ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ & ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Αξιοπιστία Αλγορίθμων Μηχανικής Μάθησης με Επαλήθευση Αποδείξεων Μηδενικής Γνώσης σε Περιβάλλοντα Αλυσίδων-Κορμού

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Νικολέττα-Ευσταθία Δ. Παπαγεωργίου

Επιβλέπουσα : Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π

Αθήνα, Νοέμβριος 2023

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ & ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Αξιοπιστία Αλγορίθμων Μηχανικής Μάθησης με Επαλήθευση Αποδείξεων Μηδενικής Γνώσης σε Περιβάλλοντα Αλυσίδων-Κορμού

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Νικολέττα-Ευσταθία Δ. Παπαγεωργίου

Επιβλέπουσα: Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 7η Νοεμβρίου 2023.

…….	…….	…….
Θεοδώρα Βαρβαρίγου	Συμεών Παπαβασιλείου	Εμμανουήλ Βαρβαρίγος
Καθηγήτρια Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος 2023

.....

Νικολέττα-Ευσταθία Δ. Παπαγεωργίου
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Η τεχνητή νοημοσύνη και ειδικότερα η μηχανική μάθηση εμφανίζουν ραγδαία ανάπτυξη σε πληθώρα εφαρμογών. Ο όγκος των δεδομένων που αυτές επεξεργάζονται, σε συνδυασμό με την περίπλοκη διαδικασία εκπαίδευσης και τις ακόμα πιο περίπλοκες αρχιτεκτονικές των νευρωνικών δικτύων που αναπτύσσονται, καθιστούν την εποπτεία τέτοιων συστημάτων πολύ δύσκολη. Προκύπτει λοιπόν η ανάγκη για τον έλεγχο της αξιοπιστίας και της εγκυρότητας των αλγορίθμων μηχανικής μάθησης αλλά και για την εξασφάλιση της αμεταβλητότητάς τους. Η τεχνολογία αιχμής που βρίσκεται στο επίκεντρο της έρευνας τα τελευταία χρόνια και μπορεί να προσφέρει τη λύση στα παραπάνω ζητήματα είναι το περιβάλλον αλυσίδας-κορμού (blockchain). Παράλληλα, η μέθοδος της Απόδειξης Μηδενικής Γνώσης (Zero Knowledge Proof) επιλύει ένα ακόμα επίκαιρο πρόβλημα που συνδέεται με την χρήση τέτοιων δικτύων: την προστασία των ευαίσθητων και ιδιωτικών δεδομένων των κόμβων. Σκοπός της παρούσας διπλωματικής εργασίας είναι η επικύρωση της αξιοπιστίας αλγορίθμων μηχανικής μάθησης με επαλήθευση Αποδείξεων Μηδενικής Γνώσης σε περιβάλλοντα αλυσίδων-κορμού. Η υλοποίηση του δικτύου γίνεται με τη βοήθεια της πλατφόρμας Hyperledger Fabric, η επιβεβαίωση των αποδείξεων γίνεται από το έξυπνο συμβόλαιο (smart contract) γραμμένο σε γλώσσα GO, ενώ η διεπαφή με το δίκτυο επιτυγχάνεται με τον σχεδιασμό ενός API κατασκευασμένου με Node.js. Για τη δημιουργία των αποδείξεων χρησιμοποιούνται διαφορετικοί αλγόριθμοι κατακερματισμού και αξιολογείται η απόδοσή τους.

Λέξεις-κλειδιά: μηχανική μάθηση, περιβάλλον αλυσίδας-κορμού, ιδιωτικότητα, προστασία δεδομένων, έξυπνο συμβόλαιο, απόδειξη μηδενικής γνώσης, επιβεβαίωση, συναρτήσεις κατακερματισμού

Abstract

Artificial intelligence (AI) and machine learning (ML) are rapidly developing in a variety of applications. The volume of data they process, combined with the complex training process and even more complex architectures of neural networks that are developed, make it very difficult to supervise such systems. This leads to the need to control the reliability and validity of ML algorithms, as well as to ensure their immutability. The cutting-edge technology that has been at the forefront of research in recent years and can offer a solution to the above issues is Blockchain. In parallel, the Zero Knowledge Proof (ZKP) method solves another pressing problem associated with the use of such networks: the protection of sensitive and private data of nodes. The purpose of this thesis is to validate the reliability of ML algorithms with verification of Zero Knowledge Proofs in Blockchain Environments. The network is implemented with the help of the Hyperledger Fabric platform, the verification of proofs is performed by the smart contract written in GO language, while the interface with the network is achieved by designing an API built with Node.js. For the creation of proofs, different hashing algorithms are used and their performance is evaluated.

Keywords: Machine Learning, Blockchain, Privacy, Data Protection, On-Chain Activities, Smart Contract, Zero Knowledge Proof, Verification, Hash function

Ευχαριστίες

Η εκπόνηση της Διπλωματικής μου Εργασίας σηματοδοτεί το πέρας των σπουδών μου στη Σχολή των Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου (ΕΜΠ).

Πρωτίστως, θα ήθελα να ευχαριστήσω την κ. Βαρβαρίγου, Καθηγήτρια ΕΜΠ και επιβλέπουσα της παρούσας Διπλωματικής Εργασίας, για την εμπιστοσύνη και την ευκαιρία που μου έδωσε να μελετήσω ένα τόσο ενδιαφέρον θέμα. Παράλληλα, θα ήθελα να ευχαριστήσω τους ερευνητές Αντώνιο Λίτκε και Νίκο Καψούλη για το ενδιαφέρον, τη βοήθεια και την πολύτιμη καθοδήγηση που μου παρείχαν καθ' όλη τη διάρκεια της εργασίας.

Τέλος, θα ήθελα να ευχαριστήσω από καρδιάς τους γονείς μου, Δημοσθένη και Χαρά, τα αγαπημένα μου αδέρφια, Κατερίνα και Θοδωρή, αλλά και τους φίλους μου, για τη συνεχή στήριξη που μου προσέφεραν όλα αυτά τα χρόνια.

Περιεχόμενα

Ευρετήριο Εικόνων

Ευρετήριο Πινάκων

Ευρετήριο Διαγραμμάτων

Κεφάλαιο 1

Εισαγωγή

1.1 Εισαγωγή στο αντικείμενο της Διπλωματικής Εργασίας

Στις μέρες μας, η ανάπτυξη της τεχνολογίας φέρνει τους ανθρώπους αντιμέτωπους με ένα μεγάλο όγκο πληροφοριών αγνώστου προέλευσης και εγκυρότητας. Καθημερινά, παράγονται εκατομμύρια δεδομένα τα οποία είναι απαραίτητο να αξιολογηθούν προτού χρησιμοποιηθούν. Αντιστοίχως, η άνθιση της τεχνητής νοημοσύνης έχει ως αποτέλεσμα να εμφανίζονται συνεχώς νέοι αλγόριθμοι, οι δημιουργοί των οποίων υποστηρίζουν ότι είναι καλύτεροι από τους προηγούμενους. Επομένως, είναι αναγκαίο να γίνεται αξιολόγηση των αλγορίθμων από κατάλληλους φορείς σε ένα ασφαλές περιβάλλον που θα εξασφαλίζει την ιδιωτικότητα και θα παράγει ένα αμετάβλητο και ταυτόχρονα έγκυρο αποτέλεσμα. Τη λύση σε αυτό το πρόβλημα έρχονται να δώσουν τα περιβάλλοντα αλυσίδας-κορμού (blockchain).

Μολονότι η τεχνολογία του blockchain έγινε γνωστή και ταυτίστηκε με την έννοια των κρυπτονομισμάτων, στις μέρες μας υιοθετείται και σε άλλους τομείς πέραν της οικονομίας. Ο λόγος που επικράτησε αυτή η τεχνολογία είναι ότι προσφέρει ένα αποκεντρωμένο σύστημα το οποίο χαρακτηρίζεται από ασφάλεια, διαφάνεια και εμπιστοσύνη, ενώ παράλληλα είναι και

αποδοτικό. Εξουσιοδοτημένα μέλη του blockchain μπορούν μέσω αλγορίθμων συναίνεσης (consensus algorithms) να αποφανθούν αν ένας αλγόριθμος μηχανικής μάθησης είναι έγκυρος. Οι συναλλαγές υπογράφονται ηλεκτρονικά με τη βοήθεια ενός ιδιωτικού κλειδιού (private key) και η αυθεντικότητα της υπογραφής μπορεί να επιβεβαιωθεί με τη χρήση του δημόσιου κλειδιού (public key). Παρότι δεν αποκαλύπτεται η ταυτότητα του υπογράφοντος, μπορεί να γίνει στοχοποίησή του αφού γνωρίζουμε πόσες και ποιες συναλλαγές έχει υπογράψει. Για την περαιτέρω ενίσχυση της ιδιωτικότητας η τεχνολογία του blockchain συνδυάζεται με την τεχνική απόδειξης μηδενικής γνώσης.
Η Απόδειξη μηδενικής Γνώσης (Zero Knowledge Proof) είναι ένα πρωτόκολλο που επιτρέπει στη μία πλευρά μίας επικοινωνίας να πείσει την άλλη για την εγκυρότητα μιας πρότασης χωρίς την φανέρωση κάποιας πληροφορίας εκτός από την πιστότητα της απόδειξης.

Στα πλαίσια της παρούσας διπλωματικής εργασίας προτείνεται ένα ολοκληρωμένο πληροφοριακό σύστημα το οποίο αξιολογεί την αξιοπιστία ενός αλγορίθμου μηχανικής μάθησης μέσω επαλήθευσης αποδείξεων μηδενικής γνώσης σε περιβάλλοντα αλυσίδων-κορμού. Η προτεινόμενη λύση χρησιμοποιεί το Hyperledger Fabric blockchain ως μέσο για την αποθήκευση δεδομένων, την γνωστοποίηση αιτημάτων και την επικοινωνία των κόμβων. Αφού εξουσιοδοτημένα μέλη έχουν αποφανθεί για την εγκυρότητα των αλγορίθμων μηχανικής μάθησης, δημιουργούμε αποδείξεις και τις υποβάλουμε στο blockchain. Έπειτα, οποιοσδήποτε χρήστης μπορεί να αποφανθεί για την αξιοπιστία του επιλεγμένου αλγορίθμου μηχανικής μάθησης μέσω της επιβεβαίωσης των αποδείξεων.

1.2 Οργάνωση της Διπλωματικής Εργασίας

Η παρούσα διπλωματική εργασία διαρθρώνεται σε 5 Κεφάλαια.
Στο Κεφάλαιο 1 κάναμε μία εισαγωγή στο θέμα , αναφέραμε τις πρακτικές εφαρμογές αλλά και τους λόγους για τους οποίους είναι απαραίτητη η έρευνα και εμβάθυνση στο συγκεκριμένο αντικείμενο.
Στο Κεφάλαιο 2 αναπτύσσουμε το θεωρητικό υπόβαθρο που είναι απαραίτητο για την κατανόηση της παρούσας εργασίας. Αναλύουμε βασικές έννοιες όπως τι είναι blockchain, την λογική και τους αλγορίθμους πίσω από τις συναρτήσεις κατακερματισμού και πως λειτουργούν

οι αποδείξεις μηδενικής γνώσης. Ταυτόχρονα, παρουσιάζουμε σχετικές μελέτες που προσεγγίζουν το αντικείμενο που μας απασχολεί.

Στο Κεφάλαιο 3 παρουσιάζουμε την αρχιτεκτονική του πληροφοριακού συστήματος. Αναφέρουμε την συλλογιστική πορεία που ακολουθήσαμε αλλά και τους λόγους που επιλέξαμε τις συγκεκριμένες τεχνολογίες και τεχνικές για να υλοποιήσουμε το συγκεκριμένο πληροφοριακό σύστημα.

Στο Κεφάλαιο 4 παραθέτουμε τα αποτελέσματα των πειραμάτων μας, εξηγούμε αν είναι αυτά που θα περιμέναμε και κάνουμε τον αντίστοιχο σχολιασμό.

Τέλος στο Κεφάλαιο 5 ολοκληρώνεται η παρούσα διπλωματική εργασία με τα τελικά συμπεράσματα που προέκυψαν και τις μελλοντικές κατευθύνσεις στις οποίες θα μπορούσε να επεκταθεί η συγκεκριμένη μελέτη.

Κεφάλαιο 2

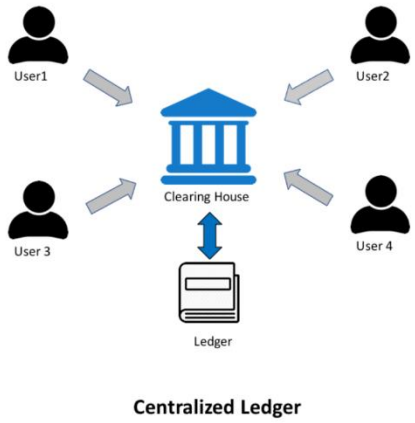
Θεωρητικό Υπόβαθρο και Σχετικές Μελέτες

2.1 Blockchain

Η τεχνολογία blockchain έχει κεντρίσει τα τελευταία χρόνια την προσοχή της παγκόσμιας κοινότητας. Από τότε που προτάθηκε ερευνητές από όλο τον κόσμο άρχισαν να εξερευνούν τις προοπτικές της τεχνολογίας αυτής. Η ιδέα του bitcoin, ενός κρυπτονομίσματος, και του blockchain συστήθηκαν πρώτη φορά από κάποιον που χρησιμοποιούσε το ψευδώνυμο Satoshi Nakamoto [1]. Στην αρχή υπήρχε καχυποψία για το bitcoin και τη χρησιμότητά του λόγω των έντονων αλλαγών της τιμής του και της πολυπλοκότητας που το περιέβαλε. Ωστόσο, τα πλεονεκτήματα της τεχνολογίας που ήταν πίσω από το bitcoin, δηλαδή του blockchain , δεν μπορούσαν να αγνοηθούν. Το ενδιαφέρον που αναπτύχθηκε γύρω από αυτήν την νέα τεχνολογία είχε ως αποτέλεσμα να έχουμε σήμερα μία πληθώρα εφαρμογών που τη χρησιμοποιούν από τους Τομείς Υγείας μέχρι τα Τραπεζικά και Πολιτικά συστήματα. [2]

2.1.1 Λογιστικός Κατάλογος (Ledger)

Ο Λογιστικός Κατάλογος είναι ένα βιβλίο με καταγραφές που περιλαμβάνει πληροφορίες σχετικά με συναλλαγές μεταξύ 2 ή περισσότερων ατόμων ή οργανισμών. Οι καταγραφές αυτές είναι πολύ σημαντικές αφού λειτουργούν ως μια πηγή αδιαμφισβήτητης αλήθειας όταν πρόκειται να ληφθούν για παράδειγμα επιχειρηματικές αποφάσεις. Με την ανάπτυξη της τεχνολογίας ο Λογιστικός Κατάλογος πήρε ηλεκτρονική μορφή. Οι Λογιστικοί Κατάλογοι χωρίζονται σε δύο κατηγορίες: Centralized Ledgers και Decentralized Ledgers.



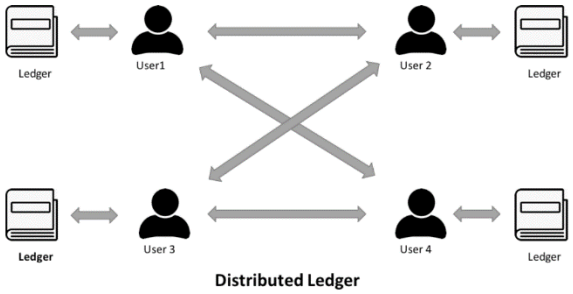
Εικόνα 1: Συγκεντρωτικός Κατάλογος [4]

Centralized Ledger

Ο Συγκεντρωτικός Κατάλογος (Centralized Ledger) διοικείται και ελέγχεται από μία μόνο οντότητα, όπως μια επιχείρηση, έναν κυβερνητικό φορέα ή ένα χρηματοπιστωτικό ίδρυμα. Όλα τα δεδομένα αποθηκεύονται σε μια ενιαία τοποθεσία σε ένα κεντρικό σύστημα και η πρόσβαση σε αυτά ελέγχεται από εξουσιοδοτημένα άτομα ή ιδρύματα. Η συγκεντρωτική φύση του centralized ledger έχει ως αποτέλεσμα να είναι πιο ευάλωτο σε επιθέσεις στον κυβερνοχώρο και να μην υπάρχει διαφάνεια καθιστώντας δύσκολο να γίνει επαλήθευση ως προς την ορθότητα και την πληρότητα των δεδομένων.

Decentralized Ledger

Ο αποκεντρωμένος Λογιστικός Κατάλογος (Decentralized Ledger) είναι ευρέως γνωστός ως Κατανεμημένος Λογιστικός Κατάλογος (Distributed Ledger). Η τεχνολογία Κατανεμημένου Λογιστικού Καταλόγου (DLT) επιτρέπει την καταγραφή και αποθήκευση δεδομένων σε ένα δίκτυο πολλών υπολογιστών. Τα δεδομένα είναι κρυπτογραφημένα και διαμοιρασμένα μεταξύ όλων των κόμβων του δικτύου, γεγονός που τα καθιστά ασφαλή και αξιόπιστα.



Εικόνα 2: Κατανεμημένος Λογιστικός Κατάλογος[4]

Η τεχνολογία DLT έχει μια σειρά από πλεονεκτήματα έναντι των παραδοσιακών συστημάτων καταγραφής δεδομένων, όπως:

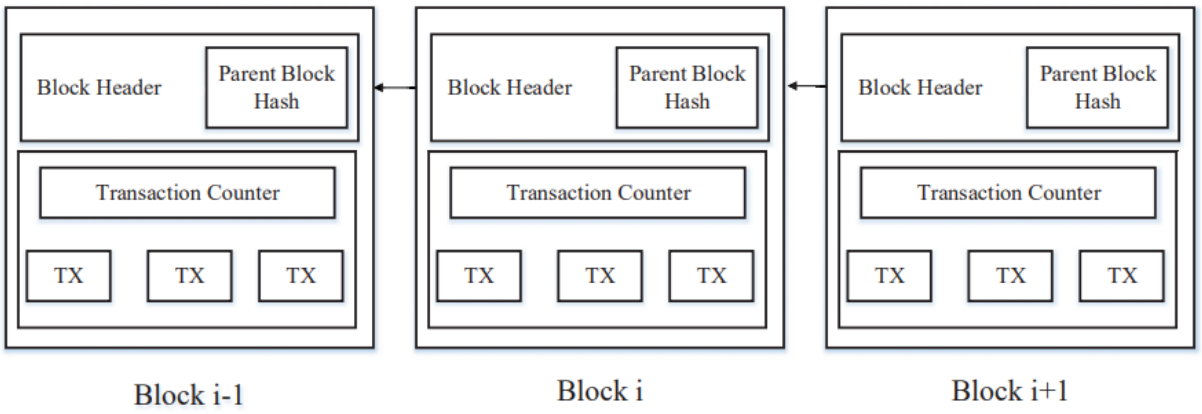
- Ασφάλεια: Τα δεδομένα είναι κρυπτογραφημένα και αποθηκεύονται σε πολλούς κόμβους του δικτύου, γεγονός που καθιστά δύσκολη την παραβίαση ή την τροποποίηση.
- Αξιοπιστία: Τα δεδομένα είναι διαθέσιμα σε όλους τους κόμβους του δικτύου, γεγονός που τα καθιστά πιο αξιόπιστα από τα δεδομένα που αποθηκεύονται σε μια κεντρική τοποθεσία.
- Διαφάνεια: Όλοι οι κόμβοι του δικτύου έχουν πρόσβαση στα ίδια δεδομένα, γεγονός που αυξάνει τη διαφάνεια.

