eprint.iacr.org/archive/2014/765/20170214:030133.

[27] A. Kiayias, A. Russell, B. David, and R. Oliynykov, *Ouroboros: A provably secure proof-of-stake blockchain protocol*, Cryptology ePrint Archive, Paper 2016/889, https://eprint.iacr.org/2016/889, 2016. [Online]. Available: https://eprint.iacr.org/2016/889.

[28] B. David, P. Gaži, A. Kiayias, and A. Russell, *Ouroboros praos: An adaptively-secure, semi-synchronous proof-of-stake protocol*, Cryptology ePrint Archive, Paper 2017/573, 2017. [Online]. Available: https://eprint.iacr.org/2017/573.

[29] C. Badertscher, P. Gazi, A. Kiayias, A. Russell, and V. Zikas, *Ouroboros genesis: Composable proof-of-stake blockchains with dynamic availability*, Cryptology ePrint Archive, Paper 2018/378, 2018. [Online]. Available: https://eprint.iacr.org/2018/378.

[30] M. Fitzi, P. Gaži, A. Kiayias, and A. Russell, *Parallel chains: Improving throughput and latency of blockchain protocols via parallel composition*, Cryptology ePrint Archive, Paper 2018/1119, https://eprint.iacr.org/2018/1119, 2018. [Online]. Available: https://eprint.iacr.org/2018/1119.

[31] V. Bagaria, S. Kannan, D. Tse, G. Fanti, and P. Viswanath, *Deconstructing the blockchain to approach physical limits*, Cryptology ePrint Archive, Paper 2018/992, https://eprint.iacr.org/2018/992, 2018. [Online]. Available: https://eprint.iacr.org/2018/992.

[32] R. Canetti, *Universally composable security: A new paradigm for cryptographic protocols*, Cryptology ePrint Archive, Paper 2000/067, https://eprint.iacr.org/2000/067, 2000. [Online]. Available: https://eprint.iacr.org/2000/067.

[33] A. Kiayias and G. Panagiotakos, *On trees, chains and fast transactions in the blockchain*, Cryptology ePrint Archive, Paper 2016/545, 2016. [Online]. Available: https://eprint.iacr.org/2016/545.

[34] J.-P. Martin and L. Alvisi, "Fast byzantine consensus," in *2005 International Conference on Dependable Systems and Networks (DSN'05)*, 2005, pp. 402–411. DOI: 10.1109/DSN.2005.48.

[35] R. Pass and E. Shi, *Hybrid consensus: Efficient consensus in the permissionless model*, Cryptology ePrint Archive, Paper 2016/917, 2016. [Online]. Available: https://eprint.iacr.org/2016/917.