Σε ένα distributed ledger δεν είναι απαραίτητο όλοι οι κόμβοι να διατηρούν όλες τις πληροφορίες του ledger. Αν πάλι τις διατηρούν, δεν είναι απαραίτητο ότι μπορούν να τις αποκρυπτογραφήσουν. Για παράδειγμα, στο Ethereum blockchain όλοι οι κόμβοι λαμβάνουν και κατανοούν όλες τις πληροφορίες. Αντίθετα, στο corda μόνο οι κόμβοι που συμμετέχουν σε μία συναλλαγή έχουν γνώση για την ύπαρξή της.[5]

Αν και οι έννοιες blockchain και distributed ledger συχνά συγχέονται στην πραγματικότητα διαφέρουν. Κάθε blockchain είναι μία υλοποίηση DLT, αλλά κάθε DLT δεν είναι blockchain. Μάλιστα, το blockchain ήταν ο πρώτος λειτουργικός τρόπος υλοποίησης ενός DLT.

2.1.2 Ορισμός Blockchain και Ανάλυση

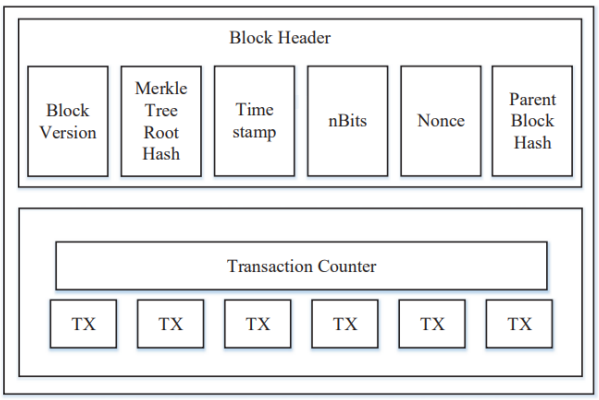
Το Blockchain είναι μια αλληλουχία από κομμάτια (blocks), η οποία διατηρεί μια πλήρη λίστα από καταγραφές συναλλαγών, όπως ένας συμβατικός λογιστικός κατάλογος. Στην παρακάτω εικόνα φαίνεται ένα διάγραμμα ενός blockchain. Κάθε block περιέχει στην κεφαλίδα του (header) την ταυτότητα του προηγούμενου block, κρυπτογραφημένη με μια συνάρτηση hash. Κάθε block έχει μόνο ένα προηγούμενο block (parent block). Αξίζει να σημειώσουμε ότι σε κάποια blockchain - όπως στο Ethereum blockchain - αποθηκεύονται επιπλέον και τα hash των uncle blocks, δηλαδή των παιδιών των προγόνων του κάθε block [6]. Το πρώτο block του blockchain ονομάζεται block γεννήσεως (genesis block) και δεν έχει προηγούμενο block. Η ανάλυση της εσωτερικής δομής του block είναι σημαντική για την κατανόηση της λειτουργίας του blockchain.



Εικόνα 3: Παράδειγμα blockchain [6]

Block

Ένα block περιέχει την κεφαλίδα του (block header) και το κυρίως σώμα (body) όπως φαίνεται στην παρακάτω εικόνα.



Εικόνα 4: Δομή Block [6]

Η κεφαλίδα του block περιέχει:

- την έκδοση του block (block version), η οποία χρησιμοποιείται για τον προσδιορισμό των κανόνων επικύρωσης και λειτουργίας που ακολουθεί το block.
- Τη ρίζα δέντρου Merkle, κρυπτογραφημένη με συνάρτηση hash (Merkle tree root hash). Αυτή είναι η τιμή hash όλων των συναλλαγών εντός του block.
- Τη χρονοσφραγίδα (timestamp), δηλαδή τη διεθνή ώρα σε δευτερόλεπτα, μετρώντας από την 1 Ιανουαρίου του 1970.
- Μια σταθερά nBits που είναι το όριο ενός επικυρωμένου block hash.
- Τη μεταβλητή Nonce, η οποία ξεκινάει από την τιμή 0 και αυξάνεται μετά από κάθε υπολογισμό κρυπτογράφησης hash.
- Την τιμή hash του προηγούμενου block, μεγέθους 256-bit.

Το κυρίως σώμα του block περιέχει έναν μετρητή συναλλαγών και τις ίδιες τις συναλλαγές. Ο μέγιστος αριθμός των συναλλαγών που μπορεί ένα block να περιέχει εξαρτάται από το μέγεθος του block και το μέγεθος της κάθε συναλλαγής.

Βασικά Χαρακτηριστικά Blockchain

Σε μια σύνοψη, τα βασικά χαρακτηριστικά του blockchain είναι τα παρακάτω:

- *Αποκέντρωση (Decentralization)*: Σε ένα συμβατικό μη αποκεντρωμένο σύστημα συναλλαγών, κάθε συναλλαγή πρέπει να επικυρώνεται μέσω μιας κεντρικής και έμπιστης οντότητας, όπως για παράδειγμα μια κεντρική τράπεζα. Έτσι, η υπολογιστική

δύναμη επικύρωσης συναλλαγών, περιορίζεται ανάλογα με την ισχύ της εκάστοτε κεντρικής οντότητας. Για παράδειγμα, μπορεί να περιορίζεται από τον αριθμό των εξυπηρετητών (servers) της κεντρικής τράπεζας. Αντιθέτως, κάτι τέτοιο δεν συμβαίνει με το blockchain, στο οποίο χρησιμοποιούνται αλγόριθμοι συναίνεσης για τη διατήρηση της συνέπειας των δεδομένων στο κατανεμημένο δίκτυο.

- *Διατηρησιμότητα (persistency):* Οι συναλλαγές μπορούν να επικυρωθούν γρήγορα και οι άκυρες συναλλαγές δεν γίνονται αποδεκτές από το δίκτυο. Είναι σχεδόν αδύνατο να διαγραφεί ή να αναιρεθεί μια έγκυρη συναλλαγή και τα blocks που περιέχουν άκυρες συναλλαγές ανακαλύπτονται ταχύτατα.
- *Ανωνυμία* (Anonymity): Κάθε χρήστης του blockchain αλληλεπιδρά με αυτό με μια ηλεκτρονική διεύθυνση χωρίς να εκθέτει την προσωπική του ταυτότητα.
- *Ελεγξιμότητα* (Auditability): ΟΙ συναλλαγές μπορούν να εντοπίζονται, να ελέγχονται και να επικυρώνονται με ευκολία από τρίτους παράγοντες.

2.1.3 Κατηγορίες Blockchain

Εξαιτίας του πλήθους των εφαρμογών που αξιοποιούν την τεχνολογία του blockchain, προέκυψαν τέσσερις τύποι blockchain ανάλογα με τα δικαιώματα και την πρόσβαση των χρηστών/κόμβων σε αυτά. Οι τύποι αυτοί είναι:

- **Δημόσιο Blockchain (Public Blockchain)**

Σε αυτόν τον τύπο Blockchain οποιοσδήποτε μπορεί να εισέλθει στο δίκτυο σαν κόμβος, χωρίς άδεια ή περιορισμούς (permissionless network). Το δίκτυο είναι πλήρως αποκεντρωμένο και οι πληροφορίες των συναλλαγών είναι κατανεμημένες στους κόμβους [7]. Λόγω της αποκεντρωμένης φύσης του δημοσίου blockchain απαιτείται να επικυρώνονται οι συναλλαγές μέσω κάποιου αλγόριθμου συναίνεσης (consensus algorithm). Όλοι οι χρήστες έχουν πρόσβαση στα αρχεία συναλλαγών και όλοι οι χρήστες μπορούν να συμμετέχουν στις διαδικασίες επικύρωσης. Καμία συναλλαγή που έχει επικυρωθεί δεν μπορεί να αλλάξει.

Πλεονεκτήματα: Το δημόσιο Blockchain είναι πλήρως ανεξαρτητοποιημένο από κεντρικούς οργανισμούς ελέγχου. Αυτό σημαίνει ότι μπορεί να λειτουργεί χωρίς την παρουσία κάποιας ρυθμιστικής αρχής, αλλά και ότι δεν μπορεί να χειραγωγηθεί ή να μειωθεί η διαφάνειά του από μια κεντρική αρχή.

Μειονεκτήματα: Το δίκτυο είναι αργό και οι διαδικασίες επικύρωσης μπορεί να δημιουργήσουν καθυστέρηση στις συναλλαγές. Επιπλέον, τα δημόσια blockchain δεν κλιμακώνονται εύκολα, καθώς αν περισσότεροι κόμβοι ενταχθούν στο δίκτυο, αυτό μπορεί να επιβαρυνθεί σημαντικά.

Χρήσεις: Τα δημόσια blockchain χρησιμοποιούνται κυρίως για συναλλαγές κρυπτονομισμάτων (όπως το Bitcoin). Οι δυνατότητες ενός τέτοιου δικτύου προσφέρονται για χρήση από οργανισμούς που απαιτούν διαφάνεια και πλήρη εμπιστοσύνη, όπως μη κυβερνητικές οργανώσεις και ομάδες κοινωνικής υποστήριξης. Οι ιδιωτικές εταιρείες στρέφονται συνήθως προς άλλα είδη blockchain.

- **Ιδιωτικό Blockchain (Private Blockchain)**

Σε ένα ιδιωτικό blockchain δεν επιτρέπεται σε οποιονδήποτε να εισέλθει στο δίκτυο, αλλά η πρόσβαση περιορίζεται σε μία ομάδα ανθρώπων και ιδρυμάτων (permissioned network). Η αποκέντρωση των δεδομένων και η επικύρωση των συναλλαγών μπορεί να εμφανίζουν ομοιότητα με ένα δημόσιο blockchain, ωστόσο μια ελεγκτική αρχή θέτει επίπεδα πρόσβασης, ασφάλειας και καθορίζει τις αρμοδιότητες των κόμβων. Συνήθως, αυτού του τύπου τα blockchain αξιοποιούνται σε μικρότερη κλίμακα από τα δημόσια και ως εκ τούτου αποκαλούνται και εταιρικά blockchain (enterprise blockchain). [9]

Πλεονεκτήματα: Εξαιτίας του μικρού μεγέθους τους, αλλά και της ευελιξίας στους τρόπους επικύρωσης των συναλλαγών, τα ιδιωτικά blockchain είναι πιο γρήγορα από τα δημόσια. Τα ρευστά δικαιώματα των κόμβων δίνουν τη δυνατότητα σε μια κεντρική αρχή να ελέγχει και να ρυθμίζει τη λειτουργία του δικτύου με ευκολία.

Μειονεκτήματα: Τα ιδιωτικά blockchain χάνουν κάποια από τα βασικά και χαρακτηριστικά πλεονεκτήματα του blockchain. Δεν υπάρχει ανωνυμία και πλήττεται συνήθως η πλήρης διαφάνεια και η ολική αποκέντρωση του δικτύου. Συχνά ο κώδικας που ρυθμίζει τη λειτουργία τέτοιων δικτύων είναι κλειστός και δεν μπορεί να αλλάξει ή να ελεγχθεί από τους χρήστες.

Χρήσεις: Τα ιδιωτικά blockchain χρησιμοποιούνται από οργανισμούς οι οποίοι επιθυμούν να αξιοποιούν την κρυπτογραφική ασφάλεια και την εσωτερική αποκέντρωση του blockchain, ενώ δεν επιθυμούν οι πληροφορίες των συναλλαγών εντός του δικτύου να είναι διαθέσιμες δημοσίως. Παραδείγματα αποτελούν οι εσωτερικές ψηφοφορίες, οι συναλλαγές ιδιωτικής ιδιοκτησίας και η διαχείριση αλυσίδων εφοδιασμού.

- **Υβριδικό Blockchain (Hybrid Blockchain)**

Πολλές φορές είναι χρήσιμο να συνδυαστούν τα πλεονεκτήματα των δύο παραπάνω τύπων blockchain. Σε ένα υβριδικό blockchain οποιοσδήποτε μπορεί να εισέλθει χωρίς άδεια. Μέρος των δεδομένων είναι διαθέσιμο σε όλους τους χρήστες ενώ για τα υπόλοιπα δεδομένα η πρόσβαση είναι περιορισμένη. Όταν ένας χρήστης εισέρχεται στο δίκτυο, έχει πλήρη πρόσβαση σε αυτό και η ταυτότητά του είναι προστατευμένη. Όταν πραγματοποιεί μια συναλλαγή, η ταυτότητα του αποκαλύπτεται στους χρήστες που έλαβαν μέρος στη συναλλαγή. Οι πληροφορίες σε ένα τέτοιο δίκτυο μπορεί να είναι εμπιστευτικές και παράλληλα επαληθεύσιμες.

Πλεονεκτήματα: Οι συναλλαγές σε ένα υβριδικό blockchain είναι γρήγορες και το δίκτυο κλιμακώνεται με μεγαλύτερη ευκολία από ένα δημόσιο blockchain. Η ευελιξία της πρόσβασης στα δεδομένα δίνει τη δυνατότητα αξιοποίησης τέτοιων δικτύων για πολλές εφαρμογές.

Μειονεκτήματα: Τα βασικά μειονεκτήματα ενός ιδιωτικού δικτύου όπως η έλλειψη διαφάνειας, ανωνυμίας και ελέγχου του κώδικα παραμένουν, αν και περιορίζονται.

Χρήσεις: Τα υβριδικά blockchain προσφέρονται για πληθώρα χρήσεων. Μπορούν να αξιοποιηθούν από οργανισμούς που διαχειρίζονται ιατρικά δεδομένα χρηστών, από εταιρίες που ασχολούνται με την αγοραπωλησία ακινήτων αλλά και άλλες επιχειρήσεις.

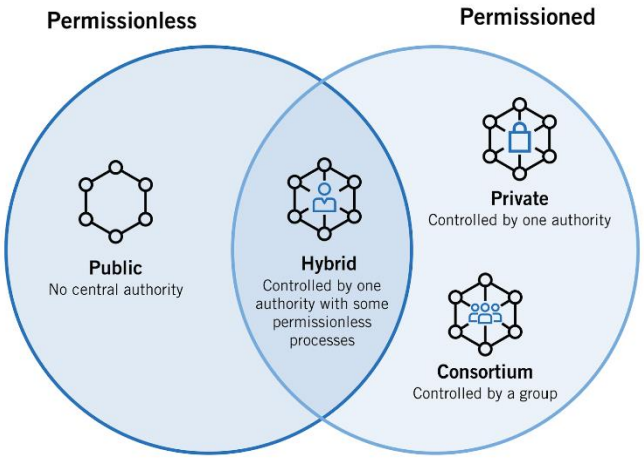
- **Blockchain Κοινοπραξίας (Consortium Blockchain)**

Το consortium Blockchain δανείζεται στοιχεία από τα δημόσια και τα ιδιωτικά blockchain. Πολλαπλές κεντρικές οντότητες ελέγχουν και ρυθμίζουν το δίκτυο. Οι διαδικασίες συναίνεσης πραγματοποιούνται από προεπιλεγμένους κόμβους, ενώ η πρόσβαση ελέγχεται από έναν ή περισσότερους κεντρικούς οργανισμούς ή κόμβους.

Πλεονεκτήματα: Το consortium blockchain διατηρεί τα πλεονεκτήματα του private blockchain αξιοποιώντας και στοιχεία του public blockchain, όπως ακριβώς και το hybrid blockchain. Διαφοροποιείται ωστόσο, ως προς τις περιπτώσεις χρήσης του.

Μειονεκτήματα: Το consortium blockchain υστερεί στη διαφάνεια σε σχέση με το δημόσιο blockchain. Επιπλέον, αν ένας σημαντικός κόμβος του δικτύου εμφανίσει ρήξη ασφάλειας, μπορεί να κινδυνέψει η λειτουργικότητα ολόκληρου του δικτύου.

Χρήσεις: Τραπεζικά συστήματα και συστήματα πληρωμών μπορούν να αξιοποιήσουν αυτού του τύπου blockchain. Για παράδειγμα, ένας αριθμός από τράπεζες μπορούν να σχηματίσουν μια κοινοπραξία και να αποφασίσουν ποιοι κόμβοι θα επαληθεύουν συναλλαγές. Ένα τέτοιο blockchain προσφέρεται και για ερευνητικούς οργανισμούς αλλά και για αλυσίδες εφοδιασμού.



Εικόνα 5: Κατηγορίες Blockchain [8]

2.1.4 Αλγόριθμος Συναίνεσης

Κάθε συναλλαγή που γίνεται στο blockchain θεωρείται έγκυρη και ασφαλής. Αυτό επιτυγχάνεται και με τη βοήθεια πρωτοκόλλου συναίνεσης (consensus protocol). Ο Αλγόριθμος συναίνεσης είναι μια διαδικασία κατά την οποία οι κόμβοι ενός blockchain φτάνουν σε κοινή συμφωνία σχετικά με την τωρινή κατάσταση του distributed ledger. Με αυτόν τον τρόπο εδραιώνεται εμπιστοσύνη και αξιοπιστία μεταξύ αγνώστων κόμβων σε ένα αποκεντρωμένο περιβάλλον.

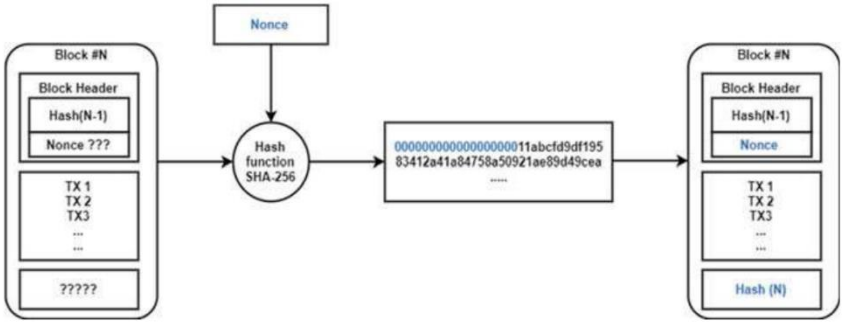
Στις εδραιωμένες αρχιτεκτονικές δικτύων, η συναίνεση αυτή δεν αποτελεί πρόβλημα, εξαιτίας της ύπαρξης ενός κεντρικού εξυπηρετητή (server). Σε αυτές τις αρχιτεκτονικές οι υπόλοιποι κόμβοι πρέπει απλά να είναι σύμφωνοι με τον server. Ωστόσο, σε ένα κατανεμημένο σύστημα όπως το blockchain, κάθε κόμβος είναι ταυτόχρονα πελάτης και εξυπηρετητής (client and server) και άρα πρέπει να υπάρχει επικοινωνία μεταξύ των κόμβων για να υπάρξει συναίνεση [10].

Συχνά κάποιοι κόμβοι του δικτύου μπορεί να είναι απενεργοποιημένοι. Ενδεχομένως να υπάρχουν στο δίκτυο κακόβουλοι κόμβοι με σκοπό να επηρεάσουν ή να ματαιώσουν πλήρως τη διαδικασία συναίνεσης (Βυζαντινά λάθη) [11]. Ως εκ τούτου, ο αλγόριθμος συναίνεσης πρέπει να σχεδιαστεί ώστε να εμφανίζει ανθεκτικότητα σε τέτοια φαινόμενα. Παράλληλα, θα πρέπει να σέβεται τον τύπο του εκάστοτε blockchain και να συμβαδίζει με τις ιδιαίτερες απαιτήσεις του. Οι επικρατέστεροι αλγόριθμοι συναίνεσης είναι:

Proof of Work (PoW)

Είναι ο παλαιότερος και πιο γνωστός αλγόριθμος συναίνεσης. Για να δημιουργηθεί ένα νέο block στο blockchain, πρέπει ένας κόμβος να αναλάβει τη δημιουργία και την προσθήκη του στο δίκτυο. Για την επιλογή του κόμβου αυτού πραγματοποιείται ένας διαγωνισμός επίλυσης ενός κρυπτογραφικού puzzle. Συγκεκριμένα, ο στόχος των κόμβων στα πλαίσια του διαγωνισμού αυτού είναι να βρουν έναν αριθμό ονόματι nonce ο οποίος όταν κρυπτογραφηθεί, σε συνδυασμό με τα δεδομένα του προς-δημιουργία-block, από μία συνάρτηση hash, δίνει μια προκαθορισμένη αλληλουχία (για παράδειγμα, μια αλληλουχία hash που έχει στην αρχή της τέσσερις φορές το ψηφίο μηδέν). Ο κόμβος που θα λύσει πρώτος το puzzle είναι αυτός που θα επικυρώσει τις συναλλαγές που έχουν πραγματοποιηθεί, θα δημιουργήσει το block και θα το προσθέσει στο δίκτυο. Οι υπόλοιποι κόμβοι του δικτύου μπορούν πολύ εύκολα να ελέγξουν αν πράγματι το nonce που βρήκε ο κόμβος αυτός είναι το σωστό. Αν αυτό ισχύει, τότε επικυρώνουν την είσοδο του νέου block στο blockchain. Ο στόχος της διαδικασίας αυτής είναι να εμποδίσει έναν συγκεκριμένο κόμβο να δημιουργήσει πολύ γρήγορα πολλά νέα block με λανθασμένα δεδομένα και να χειραγωγήσει έτσι το δίκτυο, διότι αυτό θα απαιτούσε τεράστια επεξεργαστική δύναμη από αυτόν τον κόμβο.

Ο αλγόριθμος συναίνεσης Proof of Work εγγυάται στατιστικά την συνέπεια του δικτύου. Ωστόσο, η διαδικασία εύρεσης του nonce απαιτεί επεξεργαστική δύναμη, ενέργεια και χρόνο τα οποία δεν αξιοποιούνται παρά μόνο για τον σκοπό αυτόν.

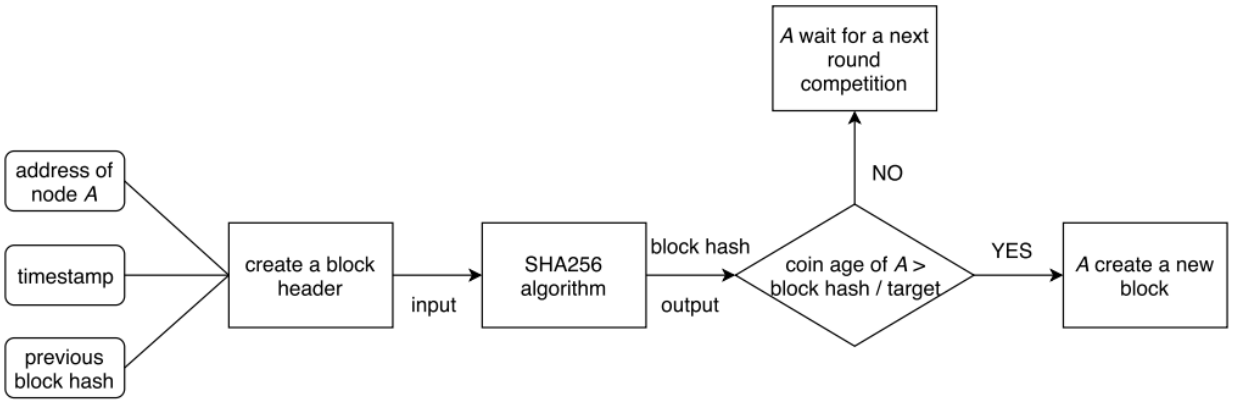


Εικόνα 6: PoW παραγωγή nonce [13]

Proof of Stake (PoS)

Ο αλγόριθμος συναίνεσης Proof of Stake εξασφαλίζει την ανθεκτικότητα σε Βυζαντινά λάθη και τη συνέπεια του δικτύου ως εξής: Κάθε κόμβος επενδύει ένα ποσό αξίας (για παράδειγμα ένα ποσό σε κρυπτονομίσματα) στο δίκτυο. Ανάλογα με το πόσο μεγάλο είναι το ποσό αυτό που

επενδύει, αυξάνεται και η πιθανότητα να είναι αυτός ο κόμβος ο οποίος θα επικυρώσει όλες τις συναλλαγές του δικτύου, θα δημιουργήσει και θα προσθέσει ένα νέο block στο blockchain. Ο κόμβος αυτός ονομάζεται κόμβος επικυρωτής (validator node). Αν αποδειχθεί ότι κάποιος κόμβος επικυρωτής επικύρωσε άκυρες συναλλαγές, τότε το stake που επένδυσε στο δίκτυο δεσμεύεται και ο κόμβος το χάνει. Υπάρχει έτσι κίνητρο για την δημιουργία αξιόπιστων νέων block, ενώ παράλληλα δεν απαιτούνται οι υπολογιστικοί και ενεργειακοί πόροι που απαιτούνται κατά την επικύρωση με proof of work.



Εικόνα 7: Ροή του PoS [14]

Proof of Authority (PoA)

Ο αλγόριθμος συναίνεσης Proof of Authority επιτρέπει μόνο σε προκαθορισμένες Αρχές (Authorities) να επικυρώνουν συναλλαγές και να προσθέτουν νέα block στο blockchain. Ορίζει κάποιους κόμβους ως επικυρωτές και τους δίνει το δικαίωμα να ανανεώνουν τον ledger και να παρέχουν ασφάλεια στο δίκτυο. Συνήθως εντοπίζεται σε private ή consortium blockchain και οι κεντρικές αρχές θεωρούνται έμπιστοι κόμβοι. Ο αλγόριθμος βασίζεται στη ταυτότητα των κόμβων επικυρωτών θεωρώντας ότι αυτοί δε θα συμπεριφερθούν κακόβουλα, ώστε να προστατέψουν τη φήμη τους. Προσφέρει υψηλή απόδοση, επεκτασιμότητα και αξιοπιστία ενώ ταυτόχρονα δεν απαιτεί κανέναν περίπλοκο υπολογισμό ή οικονομικό κίνητρο. Ωστόσο, θυσιάζει την αποκέντρωση και υπάρχει κίνδυνος έλλειψης διαφάνειας αν οι κεντρικές αρχές καταχραστούν τη δύναμή τους ή κρύψουν πληροφορίες.

Άλλοι γνωστοί αλγόριθμοι συναίνεσης είναι : Delegated Proof of Stake (DPoS), Practical Byzantine Fault Tolerance (PBFT), Proof of Elapsed Time(PoET), Proof of Burn (PoB), HotStuff.

Κάθε consensus algorithm έχει πλεονεκτήματα και μειονεκτήματα για αυτό πρέπει να γίνεται σωστή επιλογή ανάλογα με τις απαιτήσεις του κάθε συστήματος.

2.1.5 Hyperledger Fabric

Το Hyperledger Fabric είναι μια πλατφόρμα Τεχνολογίας Κατανεμημένου Λογιστικού Καταλόγου (DLT) ανοιχτού κώδικα. Έχει σχεδιαστεί για χρήση σε εταιρικά περιβάλλοντα και παρέχει ορισμένες βασικές δυνατότητες που το διαφοροποιούν από άλλες δημοφιλείς πλατφόρμες DLT ή blockchain.

Το Hyperledger ιδρύθηκε και λειτουργεί υπό την αιγίδα του Linux Foundation, το οποίο τυγχάνει μακράς και επιτυχούς ιστορίας στην ανάπτυξη λογισμικών ανοιχτού κώδικα. Εκτός αυτού, η διαχείριση και αναβάθμιση του Hyperledger Fabric γίνεται από ένα ποικίλο σύνολο συντηρητών από πολλούς οργανισμούς. Έχει μια κοινότητα προγραμματιστών που έχει αυξηθεί σε πάνω από 35 οργανισμούς και σχεδόν 200 προγραμματιστές από τότε που δημοσιεύτηκε για πρώτη φορά.[12]

Η πλατφόρμα Fabric αποτελεί επίσης μια υλοποίηση ενός δικτύου περιορισμένης πρόσβασης (permissioned network), πράγμα που σημαίνει ότι, σε αντίθεση με ένα δημόσιο δίκτυο (permissionless network), οι συμμετέχοντες είναι γνωστοί μεταξύ τους και όχι ανώνυμοι. Έτσι, ενώ οι συμμετέχοντες μπορεί να μην εμπιστεύονται πλήρως ο ένας τον άλλον (μπορεί, για παράδειγμα, να είναι ανταγωνιστές στον ίδιο κλάδο), ένα δίκτυο μπορεί να λειτουργήσει υπό ένα μοντέλο διακυβέρνησης.

Το Hyperledger Fabric έχει αρθρωτή αρχιτεκτονική με κομμάτια κώδικα που λειτουργούν άμεσα (plug-and-play modules) για την υλοποίηση διαφόρων αλγορίθμων συναίνεσης (consensus Algorithms) αλλά και λειτουργιών του Λογιστικού Καταλόγου (Ledger). Αυτό επιτρέπει στην πλατφόρμα να προσαρμόζεται εύκολα σε συγκεκριμένες περιπτώσεις χρήσης και μοντέλα εμπιστοσύνης. Για παράδειγμα, η χρήση ενός αλγορίθμου συναίνεσης ανθεκτικού σε βυζαντινά λάθη (byzantine fault tolerant consensus algorithm) θεωρείται συνήθως περιττή όταν εφαρμόζεται σε ένα δίκτυο εντός μιας εταιρείας. Επιβαρύνει, μάλιστα, την ταχύτητα των συναλλαγών. Σε μια τέτοια περίπτωση θα ήταν προτιμότερο να χρησιμοποιηθεί ένας αλγόριθμος συναίνεσης ανθεκτικός σε αποτυχία (crash fault-tolerant ή CFT consensus algorithm). Αντίθετα, ο byzantine fault tolerant consensus (BFT) αλγόριθμος, θα ήταν δόκιμο να χρησιμοποιηθεί σε ένα αποκεντρωμένο δίκτυο στο οποίο συμμετέχουν πολλές επιχειρήσεις ή οντότητες (multi-party decentralized network).

Έξυπνο Συμβόλαιο (Smart Contract)

Για να αρχίσουν οι επιχειρήσεις να συναλλάσσονται μεταξύ τους, πρέπει να ορίσουν ένα κοινό σύνολο συμβάσεων που να καλύπτει κοινούς όρους, δεδομένα, κανόνες, έννοιες και διαδικασίες. Συνολικά, αυτές οι συμβάσεις (συμβόλαια) καθορίζουν το επιχειρηματικό μοντέλο που διέπει όλες τις αλληλεπιδράσεις μεταξύ των επιχειρήσεων.

Ένα έξυπνο συμβόλαιο (smart contract) ορίζει αυτούς τους κανόνες μεταξύ διάφορων οργανισμών με τη μορφή εκτελέσιμου κώδικα. Οι εφαρμογές blockchain ενεργοποιούν ένα έξυπνο συμβόλαιο ώστε να εκτελεστεί μια συναλλαγή και να γραφτεί στον κατανεμημένο κατάλογο.

Το Fabric είναι η πρώτη πλατφόρμα κατανεμημένου καθολικού λογισμικού που υποστηρίζει έξυπνα συμβόλαια γραμμένα σε γενικές γλώσσες προγραμματισμού όπως Java, Go και Node.js, αντί για γλώσσες ειδικού τομέα (Domain Specific Languages ή DSL). Αυτό προσφέρει ευελιξία στην ανάπτυξη και τη χρήση του.

Κώδικας Αλυσίδας (Chaincode)

Συχνά στα πλαίσια της πλατφόρμας Hyperledger Fabric χρησιμοποιείται και ο όρος chaincode αντί του έξυπνου συμβολαίου. Εν γένει, τα έξυπνα συμβόλαια ορίζουν την λογική της προς εκτέλεση συναλλαγής μεταξύ δύο κόμβων του δικτύου Blockchain. Έπειτα, ένα ή περισσότερα έξυπνα συμβόλαια “πακετάρονται” σε ένα chaincode το οποίο εντάσσεται στο δίκτυο blockchain. Δηλαδή, τα συμβόλαια ελέγχουν τις συναλλαγές, ενώ το chaincode ελέγχει το πως τα συμβόλαια στοιβάζονται για να εκτελεστούν.

Endorsement

Κάθε κώδικας αλυσίδας (chaincode) συνοδεύεται πάντα από μια πολιτική αποδοχής (endorsement policy). Η πολιτική αποδοχής είναι σημαντική, διότι καθορίζει το ποιοι οργανισμοί ή κόμβοι, σε ένα δίκτυο blockchain, πρέπει να υπογράψουν μια συναλλαγή εντός ενός έξυπνου συμβολαίου, ώστε να θεωρηθεί αυτή η συναλλαγή έγκυρη.

Μια πολιτική αποδοχής θα μπορούσε για παράδειγμα να ορίζει ότι για να θεωρηθεί μια συναλλαγή έγκυρη θα πρέπει τρεις στους τέσσερις οργανισμούς που συμμετέχουν στο δίκτυο blockchain να την υπογράψουν. Να σημειωθεί ότι όλες οι συναλλαγές, έγκυρες ή μη, σημειώνονται στον κατανεμημένο λογιστικό κατάλογο, αλλά μόνο οι έγκυρες συναλλαγές προκαλούν αλλαγή στην κατάσταση των κόμβων (world state).

Οι πολιτικές αποδοχής είναι ένας σημαντικός παράγοντας διαφοροποίησης του Hyperledger Fabric από άλλα blockchain όπως το Ethereum ή το Bitcoin. Το Hyperledger Fabric αποτελεί ένα πιο ακριβές μοντέλο του πραγματικού κόσμου των επιχειρήσεων, καθώς μπορεί να ικανοποιήσει την ανάγκη για υπογραφή μιας συναλλαγής από πολλούς οργανισμούς. Στην περίπτωση των υπόλοιπων blockchain δικτύων κάτι τέτοιο δεν μπορεί να υλοποιηθεί αφού κάθε συναλλαγή μπορεί να υπογράφει από το πολύ μια οντότητα ή κόμβο.
Η πολιτική αποδοχής (endorsement policy) είναι μια από τις πολλές πολιτικές συναλλαγών που μπορούν να υλοποιηθούν στα πλαίσια του Hyperledger Fabric, καθώς η πλατφόρμα παρέχει ευελιξία και στον τομέα αυτόν.

2.2 Απόδειξη μηδενικής Γνώσης (ZKP)

Οι Αποδείξεις Μηδενικής Γνώσης (Zero Knowledge Proofs) είναι κρυπτογραφικά πρωτόκολλα τα οποία χρησιμοποιούνται για να αποδείξουν την εγκυρότητα μιας πρότασης χωρίς να αποκαλύψουν κανένα δεδομένο για την ίδια την πρόταση.

Οι επιστήμονες του MIT Shafi Goldwasser, Silvio Micali και Charles Rackoff πρότειναν πρώτοι την ιδέα της απόδειξης μηδενικής γνώσης τη δεκαετία του 1980, δείχνοντας ότι είναι δυνατό κανείς να αποδείξει ότι κάποια θεωρήματα είναι αληθινά χωρίς να δώσει την παραμικρή ένδειξη για το ποια ακριβώς είναι αυτά τα θεωρήματα [14].

Ένα πρωτόκολλο Απόδειξης Μηδενικής Γνώσης πρέπει να πληροί τις παρακάτω προϋποθέσεις [15]:

- **Πληρότητα** (Completeness): Αν η υποκείμενη πρόταση είναι αληθής, ο επαληθευτής (verifier) μπορεί να επιβεβαιώσει ή να πειστεί για την ειλικρίνεια της αποδεικνύουσας οντότητας (prover).
- **Σταθερότητα** (Soundness): Αν η υποκείμενη πρόταση είναι άκυρη, είναι πρακτικά απίθανο το πρωτόκολλο να επιστρέψει “αλήθεια”. Δηλαδή, μια ψευδή αποδεικνύουσα οντότητα δεν μπορεί να πείσει τον επαληθευτή ότι η πρότασή της είναι αληθής (πέραν μιας πολύ μικρής πιθανότητας).
- **Μηδενική Γνώση** (Zero Knowledge): Ο επαληθευτής δε γνωρίζει τίποτα περισσότερο από την εγκυρότητα ή μη της πρότασης.

Τα πρωτόκολλα ZKP χωρίζονται σε δύο μεγάλες κατηγορίες: ZKP-αλληλεπίδρασης και ZKP-μη αλληλεπίδρασης

2.2.1 Είδη ZKP

Τα πρώτα ZKP πρωτόκολλα ήταν ZKP-αλληλεπίδρασης (interactive ZKP) που σημαίνει ότι η επικύρωση της εγκυρότητας της υποκείμενης πρότασης απαιτούσε την επαναλαμβανόμενη επικοινωνία (back and forth communication) μεταξύ του επαληθευτή και της αποδεικνύουσας οντότητας. Τα ZKP-αλληλεπίδρασης είναι πιο αποδοτικά σε δίκτυα με λιγότερους συμμετέχοντες.

Τα πλέον διαδεδομένα πρωτόκολλα ZKP είναι τα ZKP-μη αλληλεπίδρασης (non-interactive zero knowledge proof). Παρακάτω τρία από τα πιο διαδεδομένα.

Bulletproofs

Τα Bulletproofs είναι πρωτόκολλα απόδειξης μηδενικής γνώσης τα οποία έχουν ως στόχο να αποδείξουν μια υποκείμενη πρόταση που αποτελεί ισχυρισμό για το εύρος μιας τιμής, χωρίς να αποκαλύψουν τίποτα περισσότερο για την τιμή αυτήν. Τα Bulletproofs δεν είναι υποχρεωτικό να λειτουργούν σε δίκτυο όπου επικρατεί εμπιστοσύνη μεταξύ των κόμβων και παράγουν μικρές αποδείξεις από πλευράς μνήμης. Ωστόσο, η επικύρωση μιας απόδειξης Bulletproof είναι ακριβή υπολογιστικά σε σχέση με άλλα πρωτόκολλα [16].

Zk-SNARKs

Τα zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Argument of Knowledge) έχουν μικρές περιεκτικές αποδείξεις (proofs), οι οποίες μπορούν να επιβεβαιωθούν με μεγάλη ταχύτητα. Περιέχουν τρεις κύριες φάσεις και ορίζονται ως εξής [17] :

Δεδομένης μιας σχέσης *R*, ένα zk-SNARK αποτελείται από ένα σύνολο αλγορίθμων Π_{snark} = (Προεργασία, Απόδειξη, Επικύρωση Απόδειξης) ή (Setup, Prove, VerProof) το οποίο διέπεται από την παρακάτω λειτουργία:

- Setup(λ, *R*)$\rightarrow$*crs* := (ek,vk), td: Ο αλγόριθμος παίρνει μια παράμετρο ασφάλειας λ και μια σχέση *R* σαν είσοδο και επιστρέφει μια κοινή συμβολοσειρά αναφοράς *crs* η οποία

περιέχει ένα κλειδί αξιολόγησης (evaluation key) ek και ένα επικυρωτικό κλειδί (verification key) vk καθώς και μια παγίδα προσομοίωσης (simulator trapdoor) td.

- Prove(ek,x,w)$\rightarrow\pi$: Ο αλγόριθμος παίρνει ως είσοδο ένα κλειδί αξιολόγησης ek, μια πρόταση x και έναν μάρτυρα (witness) w έτσι ώστε (x, w) $\in R$. Ως έξοδο επιστρέφει μια απόδειξη π. (witness ονομάζουμε το στοιχείο του οποίου την ύπαρξη και γνώση θέλει να αποδείξει ο prover.)
- VerProof(vk, x, π) $\rightarrow$ true/false: Ο αλγόριθμος παίρνει ένα verification key vk, μια πρόταση x και μια απόδειξη π σαν είσοδο και επιστρέφει αλήθεια (true) αν η απόδειξη είναι ορθή ή ψεύδος (false) αν η απόδειξη είναι λανθασμένη.

Οι αλγόριθμοι χρησιμοποιούνται από οντότητες που τις ονομάζουμε provers και verifiers. Ένας prover, προσπαθεί να αποδείξει ότι υπάρχει ένα witness w, το οποίο γνωρίζει και το οποίο ικανοποιεί μια σχέση R δεδομένου ενός x. Το w είναι μυστικό και ο prover δεν θέλει να αποκαλύψει τίποτα για αυτό στον verifier. Ο verifier λαμβάνει μια απόδειξη από τον prover και επικυρώνει την ορθότητά της

Τα zk-SNARKs είναι πολύ αποτελεσματικά και γρήγορα καθώς χάρη στην προεργασία (Setup) ο verifier μπορεί πολύ γρήγορα να επικυρώσει τις αποδείξεις του prover.

Η διαδικασία δημιουργίας των κλειδιών ek και vk πρέπει να λάβει χώρα σε ένα ασφαλές περιβάλλον (**trusted setup ceremony**). Κατά τη διαδικασία αυτή, χρησιμοποιούνται κάποιες μυστικές παράμετροι, οι οποίες αν εκτεθούν σε κάποιον κακόβουλο prover, τότε αυτός θα μπορούσε να κατασκευάζει αποδείξεις που είναι επικυρώσιμες χωρίς να είναι πράγματι ορθές. Ως εκ τούτου, μετά τη διαδικασία δημιουργίας κλειδιών η οποία είναι διασφαλισμένη, οι μυστικές παράμετροι που χρησιμοποιήθηκαν πρέπει να καταστραφούν.

zk-STARKs

Τα zk-STARKs (Zero-Knowledge Scalable Transparent Argument of Knowledge) είναι μια εναλλακτική για τα zk-SNARKs που προτάθηκε πρώτη φορά το 2018 [18]. Τα zk-STARKs

μπορούν να προσφέρουν μεγαλύτερη ασφάλεια από τα zk-SNARKs, καθώς δεν απαιτούν περιβάλλον εμπιστοσύνης (trusted setup) για τη δημιουργία κλειδιών.

Εντούτοις, τα zk-STARKs παράγουν μεγαλύτερα μεγέθη αποδείξεων και οι verifiers χρειάζονται περισσότερο χρόνο για την επικύρωση των αποδείξεων από ότι στα zk-SNARK. Αξίζει να σημειωθεί ότι και τα δύο αυτά πρωτόκολλα βρίσκονται στην αιχμή της τεχνολογίας και η επιλογή του ενός έναντι του άλλου εξαρτάται άμεσα από την εκάστοτε εφαρμογή αλλά και τις επιστημονικές εξελίξεις στον σχετικό τομέα της κρυπτογραφίας.

2.2.2 Circom

Η Circom είναι μια γλώσσα προγραμματισμού για την κατασκευή αριθμητικών κυκλωμάτων και ο μεταγλωττιστής (compiler) της είναι γραμμένος σε Rust.

Όπως τα περισσότερα πρωτόκολλα ZKP, έτσι και τα zk-SNARKs, πραγματεύονται αποδείξεις που σχετίζονται με υπολογιστικές προτάσεις ή σχέσεις R όπως αναφέρθηκε παραπάνω. Οι σχέσεις αυτές πρέπει να εκφραστούν με την κατάλληλη μορφή για να εφαρμοστούν τα πρωτόκολλα. Τα zk-SNARKS, συγκεκριμένα, απαιτούν την έκφραση των σχέσεων αυτών με τη μορφή αριθμητικών κυκλωμάτων (arithmetic circuits).

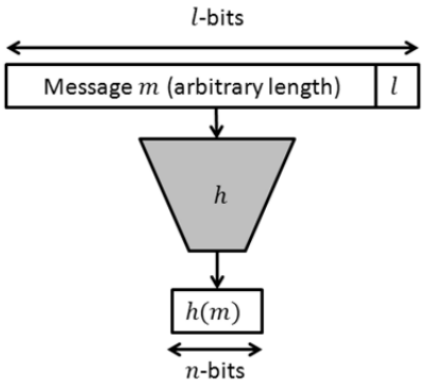
Ένα F_p αριθμητικό κύκλωμα είναι ένα κύκλωμα που αποτελείται από ένα σύνολο καλωδίων (wires) τα οποία φέρουν τιμές από το πεδίο F_p και τις συνδέουν με πύλες (gates) πρόσθεσης και πολλαπλασιασμού. Το αποτέλεσμα διαιρείται με έναν πρώτο αριθμό p και λαμβάνεται το υπόλοιπο (modulo p).

Δεδομένου ενός πρώτου αριθμού p, το πεδίο F_p είναι το πεπερασμένο πεδίο που αποτελείται από το σύνολο αριθμών {0,..., p-1} στο οποίο μπορούμε να προσθέτουμε και να πολλαπλασιάζουμε αυτούς τους αριθμούς αν έπειτα υπολογίσουμε το υπόλοιπο της διαίρεσης του αποτελέσματος με το p.

Με τη γλώσσα circom, ο σχεδιαστής ενός zk-SNARK μπορεί να κατασκευάσει μεγάλα αριθμητικά κυκλώματα συνδυάζοντας μικρά έτοιμα κυκλώματα που ονομάζονται πρότυπα κυκλώματα (templates).

2.3 Συναρτήσεις Κατακερματισμού

Ο όρος συνάρτηση κατακερματισμού (hash function) υποδηλώνει έναν μετασχηματισμό που παίρνει σαν είσοδο ένα μήνυμα m οποιουδήποτε μήκους και επιστρέφει στην έξοδο μία ακολουθία χαρακτήρων h περιορισμένου μήκους που καλείται hash value, δηλαδή είναι h = H(m).



Εικόνα 8: Συνάρτηση Κατακερματισμού [20]

2.3.1 HMAC SHA256 Hash

Το **HMAC** (Hash-based Message Authentication Code ή Κωδικός Πιστοποίησης Μηνύματος βασισμένος σε συνάρτηση HASH) είναι ένας αλγόριθμος πιστοποίησης μηνυμάτων που χρησιμοποιείται ευρέως σε δίκτυα όπου απαιτείται ακεραιότητα και ασφάλεια. Τα HTTPS, SFTP, FTPS, και άλλα γνωστά πρωτόκολλα δικτύων αξιοποιούν το HMAC.

Ο αλγόριθμος βασίζεται στην μη αντιστρεψιμότητα των συναρτήσεων κατακερματισμού (hash functions). Όλοι οι κόμβοι του δικτύου διαθέτουν ένα κλειδί (shared key) το οποίο διατηρείται μυστικό και δεν μοιράζεται με κόμβους που δεν ανήκουν στο δίκτυο. Κάθε μήνυμα που στέλνεται ανάμεσα σε δύο κόμβους ενώνεται - με τρόπο που θα αναλυθεί στη συνέχεια - με το shared key και περνάει από τη συνάρτηση Hash SHA256. Το αποτέλεσμα της συνάρτησης Hash είναι μια συμπιεσμένη μορφή του συνδυασμού μηνύματος και shared key.

Πιο συγκεκριμένα ο αλγόριθμος HMAC-SHA256 ορίζεται ως εξής [13]:

$$HMAC(K,m) = H((K \oplus opad) \parallel H((K \oplus ipad) \parallel m))$$

με τις παραμέτρους να ορίζονται ως:

H = cryptographic hash function = SHA256

K = secret key

m = message

∥ = concatenation

⊕ = exclusive OR

opad = outer padding

ipad = inner padding

Ένα βασικό πλεονέκτημα των HMAC είναι ότι είναι ιδανικά για δίκτυα που απαιτούν υψηλή ταχύτητα ανταλλαγής μηνυμάτων (όπως για παράδειγμα οι δρομολογητές (routers)), εξαιτίας του γρήγορου υπολογισμού αλλά και της επαλήθευσης των αποτελεσμάτων της συνάρτησης hash.

Το κύριο μειονέκτημά τους είναι ότι αν το shared key διαρρεύσει με κάποιον τρόπο σε κάποιον εξωτερικό κόμβο, τότε αυτός μπορεί να δημιουργεί και να επαληθεύει μηνύματα χωρίς πραγματική εξουσιοδότηση.

Το “256” στο SHA256 αναφέρεται στο μέγεθος λέξης που χρησιμοποιείται κατά τη δημιουργία του συμπιεσμένου αποτελέσματος, το οποίο είναι 32-byte.

2.3.2 MiMC Hash

Το MiMC Hash (Minimal Multiplicative Complexity Hash) βασίζεται στην ιδέα της εφαρμογής μεταθέσεων (permutations) στα δεδομένα εισόδου. Οι μεταθέσεις υλοποιούνται με αποτελεσματικό τρόπο (με μικρό αριθμό πολλαπλασιασμών), πράγμα που κάνει το MiMC Hash να εφαρμόζεται αποδοτικά σε επίπεδο hardware και software. Πολλές άλλες συναρτήσεις κατακερματισμού τελευταίας τεχνολογίας βασίζονται στην συνάρτηση κατακερματισμού MiMC.

Το κύριο στοιχείο του MiMC είναι η συνάρτηση APN $F(x) = x^3$ [24]. Η συνάρτηση υπολογίζεται στο πεδίο $\mathbb{F}_q$, όπου q = p ή q = 2^n για έναν πρώτο αριθμό p και έναν φυσικό αριθμό n.

Λεπτομερώς, η κρυπτογραφική συνάρτηση MiMC ορίζεται ακολούθως:

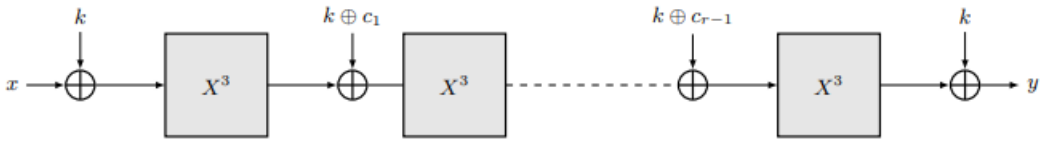
$$E_k(x) = (F_{r-1} \circ F_{r-2} \circ \dots \circ F_0)(x) + k,$$

με $x \in \mathbb{F}_q$, r ο αριθμός των γύρων, F_i είναι η συνάρτηση των γύρων για τον γύρο $i \geq 0$ και $k \in \mathbb{F}_q$

Το F_i ορίζεται ως:

$$F_i(x) = (x + k + c_i)^3$$

με $c_i \in \mathbb{F}_q$ οι σταθερές των γύρων και $c_0 = 0$.



Εικόνα 9: MiMC hash[24]

2.3.3 Pedersen Hash

Η συνάρτηση Pedersen Hash είναι μια συνάρτηση Hash που λειτουργεί με ένα παράθυρο μεγέθους 3-bit ή 4-bit και αντιστοιχίζει μια αλληλουχία από bits σε ένα συμπιεσμένο σημείο μιας ελλειπτικής καμπύλης. Ο υπολογισμός της από μια αλληλουχία bits με χρήση ενός αριθμητικού κυκλώματος μπορεί να χρησιμοποιηθεί μέσα σε zero knowledge proofs.

Η συνάρτηση Pedersen για M (ακολουθία M bits) ορίζεται ως εξής [26]:

$$H(M) = \langle M_0 \rangle \cdot P_0 + \langle M_1 \rangle \cdot P_1 + \langle M_2 \rangle \cdot P_2 + \cdots + \langle M_l \rangle \cdot P_l$$

Με $P_0 P_1, \dots, P_k$ να είναι ομοιόμορφα δειγματοληπτημένες γεννήτριες του χώρου G (ο χώρος G είναι ένα σύνολο σημείων που ορίζεται μέσω της καμπύλης Baby-Jubjub) για κάποιο ακέραιο k και έχοντας χωρίσει το M σε ακολουθίες μεγέθους το πολύ 200 bits και κάθε μία από αυτές σε κομμάτια των 4 bits.

2.3.4 Poseidon Hash

Η Poseidon Hash είναι μια συνάρτηση κατακερματισμού που έχει σχεδιαστεί με γνώμονα τη χρήση της σε συστήματα απόδειξης μηδενικής γνώσης (ZKP). Στόχος της σχεδίασης της Poseidon Hash ήταν να προσφέρει ταχύτητα και ασφάλεια διατηρώντας τα επιθυμητά χαρακτηριστικά των μέχρι τότε δημοφιλών συναρτήσεων κατακερματισμού. Πράγματι, η Poseidon Hash είναι συνάρτηση απορρόφησης (sponge functions) που σημαίνει ότι παίρνει ως είσοδο ένα μήνυμα αυθαίρετου μεγέθους και δίνει έξοδο σταθερού μεγέθους. Επιπλέον μπορεί να χρησιμοποιηθεί αποτελεσματικά για μεγάλο εύρος μεγεθών εισόδου πράγμα που της δίνει χαρακτηριστική ευελιξία χρήσης. Οι τελευταίες εκδόσεις του Poseidon Hash αλγορίθμου δανείζονται στοιχεία από τη μεθοδολογία HadesMIMC hash που δημοσιεύτηκε το 2020.

Πιο συγκεκριμένα, ο αλγόριθμος Poseidon$^{\pi}$ Permutation στον οποίο βασίζονται οι τελευταίες βελτιστοποιημένες υλοποιήσεις του Poseidon Hash ορίζεται ως εξής [25]:

Έστω $p > 2^{30}$ ένας πρώτος αριθμός και έστω $t \geq 2$. Το Poseidon$^{\pi}$ permutation P πάνω στο πεδίο F_p^t είναι:

$$P(x) = \mathcal{E}_{R_F-1} \circ \cdots \circ \mathcal{E}_{R_F/2} \circ I_{R_P-1} \circ \cdots \circ I_0 \circ \mathcal{E}_{\frac{R_F}{2}-1} \circ \cdots \circ \mathcal{E}_0(x),$$

όπου $\mathcal{E}$ είναι ένας εξωτερικός (πλήρης) γύρος (permutation round), I είναι ένας εσωτερικός (μερικός) γύρος, R_f είναι ο αριθμός των εξωτερικών γύρων και R_P είναι ο αριθμός των εσωτερικών γύρων. Για επίπεδο ασφαλείας k bits οι παράμετροι αυτοί επιλέγονται με καθορισμένο τρόπο που περιγράφεται στην αντίστοιχη δημοσίευση.

2.4 Representational State Transfer (Rest)

Το REST είναι ένας γνώμονας, μια σειρά δηλαδή από κανόνες για την υλοποίηση της αρχιτεκτονικής ενός συστήματος υπερμέσων [21]. Προτάθηκε για να καθοδηγήσει τη σχεδίαση και την ανάπτυξη εφαρμογών του Παγκόσμιου Ιστού. Εντούτοις, οποιαδήποτε εφαρμογή υπόκειται στους κανόνες που προτείνει το REST θεωρείται RESTful και πολλές φορές γίνεται αναφορά σε τέτοιες εφαρμογές με τον όρο RESTful APIs.

Εν συντομία οι χαρακτηριστικοί κανόνες της αρχιτεκτονικής REST είναι οι ακόλουθοι:

- **Διευθυνσιοδότηση Πόρων** (Resource addressability): Για την αναγνώριση του κάθε πόρου χρησιμοποιούνται Uniform Resource Identifiers (URIs).

- **Αναπαράσταση Πόρων** (Resource representations) Οι πόροι έχουν συγκεκριμένη δομή και μορφή (συνήθως τύπου JSON ή XML) επομένως η επεξεργασία τους γίνεται με συγκεκριμένα πρωτόκολλα.

- **Ομοιόμορφη Διεπαφή** (Uniform Interface): Η πρόσβαση στους πόρους αλλά και η επεξεργασία τους γίνεται μέσω των πρότυπων μεθόδων που ορίζονται από το πρωτόκολλο HTTP. Οι μέθοδοι αυτές είναι: Post, Get, Put, Delete, Options και Head.

- **Έλλειψη Κατάστασης** (Statelessness): Οι αλληλεπιδράσεις μεταξύ κάποιου που αιτείται κάποια πράξη από το API και του ίδιου του API, δεν απαιτούν την πληροφορία κάποιας κατάστασης. Δηλαδή, οι αιτήσεις του αιτούντος είναι πλήρεις και περιέχουν όλες τις πληροφορίες που χρειάζονται για να τις επεξεργαστεί το API.

- **Χρήση Υπερμέσων** (Hypermedia utilization): Οι πόροι συνδέονται μεταξύ τους μέσω υπερμέσων.

Για την ανάπτυξη και την αξιολόγηση ενός RESTful API είναι συχνό να χρησιμοποιούνται εργαλεία που προσομοιώνουν έναν πελάτη (client) ώστε να δημιουργηθούν διεπαφές πελάτη-API. Ένα δημοφιλές τέτοιο εργαλείο είναι το Postman.

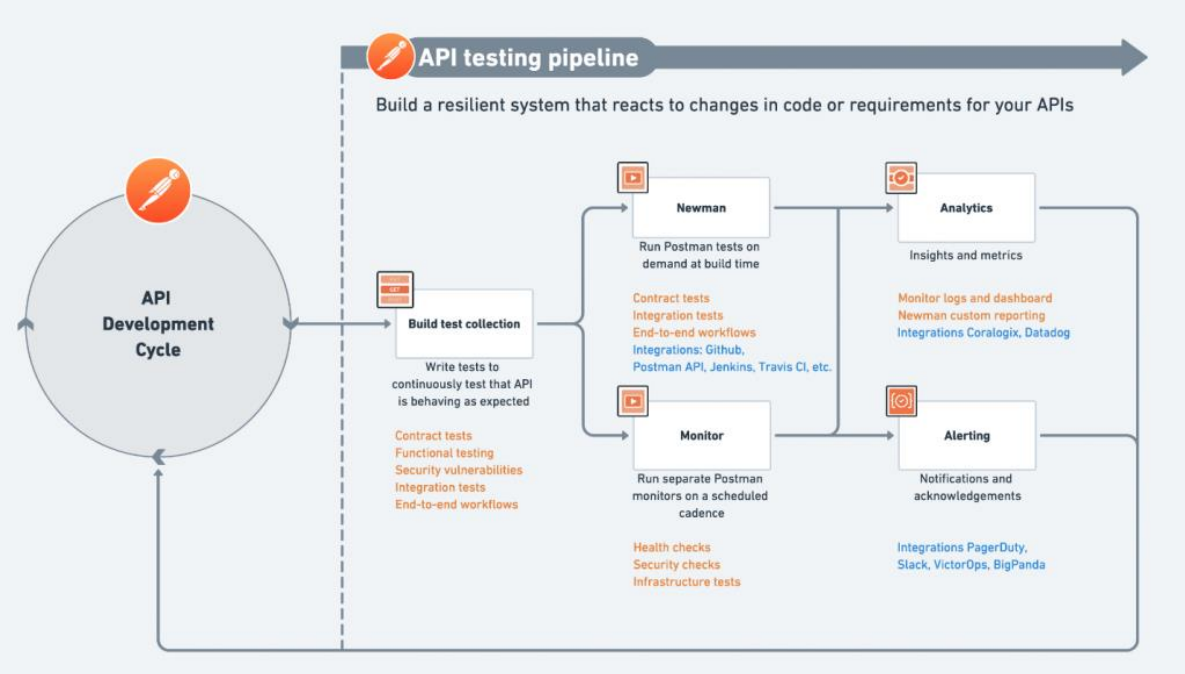
2.4.1 Postman

Το Postman είναι μια πλατφόρμα API (Application Programming Interface), η οποία έχει σχεδιαστεί για την βελτίωση, την απλούστευση και την επιτάχυνση της διαδικασίας κατασκευής ενός API [22]. Αυτό επιτυγχάνεται μέσω των παρακάτω λειτουργιών που προσφέρει το Postman:

- **Εργαλεία Ανάπτυξης:** Το βασικότερο εργαλείο που προσφέρει η πλατφόρμα είναι το Postman API client. Μέσω αυτού, το Postman προσομοιώνει έναν client ο οποίος μπορεί να ενεργοποιεί τη διεπαφή με πόρους του υπό ανάπτυξη API στέλνοντας τις πρότυπες μεθόδους του HTTP (Post, Get, Put, Delete, Options, Head), αλλά και άλλων πρωτοκόλλων αν είναι επιθυμητό, σε οποιοδήποτε URI επιθυμεί ο προγραμματιστής.

Αυτό επιτρέπει στον προγραμματιστή να ελέγξει το API του και να εποπτεύσει την συμπεριφορά των πόρων που έχει κατασκευάσει ανάλογα με την πρότυπη μέθοδο που αυτοί δέχονται. Τα υπόλοιπα εργαλεία που προσφέρει το Postman αφορούν την επιτήρηση επίδοσης του υπό ανάπτυξη API (API Monitoring), τη διευκόλυνση στο σχεδιασμό με σχηματικούς εκδότες (schema editors) αλλά και την αξιοποίηση ψευδο-εξυπηρετητών (mock-servers).

- **Αποθήκευση Δεδομένων:** Όντας εκτός άλλων ένα αποθετήριο API (API repository), το Postman επιτρέπει στον προγραμματιστή να αποθηκεύσει τη δομή και τα χαρακτηριστικά, τις οδηγίες χρήσης, τα εκάστοτε αποτελέσματα αλλά και τις επιδόσεις του API που κατασκευάζει.
- **Χώροι Εργασίας (Workspaces):** Το postman επιτρέπει τον διαχωρισμό των υποέργων ανάλογα με τις ανάγκες των προγραμματιστών που τα αναπτύσσουν, διευκολύνοντας έτσι την οργάνωση μεγάλων έργων.
- **Ενσωματώσεις (Integrations):** Η ευρεία χρήση του Postman, καθώς και το γεγονός ότι πρόκειται για μια πλατφόρμα ανοιχτού κώδικα, έχει οδηγήσει στην ενσωμάτωση πληθώρας τεχνολογιών και πακέτων λογισμικού με τα οποία οι περισσότεροι προγραμματιστές είναι ήδη εξοικειωμένοι, διευκολύνοντας και επιταχύνοντας έτσι την ανάπτυξη API.



Εικόνα 10: Η διαδικασία ελέγχου και ανάπτυξης API με το Postman [23]

2.5 Σχετικές Εργασίες

Όσο αναπτύσσεται η τεχνολογία, τόσο σημαντικότερη γίνεται και η διασφάλιση των προσωπικών στοιχείων. Σύμφωνα με τις Tyagi και Kathuria [27] η ανωνυμία των χρηστών ενός δικτύου δεν συνεπάγεται πλήρη ιδιωτικότητα. Το ζήτημα αυτό έχει γίνει αντιληπτό αλλά παρ' όλα αυτά δεν διευθετείται στα περισσότερα πρωτόκολλα Blockchain. Η σημασία του αυξάνεται με την πάροδο του χρόνου, καθώς η ανάγκη για αποθήκευση δεδομένων σε δίκτυα μεγαλώνει. Η Απόδειξη Μηδενικής Γνώσης (Zero Knowledge Proof) ως τεχνολογία αποτελεί την πιο άμεση λύση σε αυτό το πρόβλημα, μιας και σε θεωρητικό επίπεδο επιτρέπει την αξιοπιστία ενός δικτύου χωρίς να εκθέτει παραμέτρους ή δεδομένα που είναι μυστικά στα πλαίσια της ιδιωτικότητας.

Ένας τομέας στον οποίο η τήρηση της ιδιωτικότητας παρουσιάζει αυξημένη σημασία είναι αυτός της υγειονομικής περίθαλψης. Οι περισσότεροι οργανισμοί παροχής υπηρεσιών υγείας χρησιμοποιούν κεντροποιημένα συστήματα διαχείρισης ταυτότητας (centralized identity management systems - IDMs) τα οποία περιορίζουν την ανταλλαγή δεδομένων μεταξύ ινστιτούτων υγείας και δημιουργούνται έτσι απομονωμένες νησίδες δεδομένων, ή υπάρχει κίνδυνος διαρροής δεδομένων. Οι Tianyu Bai, Yangsheng Hu κ.α.[28] πρότειναν το Health-zkIDM, ένα αποκεντρωμένο σύστημα επικύρωσης ταυτότητας. Αυτό βασίζεται στην τεχνολογία του Blockchain και του ZKP, επιτρέποντας έτσι στους ασθενείς να ταυτοποιούν και να επικυρώνουν τα αναγνωριστικά τους ξεχωριστά και με ασφάλεια σε διαφορετικά πεδία υγειονομικού ενδιαφέροντος, ενώ παράλληλα προωθεί την αλληλεπίδραση μεταξύ των παροχων συστημάτων διαχείρισης ταυτότητας και των ίδιων των ασθενών. Χρησιμοποιήθηκε το Fabric Blockchain και εφαρμόστηκαν πρωτόκολλα ZKP ώστε η επικύρωση της εγγραφής των ασθενών στο δίκτυο να επιτελείται με επιτυχία και σε ικανοποιητικές ταχύτητες, χωρίς να διαρρέουν στοιχεία για την ταυτότητά τους.

Προσεγγίζοντας το ίδιο πρόβλημα, οι Gweonho Jeong, Nuri Lee κ.α. [17] αναφέρουν ότι οι δημόσιες πλατφόρμες Blockchain όπως το Bitcoin και το Ethereum παραβιάζουν κανονισμούς που τίθενται εντός του προτύπου EU GDPR, αφού οι δραστηριότητες των λογαριασμών των χρηστών είναι δημόσιες. Επιπλέον, η μυστικότητα των συναλλαγών μπορεί να προσφέρει ιδιωτικότητα, αλλά εμποδίζει τη διαφάνεια και την επιτήρηση των συναλλαγών.

Οι ερευνητές προτείνουν το σύστημα Azeroth, ένα λογισμικό το οποίο αξιοποιώντας το πρωτόκολλο ZKP επιτρέπει την επιτήρηση των συναλλαγών στο δίκτυο, χωρίς να εκθέτει στοιχεία των χρηστών.

Ενδιαφέρον παρουσιάζει επίσης η έρευνα πάνω στην επιτάχυνση των πρωτοκόλλων ZKP τα οποία δεν απαιτούν ένα περιβάλλον εμπιστοσύνης (trusted setup) για τη λειτουργία τους. Οι Shang Gao, Zhe Peng κ.α. προτείνουν το SymmeProof [29], ένα εργαλείο που καταφέρνει να παράξει μια απόδειξη μεγέθους log(n) εισάγοντας και ενσωματώνοντας δύο νέες τεχνικές: την συμπίεση διανυσμάτων (vector compression) και μια απόδειξη εσωτερικού γινομένου (inner-product range proof).

Κεφάλαιο 3

Αρχιτεκτονική Πληροφοριακού Συστήματος

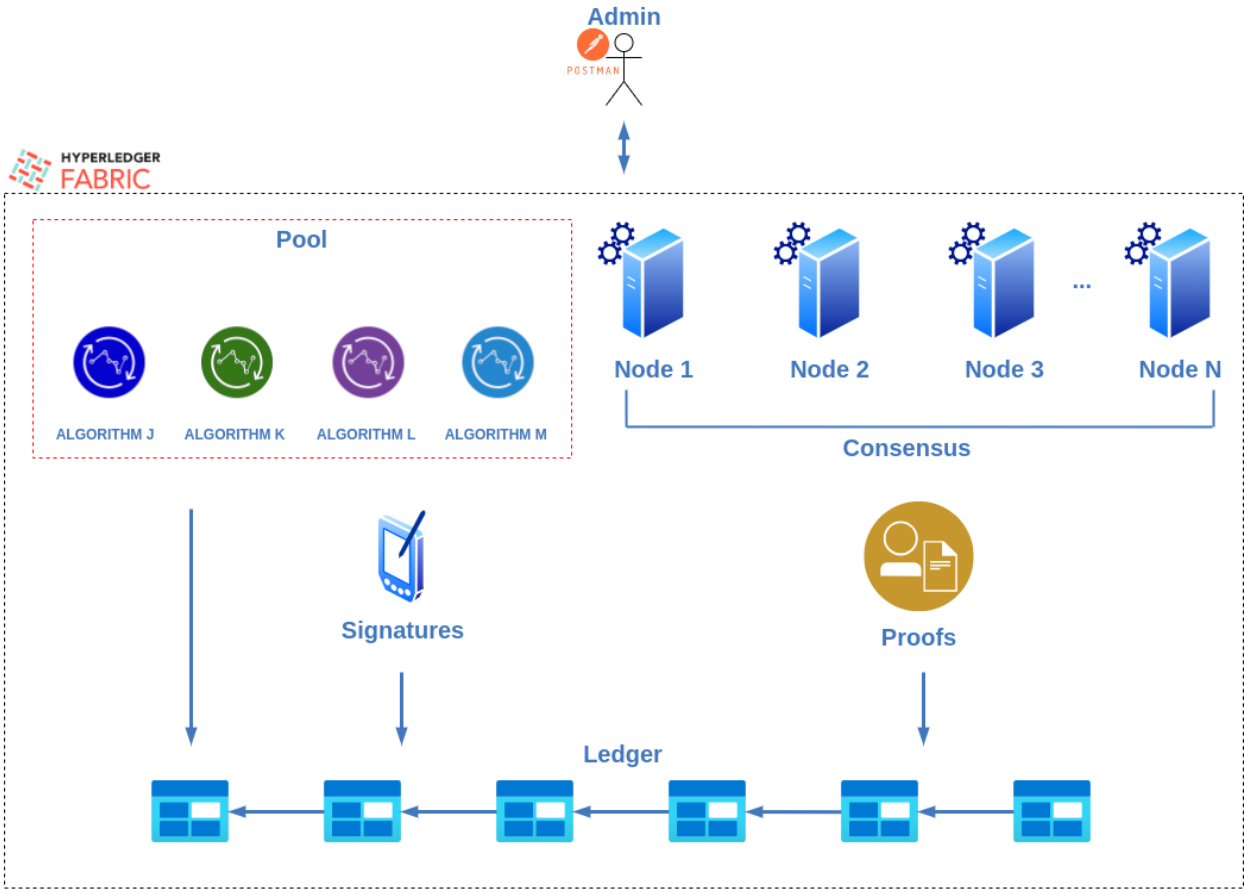
3.1 Εισαγωγή

Σκοπός της παρούσας διπλωματικής εργασίας είναι η αξιολόγηση και η εγγύηση της αξιοπιστίας των αλγορίθμων μηχανικής μάθησης μέσω επαλήθευσης αποδείξεων μηδενικής γνώσης. Στο παρακάτω σχήμα απεικονίζεται η απλουστευμένη και γενική μορφή της αρχιτεκτονικής του συστήματος που παρουσιάζουμε. Η αρχιτεκτονική του συστήματος, αφού δημιουργήσουμε το δίκτυο αλυσίδας-κορμού (blockchain), μπορεί να χωριστεί στις παρακάτω κατηγορίες:

1. Επιλογή και επικύρωση αλγορίθμου μηχανικής μάθησης με τη βοήθεια αλγορίθμου συναίνεσης (consensus algorithm).

2. Δημιουργία και υποβολή αποδείξεων στην αλυσίδα κορμού.

3. Επιβεβαίωση των αποδείξεων από το έξυπνο συμβόλαιο (smart contract).



Εικόνα 11 : Αρχιτεκτονική του συστήματος

3.2 Δημιουργία Αλυσίδας-Κορμού (Blockchain)

3.2.1 Επιλογή της αλυσίδας-κορμού

Για την υλοποίηση του δικτύου σε blockchain επιλέξαμε να κάνουμε χρήση του Hyperledger Fabric. Η επιλογή του Hyperledger Fabric δεν είναι τυχαία αφού αυτή η πλατφόρμα ανοικτού κώδικα προσφέρει μεγαλύτερη ασφάλεια, εμπιστευτικότητα, ευελιξία, ανθεκτικότητα και επεκτασιμότητα σε σχέση με άλλες. Επιπλέον, μόνο εξουσιοδοτημένα μέλη (permissioned blockchain) έχουν τη δυνατότητα να αλληλεπιδρούν με το δίκτυο αποκρύπτοντας έτσι πολύτιμες και εμπιστευτικές πληροφορίες από εξωτερικούς παράγοντες, εμποδίζοντας οποιαδήποτε

χειραγώγηση. Δε θα θέλαμε άλλωστε να επικυρώνουν και να αξιολογούν έναν αλγόριθμο χρήστες που είτε δεν έχουν τη γνωστική ικανότητα είτε έχουν κακόβουλους σκοπούς .

3.2.2 Έξυπνο Συμβόλαιο (Smart Contract)

Ένα από τα πιο σημαντικά εργαλεία που παρέχει η τεχνολογία του blockchain είναι το έξυπνο συμβόλαιο (smart contract). Το smart Contract ορίζει τη λογική της συναλλαγής που ελέγχει τον κύκλο ζωής ενός επιχειρηματικού αντικειμένου που περιέχεται στην παγκόσμια κατάσταση (global state). Το έξυπνο συμβόλαιο Hyperledger Fabric ονομάζεται επίσης chaincode και επιλέχθηκε η προγραμματιστική γλώσσα Go για την υλοποίηση του. Μέσω του έξυπνου συμβολαίου ορίζουμε τις δομές των δεδομένων, το πως μπορούν οι κόμβοι να αλληλεπιδρούν με τα δεδομένα, τις επιτρεπτές συναλλαγές αλλά και τις αρμοδιότητες και τα δικαιώματα κάθε κόμβου του δικτύου. Περισσότερα χαρακτηριστικά για το smart contract του δικτύου μας θα αναλυθούν και στις επόμενες ενότητες.

3.2.3 Προδιαγραφές Blockchain

3.2.3.1 Οργανισμοί - Κόμβοι

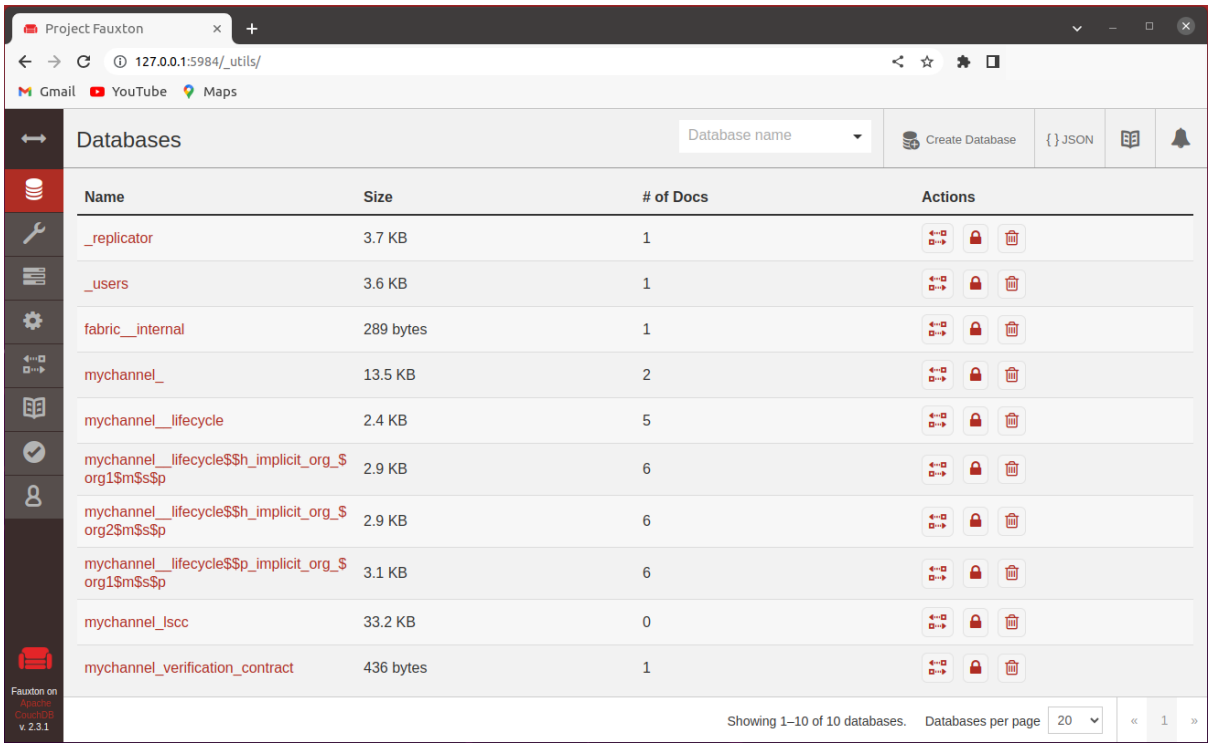
Για την υλοποίηση χρησιμοποιούμε δύο οργανισμούς Org1 και Org2 με δύο peer στον καθένα. Έτσι έχουμε peer0.org1 και peer1.org1 για τον πρώτο οργανισμό και peer0.org2 και peer0.org2 για τον δεύτερο.

Σύμφωνα με πρωτόκολλο διασποράς δεδομένων gossip κάθε οργανισμός πρέπει να έχει τουλάχιστον έναν anchor peer για κάθε κανάλι στο οποίο βρίσκεται. Σκοπός των anchor peers είναι να ανακαλύπτουν όλους τους peers που υπάρχουν στο δίκτυο και έτσι να βοηθούν στη σωστή επικοινωνία μεταξύ των οργανισμών .

Στο blockchain μας έχει επιλεγεί η χρήση της υπηρεσίας διάταξης συναλλαγών (Ordering Service) Raft. Η Raft είναι ανθεκτική σε αποτυχίες (crash fault tolerant – CFT) και βασίζεται σε leader nodes και orderer nodes. Ο ρόλος των orderer nodes είναι να θέτουν σε αυστηρή και παγιωμένη σειρά τις συναλλαγές που λαμβάνουν χώρα σε ένα κανάλι. Η χρήση των orderer

nodes έρχεται σε αντίθεση με τους πιθανοτικούς αλγορίθμους συναίνεσης που χρησιμοποιούνται για τον προσδιορισμό της σειράς των αλλαγών σε δημόσια δίκτυα, όπως το Ethereum και το Bitcoin, και εγγυάται ότι κάθε επικυρωμένο block συναλλαγών είναι τελικό και σωστό. Στο σύστημά μας χρησιμοποιούμε 3 orderer nodes οι οποίοι συντονίζονται και αποτελούν την υπηρεσία Διάταξης Συναλλαγών του δικτύου μας.
Όταν ένα μπλοκ πρέπει να σταλεί σε όλου τους κόμβους του δικτύου, προωθείται μόνο στον leader peer κάθε οργανισμού. Αυτός με τη σειρά του το διαμοιράζει στους υπόλοιπους κόμβους που υπάρχουν στον οργανισμό του. Η εκλογή των leader peer, γίνεται δυναμικά με ψηφοφορία.

Ως βάση δεδομένων για τη διατήρηση της παγκόσμιας κατάστασης του δικτύου χρησιμοποιείται η CouchDB. Η CouchDB είναι μια NoSQL βάση δεδομένων ανοιχτού κώδικα που αποθηκεύει δεδομένα σε μορφή JSON. Η CouchDB επιτρέπει την έκδοση ερωτημάτων (queries) μορφής JSON και τη χρήση δεικτών για την διάκριση των ερωτημάτων. Αυτό κάνει τα ερωτήματα πιο ευέλικτα και αποτελεσματικά και επιτρέπει την ανάληψη μεγάλων συνόλων δεδομένων από την βάση. Η σύγκριση αρχείων της μορφής JSON για την άντληση δεδομένων αντί για ένα απλό query αναζήτησης κλειδιών (key query) κάνει πιο εύκολη την ανάγνωση δεδομένων από το blockchain για τις εφαρμογές και το έξυπνο συμβόλαιο.



Εικόνα 12: CouchDB

3.2.3.2 Certificate Authority

Το Hyperledger fabric σαν permission blockchain απαιτεί όλες οι οντότητες, ανεξαρτήτως αν είναι μέρος του δικτύου ή απλά χρήστες που αλληλεπιδρούν με αυτό, να μπορούν να αναγνωριστούν. Η Δημιουργία των ψηφιακών πιστοποιητικών πραγματοποιείται με τη βοήθεια ενός Certificate Authority (CA) .

Για κάθε οργανισμό γίνονται με τη σειρά τα παρακάτω:

- Εγγραφή του CA admin
- Καταγραφή του peer0
- Καταγραφή του peer1
- Καταγραφή του χρήστη
- Καταγραφή του διαχειριστή του οργανισμού
- Παραγωγή του msp του peer0
- Παραγωγή του tls πιστοποιητικού του peer0
- Παραγωγή του msp του peer1
- Παραγωγή του tls πιστοποιητικού του peer1
- Παραγωγή του msp του user
- Παραγωγή του msp του διαχειριστή του οργανισμού

Παρόμοια διαδικασία ακολουθείται και για τους orderer.

3.3 Consensus Algorithm και Pool

Σε αυτήν την ενότητα θα μιλήσουμε για την διαδικασία με την οποία γίνεται ο έλεγχος και η επιλογή του αλγορίθμου μηχανικής μάθησης. Η προσέγγιση μας σε αυτό το θέμα είναι καθαρά θεωρητική γιατί οι διαδικασίες που χρειάζονται για τον έλεγχο και την επιλογή του αλγορίθμου ξεφεύγουν από τον σκοπό της παρούσας διπλωματικής εργασίας. Ωστόσο, πρέπει να γίνει

περιγραφή της διαδικασίας ώστε να είναι πιο κατανοητή η συνέχεια της αρχιτεκτονικής του συστήματός μας.

3.3.1 Επιλογή Consensus

Ο έλεγχος και επικύρωση του αλγορίθμου μηχανικής μάθησης γίνεται με τη βοήθεια των πρωτοκόλλων συναίνεσης. Η επικύρωση των αλγορίθμων μηχανικής μάθησης δεν μπορεί να γίνει από οποιονδήποτε χρήστη, αφού είναι απαραίτητο να υπάρχει το αντίστοιχο θεωρητικό υπόβαθρο, η προγραμματιστική ικανότητα, και η γνώση των σύγχρονων μεθόδων μηχανικής μάθησης. Χάριν περιγραφής, στη συγκεκριμένη περίπτωση, θεωρούμε ότι ο καταλληλότερος αλγόριθμος συναίνεσης είναι ο Proof of Authority (PoA). Όπως έχουμε ήδη αναφέρει στο Κεφάλαιο 2, ο αλγόριθμος αυτός δεν απαιτεί κανέναν δύσκολο υπολογισμό ή οικονομικό κίνητρο, αλλά βασίζεται στη ταυτότητα των κόμβων επικυρωτών θεωρώντας ότι αυτοί δε θα συμπεριφερθούν κακόβουλα, ώστε να προστατέψουν τη φήμη τους. Με αυτόν τον τρόπο, καταφέρνουμε να έχουμε σωστή αξιολόγηση της αξιοπιστίας των αλγορίθμων μηχανικής μάθησης. Η επιλογή του συγκεκριμένου αλγορίθμου συναίνεσης γίνεται διότι θυμίζει τη πραγματική διαδικασία επικύρωσης και ελέγχου ενός επιστημονικού άρθρου (paper). Κάθε paper περνάει από μία επιστημονική επιτροπή η οποία αποφασίζει για την εγκυρότητά του. Κάθε μέλος της επιτροπής ελέγχει σχολαστικά τα ευρήματα που παρουσιάζει το paper, και αν ομόφωνα αποφανθούν ότι είναι έγκυρα, γίνεται δημοσίευσή του. Τα μέλη της επιτροπής προέρχονται από διάφορα μέρη του κόσμου, είναι γνώστες του αντικειμένου, ελέγχουν επώνυμα τις δημοσιεύσεις και θεωρούνται αξιόπιστοι αξιολογητές.

3.3.2 Pool αλγορίθμων

Αφού έχει γίνει ο έλεγχος και η έγκριση των αλγορίθμων τοποθετούνται σε ένα pool, δηλαδή σε ένα δωμάτιο αναμονής. Στο pool βρίσκονται οι αλγόριθμοι για τους οποίους δεν έχουμε δημιουργήσει ακόμα αποδείξεις. Οι αλγόριθμοι περιμένουν να καλεστούν από κάποιον χρήστη ώστε να προχωρήσει η διαδικασία της δημιουργίας και της επαλήθευσης των αποδείξεων.

3.4 Hash function

Με τις συναρτήσεις κατακερματισμού (hash function) έχουμε τη ικανότητα να επεξεργαστούμε μια πληροφορία σε βαθμό που να μην μπορεί να αναπαραχθεί η αρχική της μορφή . Κάθε φορά που επεξεργαζόμαστε την ίδια πληροφορία με μία συνάρτηση hash προκύπτει το ίδιο αποτέλεσμα. Με αυτόν τον τρόπο μπορούμε να αποκρύψουμε σημαντικά στοιχεία επιτρέποντας όμως τον έλεγχο για την εγκυρότητά τους. Συναρτήσεις κατακερματισμού χρησιμοποιούμε σε τρία μέρη της αρχιτεκτονικής του συστήματός μας.

- Αρχικά, όταν γίνει ψηφοφορία μέσω του αλγορίθμου συναίνεσης και ελεγχθεί ότι ο αλγόριθμος της μηχανικής μάθησης είναι έγκυρος δημιουργείται μια πλειάδα (κλειδί, hash). Το hash έχει παραχθεί με τη βοήθεια του αλγορίθμου κατακερματισμού sha256.

- Έπειτα, έχουμε σαν είσοδο την πλειάδα (key, hash) στη συνάρτηση κατακερματισμού HMAC SHA256 και έτσι προκύπτει ένα νέο hash για κάθε αλγόριθμο μηχανικής μάθησης. Το hash αυτό έχει δεκαεξαδική μορφή σταθερού μήκους ανεξαρτήτως εισόδου.

- Αυτό το νέο hash θα είναι η είσοδός μας στις συναρτήσεις, Poseidon, MiMC5, MiMC7 και Pedersen που θα χρησιμοποιηθούν για την επιβεβαίωση μηδενικής γνώσης (ZKP) των αποδείξεων που θα αναλύσουμε στις επόμενες ενότητες.

Να σημειωθεί ότι όταν αναφέρουμε MiMC5 και MiMC7 εννοούμε τη συνάρτηση κατακερματισμού MiMC με δύναμη ύψωσης του x της συνάρτησης APN, 5 και 7 αντίστοιχα.

Η συνάρτηση κατακερματισμού HMAC SHA256 είναι γραμμένη σε javascript, ενώ οι Poseidon, MiMC5, MiMC7 και Pedersen είναι γραμμένες σε javascript και circom και είναι υλοποιημένες με τις βιβλιοθήκες circomlib και circomlibjs.
Για κάθε μία από τις Poseidon, MiMC5, MiMC7 και Pedersen πρώτα τρέχουμε τον κώδικα σε javascript για να πάρουμε το παραγόμενο hash, το οποίο είναι σε δεκαδική μορφή, και να το τοποθετήσουμε στο αντίστοιχο αρχείο input.json για τη συνάρτηση κατακερματισμού σε circom.

3.5 Circuits Compilation

Για να παράξουμε τις αποδείξεις πρέπει πρώτα να γράψουμε το κύκλωμά μας σε circom, και να το μεταγλωττίσουμε. Από τη μεταγλώττιση προκύπτουν τρία αρχεία με καταλήξεις .r1cs, .wasm και .sym.

- --r1cs: περιέχει το R1CS σύστημα περιορισμών του κυκλώματος σε δυαδική μορφή.
- --wasm: περιέχει τον κώδικα Wasm και άλλους φακέλους που χρησιμεύουν στη δημιουργία του witness.
- --sym: ένας φάκελος συμβόλων που απαιτείται για αποσφαλμάτωση ή για να εκτυπωθεί το σύστημα περιορισμών στην επιθυμητή μορφή.

Δημιουργούμε ένα έμπιστο περιβάλλον (trusted setup) για τη δημιουργία των απαραίτητων κλειδιών για το proof (proving key and verification key).

Ακολουθεί η δημιουργία του μυστικού (witness) που υπολογίζεται χρησιμοποιώντας και το αρχείο input.json το οποίο περιέχει το αποτέλεσμα της συνάρτησης κατακερματισμού HMAC SHA256 και το αποτέλεσμα το εκάστοτε αλγορίθμου κατακερματισμού που δημιουργήθηκε νωρίτερα μέσω της javascript (Poseidon ή MiMC5 ή MiMC7 ή Pedersen). Εσωτερικά η δημιουργία των παραπάνω γίνεται με τη χρήση του npm πακέτου “snarkjs”, το οποίο αποτελεί ένα εργαλείο υλοποίησης zk-SNARK σε javascript.
Τέλος, γίνεται η μετατροπή των proving key και witness σε δυαδικό αρχείο και δυαδικό αρχείο go αντίστοιχα για να χρησιμοποιηθούν έπειτα στη δημιουργία αποδείξεων.

3.6 Δημιουργία Αποδείξεων

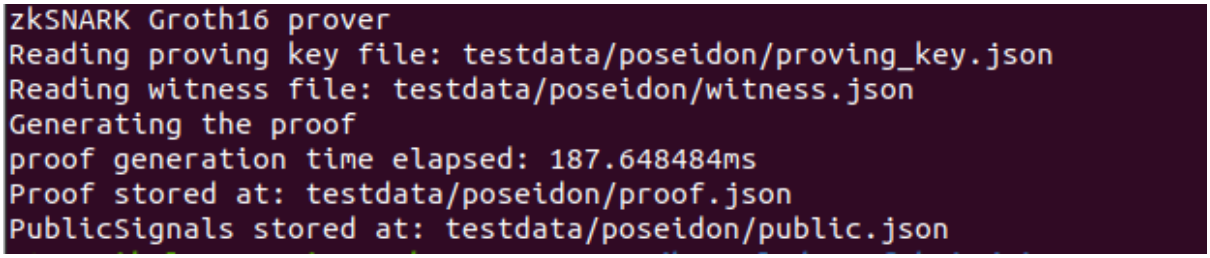
Παράγουμε την απόδειξη έξω από το δίκτυο αλυσίδας-κορμού για να αυξήσουμε την αποδοτικότητα του δικτύου και να μπορεί αυτό να κλιμακωθεί.

Το σενάριο δημιουργίας απόδειξης είναι γραμμένο σε go [30] και λαμβάνει ως είσοδο τις παραμέτρους proving key και witness που παρήχθησαν στο προηγούμενο βήμα και δίνει ως έξοδο ένα αρχείο όπου είναι αποθηκευμένη η απόδειξη (proof.json) αλλά και ένα αρχείο με τα

δημόσια σήματα (public signals), δηλαδή τις εισόδους και εξόδους των αριθμητικών κυκλωμάτων που χρησιμοποιούνται και οι οποίες δεν θεωρούνται μυστικές.

Σημειώνεται ότι το σύστημα απόδειξης που χρησιμοποιείται είναι το Groth16. Το Groth16 προτάθηκε από τον ερευνητή Jens Groth και παρουσιάζει έναν αλγόριθμο SNARK που βασίζεται σε γινόμενα ταιριάσματος (pairing based) και ικανοποιεί αριθμητικά κυκλώματα. Για περισσότερες λεπτομέρειες παραπέμπουμε στην αντίστοιχη δημοσίευση [31] .

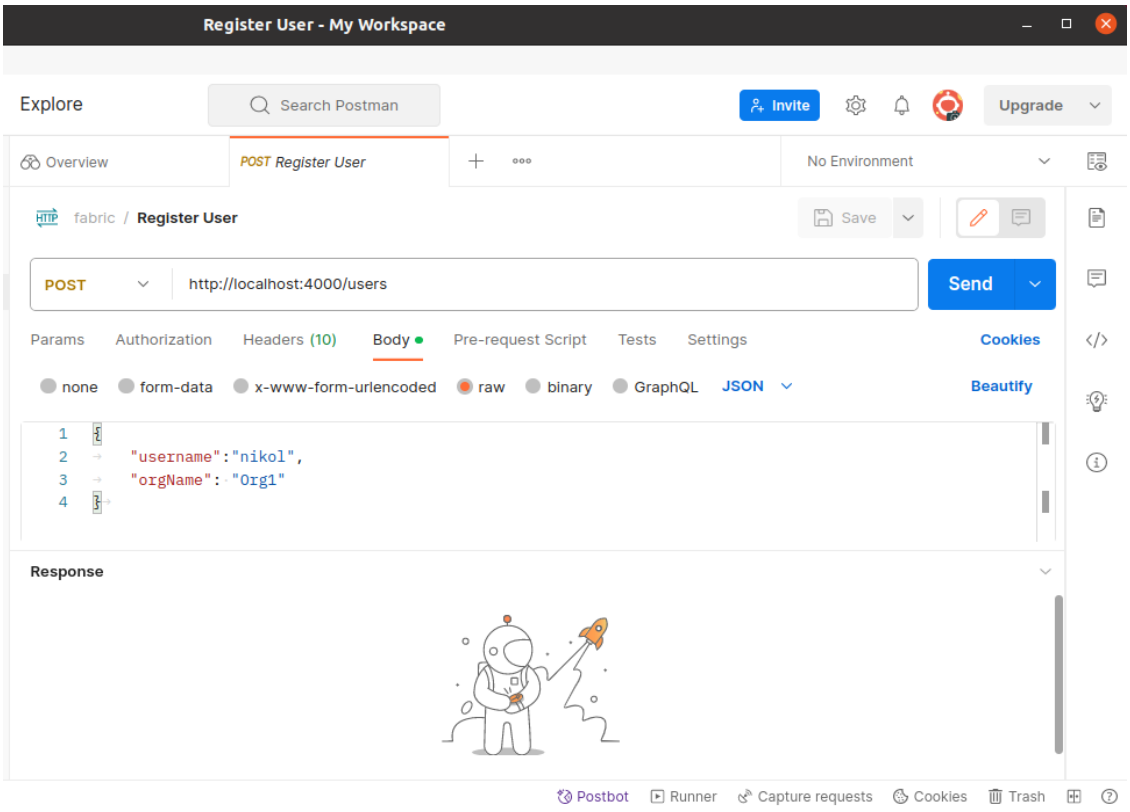
Το πρωτόκολλο βασίζεται σε κρυπτογραφία ταιριάσματος καμπύλων (Pairing Based Cryptography) η οποία υλοποιείται μέσω της βιβλιοθήκης bn256.



Εικόνα 13: Δημιουργία απόδειξης

3.7 API

Για να υλοποιηθεί η διεπαφή με το δίκτυο Blockchain, δημιουργήθηκε ένα API με χρήση της πλατφόρμας Nodejs. Συγκεκριμένα, οι λειτουργίες του API που έχουν υλοποιηθεί μέχρι στιγμής είναι οι εξής:



Εικόνα 14: Εγγραφή χρήστη

- Register User: Με αποστολή μιας πρότυπης μεθόδου POST στο κατάλληλο URI, γίνεται εγγραφή χρήστη στο δίκτυο. Ο χρήστης λαμβάνει το επιθυμητό όνομα και καταχωρείται στον επιθυμητό οργανισμό. Δημιουργούνται για αυτόν ένα private, ένα public key και ένα token που θα χρησιμεύσει στην εξουσιοδότηση του χρήστη για συναλλαγές που θα καταχωρηθούν στο Blockchain. Το token αυτό ακολουθεί το πρότυπο jwt (jason web token).


```
{
    "success": true,
    "message": "nikol enrolled Successfully",
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
        eyJleHAiOjE2OTg3ODEwNzIsInVzZXJuYW1lIjoibmlrb2wiLCJvcmdOYW1lIjoiT3JnMSIsImlhdCI6MTY5ODc0NTA3M
        n0.1-gyKdgrHnFuTgqP0SHqfbbXdZvhSNgznekm7yH_Ep8"
}
```


Εικόνα 15: Επιτυχής Εγγραφή Χρήστη

Στην περίπτωση που λαμβάνει χώρα η εγγραφή ενός χρήστη αλλά δεν υπάρχει ήδη admin στον οργανισμό όπου γίνεται η εγγραφή, δημιουργείται ένας admin και ακολουθεί η εγγραφή του χρήστη.

```
Wallet path: /home/ntua/hyperledger_fabric/with_ca/api-2.0/org1-wallet
An identity for the admin user "admin" does not exist in the wallet
calling enroll Admin method
Wallet path: /home/ntua/hyperledger_fabric/with_ca/api-2.0/org1-wallet
Successfully enrolled admin user "admin" and imported it into the wallet
Admin Enrolled Successfully
```

Εικόνα 16: Αν δεν υπάρχει admin γίνεται η εγγραφή του

- Submit Proof: Κάποιος εξουσιοδοτημένος χρήστης υποβάλλει (στέλνοντας και πάλι μέθοδο POST στο κατάλληλο URI) στο Blockchain τα αρχεία που έχουν παραχθεί κατά τη δημιουργία της απόδειξης (proof) και θα είναι χρήσιμα για την επιβεβαίωσή της. Μετά την υποβολή η απόδειξη αποθηκεύεται στη βάση δεδομένων CouchDB.

- Verification: Κάποιος εξουσιοδοτημένος χρήστης ζητάει την επιβεβαίωση μιας απόδειξης που έχει υποβληθεί στο blockchain.

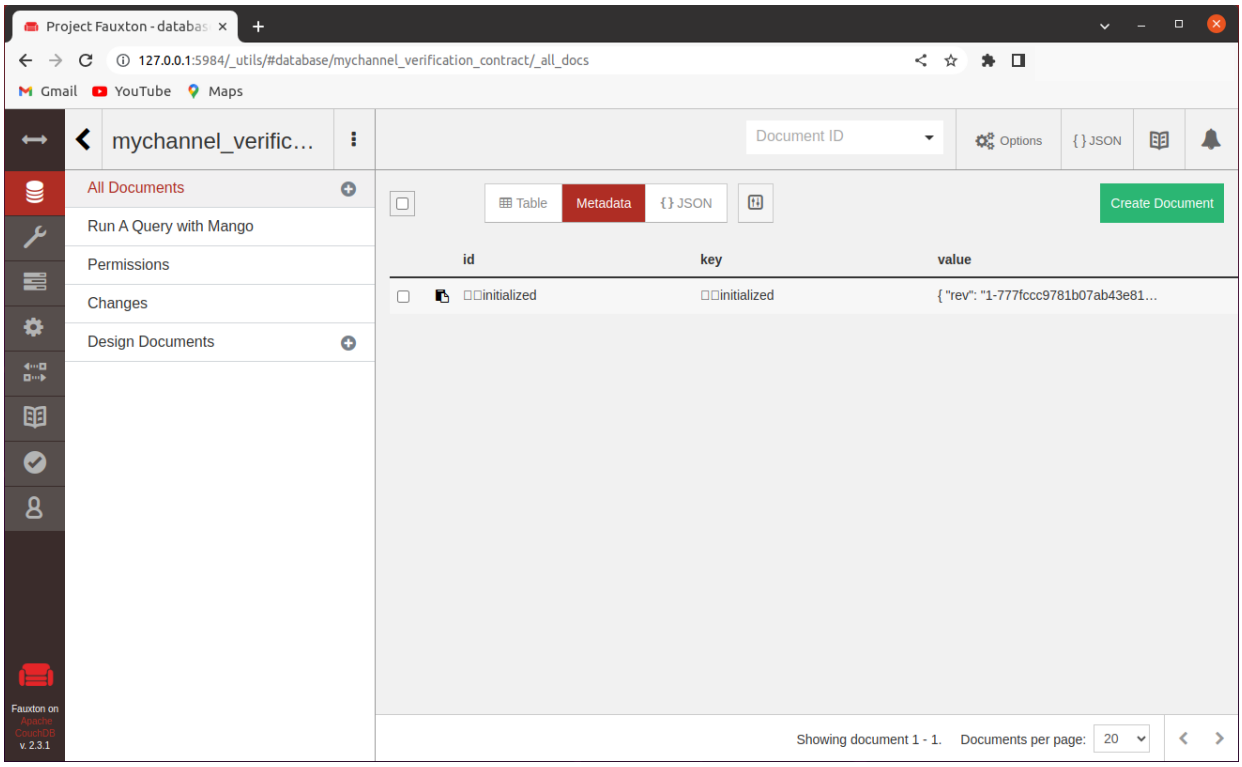
Κάθε φορά που αποστέλλεται μια πρότυπη μέθοδος μέσω του API, ενεργοποιείται το δίκτυο blockchain, αφού καλείται αντίστοιχη συνάρτηση από το smart contract.

Στην περίπτωση που αποσταλεί μια πρότυπη μέθοδος που δεν είναι σωστά ορισμένη (το πεδίο της συνάρτησης στο σώμα της μεθόδου είναι λανθασμένο), επιστρέφεται το κατάλληλο μήνυμα σφάλματος, όπως φαίνεται στην παρακάτω εικόνα:

Εικόνα 17: Η έξοδος του API για λανθασμένο όνομα συνάρτησης όπως φαίνεται στο Postman

3.8 Submit on chain

Πριν υποβάλλουμε κάποια απόδειξη στο δίκτυο blockchain η βάση δεδομένων μας είναι άδεια και βρίσκεται στην αρχική της κατάσταση.

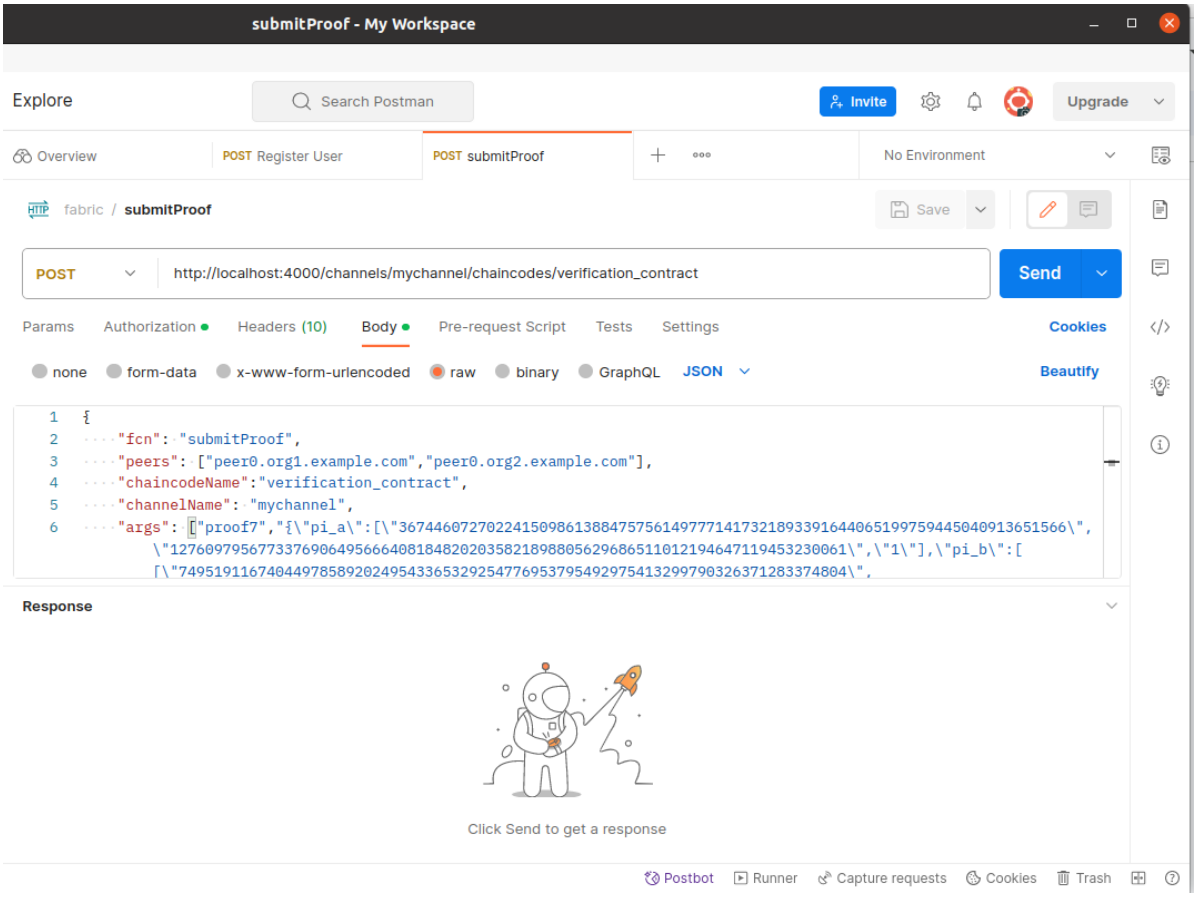


Εικόνα 18: Η αρχική κατάσταση του CouchDB Database

Στα πλαίσια του API έχουμε δημιουργήσει μια μέθοδο submitProof η οποία όταν αποστέλλεται με χρήση πρότυπης μεθόδου POST στο κατάλληλο URI, ενεργοποιεί τη συνάρτηση submitProof του έξυπνου συμβολαίου.

Τα ορίσματα της συνάρτησης αυτής είναι ένα αναγνωριστικό της απόδειξης που υποβάλλεται (π.χ. “proof1”), τα αρχεία proof.json και public.json που παρήχθησαν κατά τη δημιουργία της απόδειξης και το verification_key.json που δημιουργήθηκε κατά τη διάρκεια της εδραίωσης ασφαλούς περιβάλλοντος (trusted setup).

Τα αρχεία proof.json, public.json και verification_key.json, πριν τεθούν ως ορίσματα της συνάρτησης έχουν μετατραπεί σε συμβολοσειρές μέσω της συνάρτησης stringify.

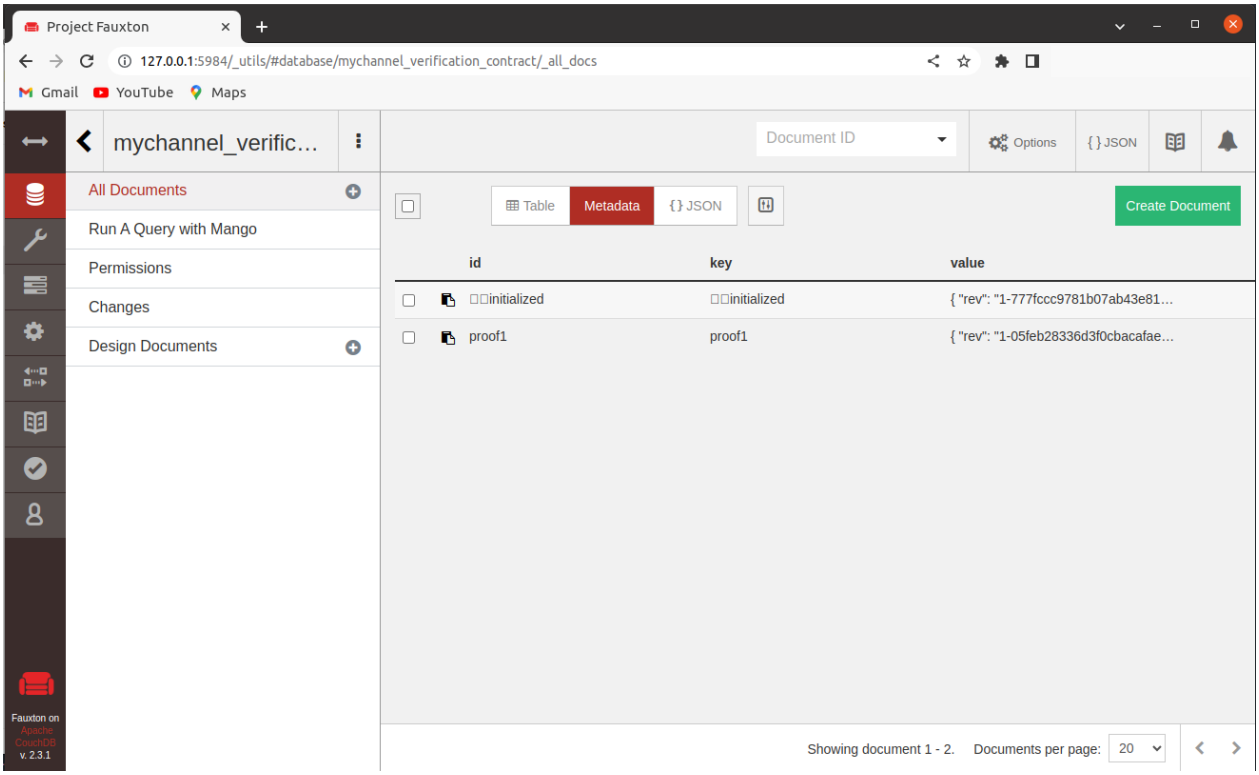


Εικόνα 19: Μέθοδος για την υποβολή της απόδειξης στο Blockchain

Εάν η υποβολή απόδειξης εκτελεστεί σωστά, λαμβάνουμε το αντίστοιχο μήνυμα μέσω του Postman.

Εικόνα 20: Επιτυχής Υποβολή της απόδειξης στο Blockchain

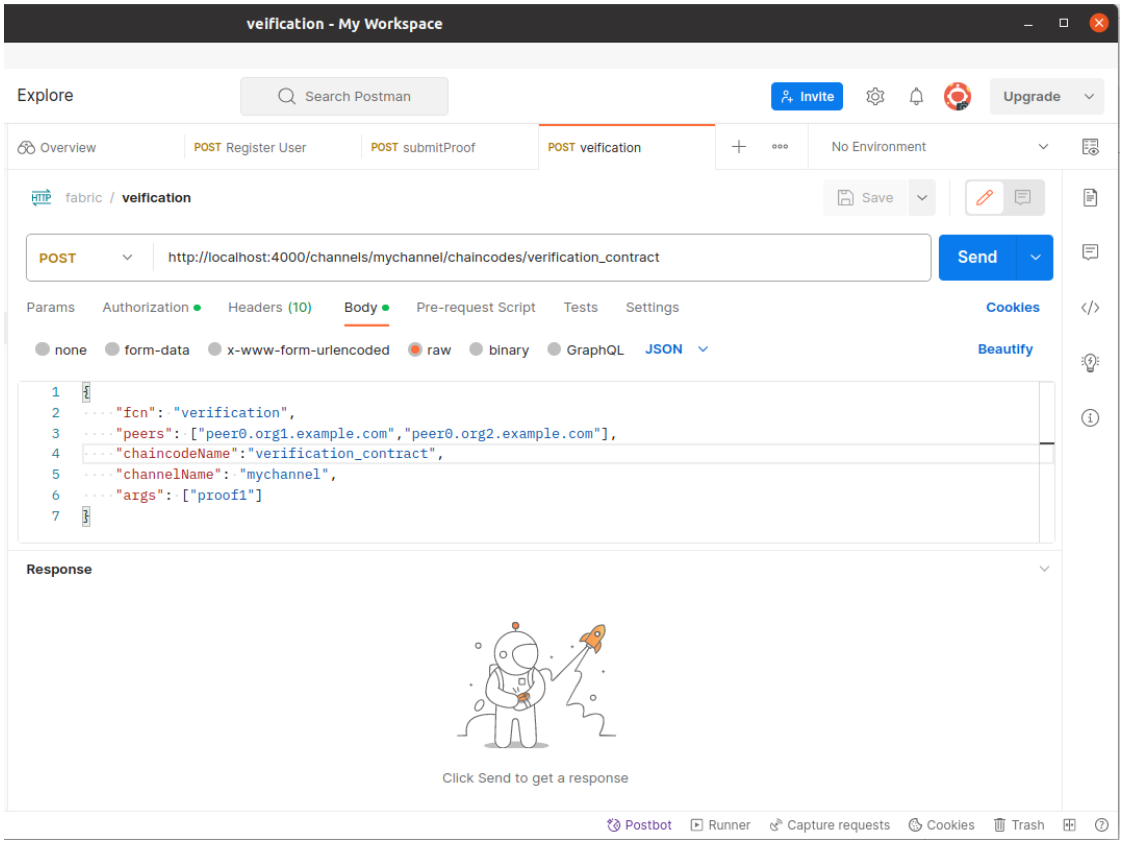
Παρατηρούμε επιπλέον ότι η βάση δεδομένων μας έχει ανανεωθεί και εμφανίζεται η απόδειξη που υποβλήθηκε με το αντίστοιχο αναγνωριστικό.



Εικόνα 21: Η βάση Δεδομένων μετά την υποβολή της πρώτης απόδειξης

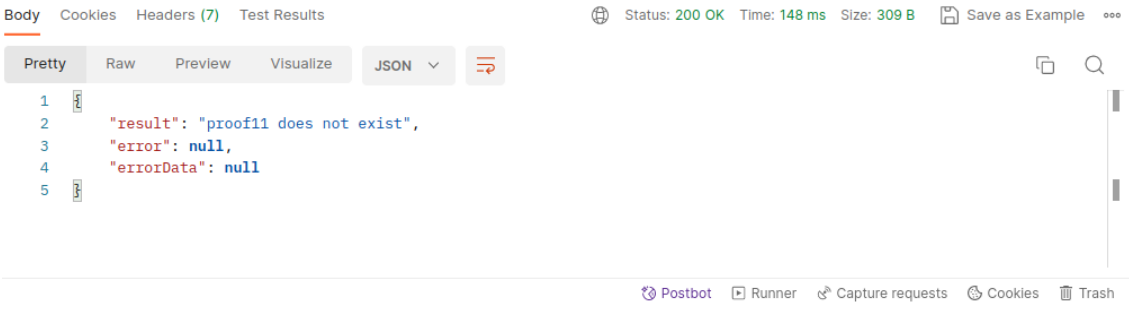
3.9 Verification from smart contract

Για την επαλήθευση της απόδειξης από το έξυπνο συμβόλαιο έχουμε υλοποιήσει μια νέα συνάρτηση εντός του API, τη verification. Όταν αυτή αποσταλεί με πρότυπη μέθοδο POST στο κατάλληλο URI, ενεργοποιεί τη συνάρτηση verification του έξυπνου συμβολαίου που είναι γραμμένη σε go [30]. Η συνάρτηση αυτή λαμβάνει ως είσοδο το αναγνωριστικό της προς-επαλήθευση-απόδειξης και διαβάζει από την βάση δεδομένων CouchDB τα στοιχεία της. Τα στοιχεία αυτά μορφοποιούνται κατάλληλα ώστε να εισέλθουν ως ορίσματα στη συνάρτηση verify του έξυπνου συμβολαίου, η οποία αποφαίνεται για το αν η απόδειξη είναι έγκυρη ή όχι.



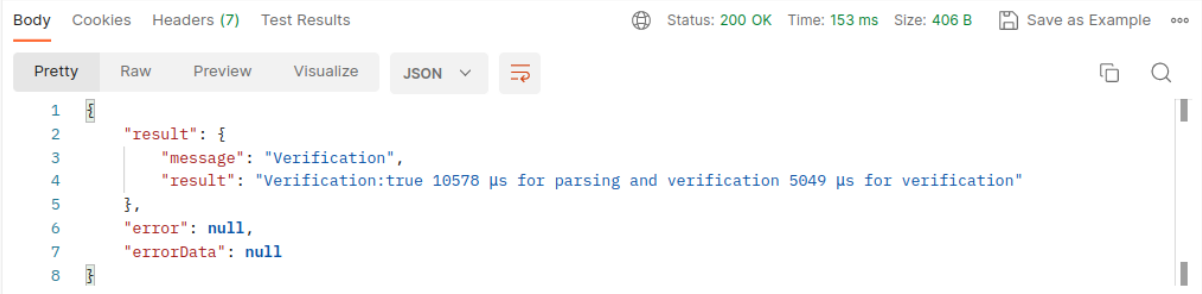
Εικόνα 22: Μέθοδος για την επαλήθευση της απόδειξης

Εάν το αναγνωριστικό που δώσαμε ως είσοδο δεν αντιστοιχεί σε κάποια απόδειξη που έχει υποβληθεί στη βάση δεδομένων μας, τότε λαμβάνουμε το κατάλληλο μήνυμα σφάλματος από το Postman.



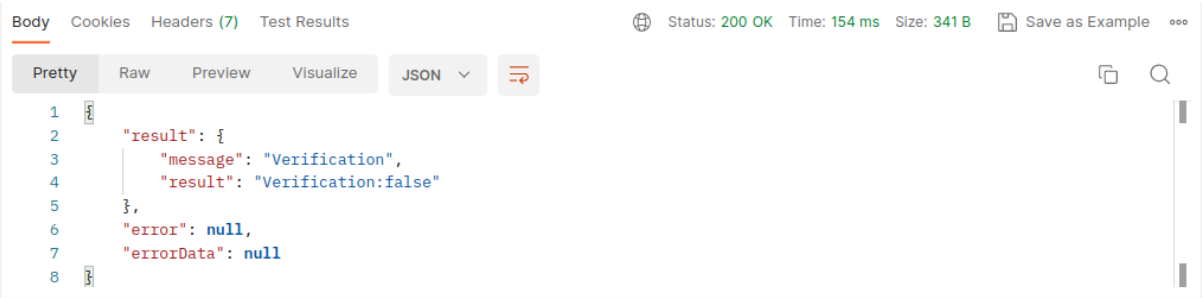
Εικόνα 23: Η απόδειξη που ζητήθηκε να επαληθευτεί δεν υπάρχει

Αντίθετα, αν έγινε σωστή υποβολή του αναγνωριστικού και η διαδικασία της επαλήθευσης είναι επιτυχής, τότε λαμβάνουμε το ανάλογο μήνυμα μέσω του Postman, ενώ παρατίθενται σε αυτό και οι χρόνοι που απαιτήθηκαν για την επαλήθευση.



Εικόνα 24: Επιτυχής επαλήθευση της απόδειξης και χρόνοι επαλήθευσης

Τέλος, εάν το αναγνωριστικό αντιστοιχεί σε κάποια απόδειξη στη βάση, αλλά η απόδειξη αυτή δεν εγκρίνεται από τον verifier, λαμβάνουμε μήνυμα αποτυχίας επαλήθευσης από το Postman.



Εικόνα 25: Αποτυχία Επαλήθευσης της απόδειξης

Συνεπώς, με την αρχιτεκτονική του πληροφοριακού συστήματος που σχεδιάσαμε, διασφαλίζεται η αξιοπιστία των αλγορίθμων μηχανικής μάθησης, χωρίς να προδίδεται η ταυτότητά τους λόγω της χρήσης της τεχνολογίας αποδείξεων μηχανικής γνώσης (ZKP).

Κεφάλαιο 4

Πειραματικές Μετρήσεις και Αποτελέσματα

Παρακάτω παρατίθενται τα αποτελέσματα και οι μετρήσεις των χρονικών αποδόσεων που εμφάνισαν οι διαφορετικές συναρτήσεις κατακερματισμού. Μετράμε συγκεκριμένα τον χρόνο που απαιτείται για την εκτέλεση τριών εργασιών:

- Χρόνος δημιουργίας ασφαλούς περιβάλλοντος (trusted setup)
- Χρόνος παραγωγής απόδειξης (proof generation)
- Χρόνος επιβεβαίωσης απόδειξης (proof validation)

Για κάθε μια από τις συναρτήσεις κατακερματισμού (Poseidon, MiMC5, MiMC7) έχουμε σαν είσοδο εντός του input.json:

- Τον δεκαεξαδικό αριθμό που προέκυψε από την HMAC SHA256
- Τον δεκαδικό αριθμό που προέκυψε από την αντίστοιχη συνάρτηση κατακερματισμού σε javascript

Για τις συναρτήσεις MiMC5 και MiMC7 στο αρχείο input.json βάζουμε εκτός των παραπάνω και την τιμή της παραμέτρου k. Επιλέγουμε να παρουσιάσουμε αποτελέσματα για k=2 και για k=α

με α = 4979690610316221117590372895361002649292617878981 7

που προέκυψε από τυχαία παραγωγή αριθμού, ο οποίος είναι σύμφωνος με τους περιορισμούς της συνάρτησης MiMC. Ο αριθμός των γύρων (rounds) για την συνάρτηση MiMC είναι 91.

Για τη συνάρτηση Pedersen έχουμε σαν είσοδο το Little Endian του δεκαεξαδικού αριθμού που προέκυψε από την HMAC SHA256 και τον δεκαδικό αριθμό από την αντίστοιχη συνάρτηση κατακερματισμού σε javascript.

Διαπιστώθηκε ότι η συνάρτηση Pedersen σε javascript δεν παράγει πάντα το ίδιο αποτέλεσμα με τη συνάρτηση Pedersen σε circom για την ίδια είσοδο μεγέθους 256bit. Ωστόσο, για είσοδο μεγέθους 248bit παράγουν πάντα το ίδιο αποτέλεσμα. Αυτό υποδεικνύει ότι υπάρχει θέμα στον τρόπο με τον οποίο γίνεται η κρυπτογράφηση της εισόδου πριν αυτή εισέλθει στη συνάρτηση. Η λύση όμως του παραπάνω θέματος ξεφεύγει από τον ερευνητικό σκοπό της παρούσας διπλωματικής εργασίας. Εντούτοις, μέσω διαδικασίας δοκιμής και σφάλματος (trial and error) βρέθηκε είσοδος μεγέθους 256bit για την οποία πράγματι και οι δύο υλοποιήσεις της Pedersen δίνουν το ίδιο αποτέλεσμα.

Θεωρώντας ότι οι χρόνοι δεν διαφέρουν για εισόδους ίδιου μεγέθους, παρουσιάζουμε τα αποτελέσματα της Pedersen χρησιμοποιώντας την είσοδο που προαναφέρθηκε.

4.1 Circuits

Μετά την μεταγλώττιση των κυκλωμάτων σε circom, μπορούμε να δούμε τα καλώδια από τα οποία αυτά αποτελούνται αλλά και τον αριθμό των περιορισμών που θέτουν. Τα παραθέτουμε στον πίνακα 1. Ο αριθμός των wires σχετίζεται με τη μεταφορά δεδομένων εντός του κυκλώματος, ενώ ο αριθμός των περιορισμών σχετίζεται με τις αριθμητικές πράξεις (operations) που εκτελούνται στα πλαίσια του κάθε κυκλώματος. Το πιο περίπλοκο κύκλωμα είναι αυτό που δημιουργήθηκε για τη συνάρτηση Pedersen, ενώ αντίθετα το απλούστερο είναι αυτό που αντιστοιχεί στη συνάρτηση Poseidon.

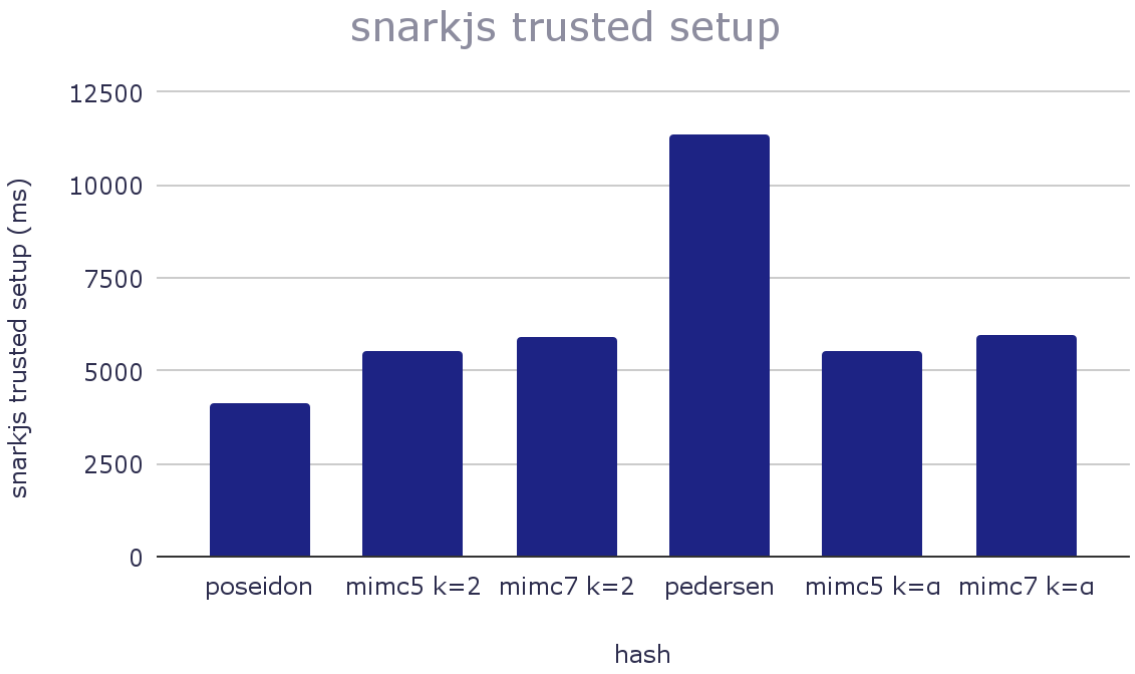
hash	Wires	Constraints
poseidon	215	213
mimc5 k = 2	276	273
mimc7 k = 2	367	364
pedersen	709	708
mimc5 k = a	276	273
mimc7 k = a	367	364

Πίνακας 1 : Wires και Constraints για κάθε hash

4.2 Trusted Setup

Στο Διάγραμμα 1 παραθέτουμε τον χρόνο που απαιτήθηκε για την εδραίωση περιβάλλοντος εμπιστοσύνης (trusted setup) για τα διαφορετικά hashes. Παρατηρούμε ότι οι χρόνοι ακολουθούν την πολυπλοκότητα των κυκλωμάτων της κάθε συνάρτησης.

Εύκολα διακρίνεται η Pedersen χρειάζεται σχεδόν τον διπλάσιο χρόνο για τη δημιουργία του trusted setup. Για τα υπόλοιπα hashes ο χρόνος που απαιτείται για τη δημιουργία του περιβάλλοντος κυμαίνεται σε παρόμοια επίπεδα. Ο μικρότερος χρόνος εμφανίζεται στην περίπτωση της Poseidon Hash πράγμα που συμπίπτει με την απλότητα του αντίστοιχου κυκλώματος.

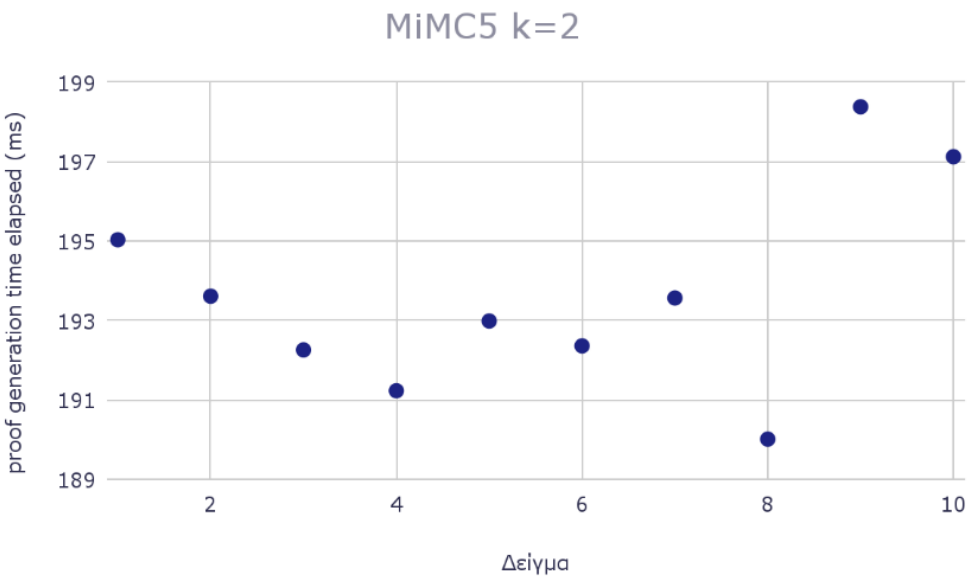


Διάγραμμα 1: Απαιτούμενος χρόνος για trusted setup για κάθε hash

4.3 Proof Generation

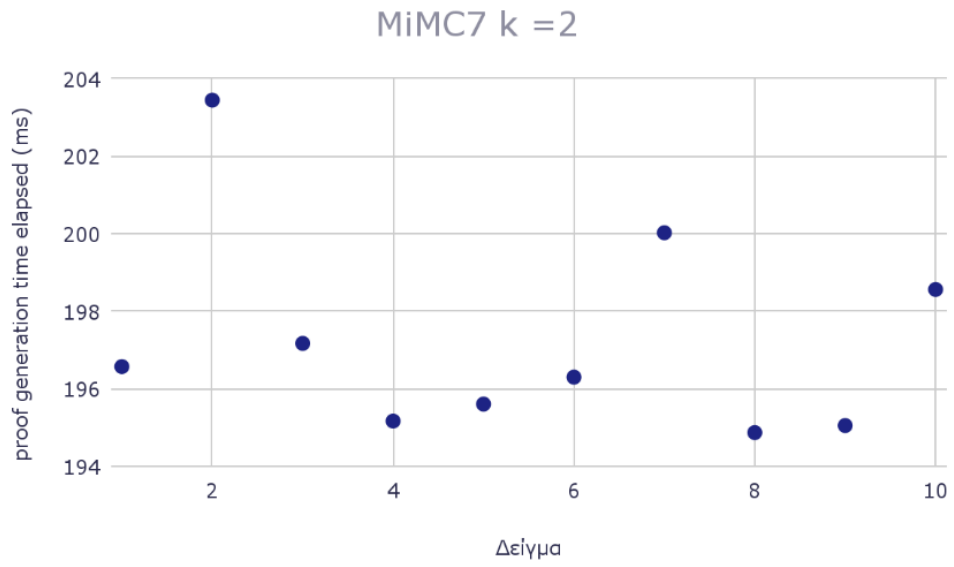
Στα ακόλουθα διαγράμματα παρουσιάζονται οι χρόνοι που απαιτούνται για την δημιουργία αποδείξεων για τις διαφορετικές συναρτήσεις κατακερματισμού. Για κάθε συνάρτηση επαναλαμβάνουμε το πείραμα παραγωγής απόδειξης δέκα φορές χρησιμοποιώντας την ίδια είσοδο input.json. Λόγω της τυχαιότητας που εμφανίζεται στη διαδικασία, αλλά και της κατάστασης του μηχανήματος στο οποίο εκτελείται η διαδικασία παρατηρούμε ότι δεν έχουμε σταθερή τιμή χρόνου για όλα τα πειράματα, αλλά υπάρχουν διακυμάνσεις. Στον πίνακα 2 παρουσιάζεται ο μέσος όρος των αποτελεσμάτων για κάθε μια από τις συναρτήσεις.

4.3.1 MiMC



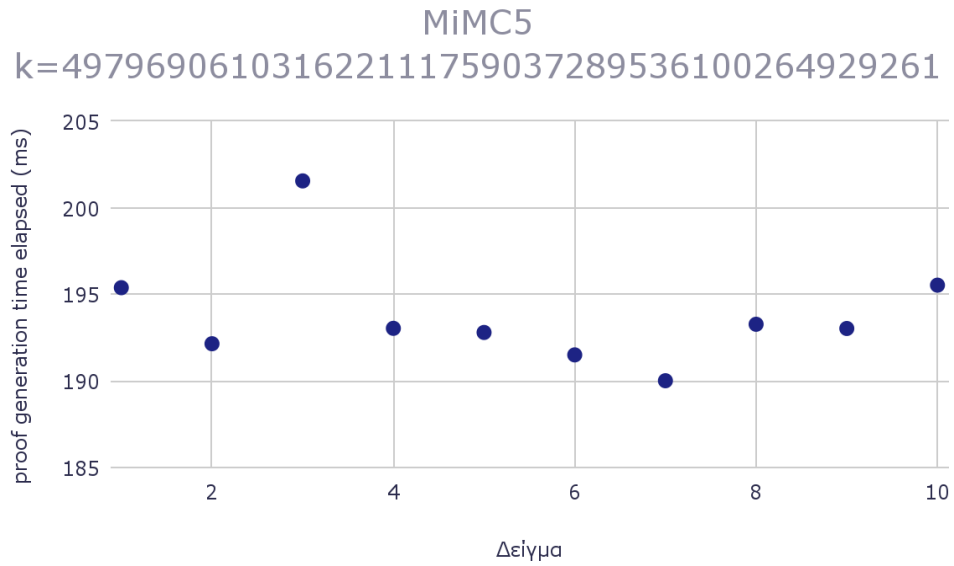
Διάγραμμα 2: *Μετρήσεις χρόνων παραγωγής αποδείξεων με MiMC5, k = 2*

Στο διάγραμμα 2 παρουσιάζονται τα 10 διαφορετικά δείγματα χρόνου για την παραγωγή αποδείξεων με τη συνάρτηση MiMC5 για k=2. Οι περισσότερες τιμές κυμαίνονται ανάμεσα στα 191ms και 195ms. Παρατηρούμε ότι υπάρχουν τιμές που αποκλίνουν αρκετά από τον μέσο όρο όπως τα δείγματα 8,9 και 10.



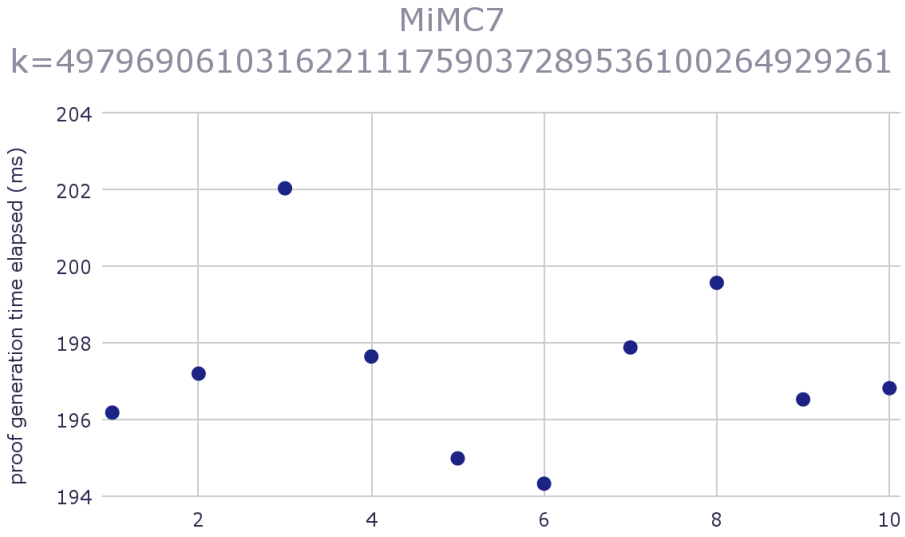
Διάγραμμα 3 : *Μετρήσεις χρόνων παραγωγής αποδείξεων με MiMC7, k = 2*

Στο διάγραμμα 3 παρουσιάζονται τα 10 διαφορετικά δείγματα χρόνου για την παραγωγή αποδείξεων με τη συνάρτηση MiMC7 για k=2. Οι περισσότερες τιμές κυμαίνονται ανάμεσα στα 195ms και 197ms. Παρατηρούμε ότι υπάρχουν τιμές που αποκλίνουν αρκετά από τον μέσο όρο όπως τα δείγματα 2,7 και 10.



Διάγραμμα 4: Μετρήσεις χρόνων παραγωγής αποδείξεων με MiMC5, k = a

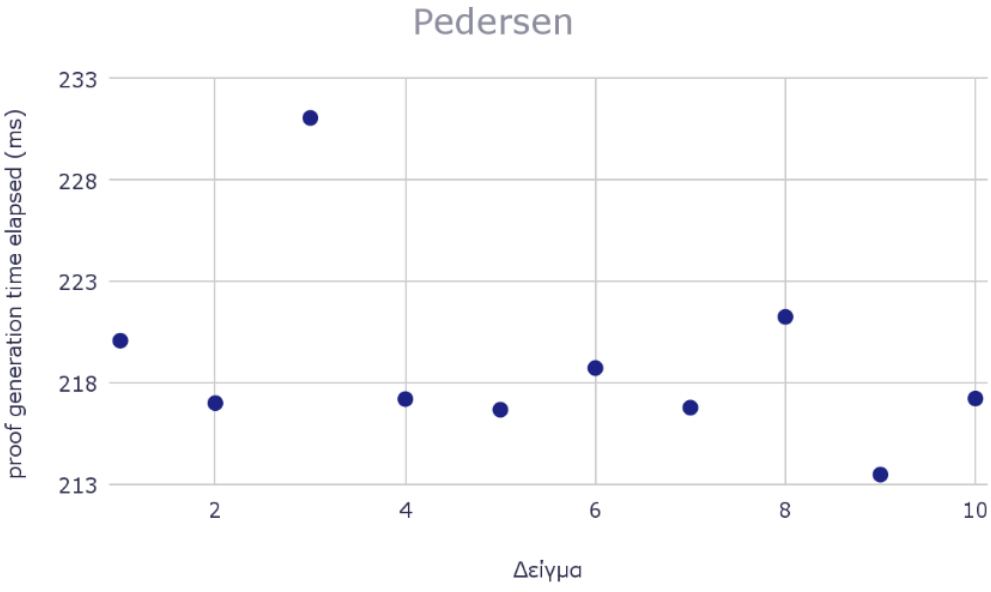
Στο διάγραμμα 4 παρουσιάζονται τα 10 διαφορετικά δείγματα χρόνου για την παραγωγή αποδείξεων με τη συνάρτηση MiMC5 για k=α. Οι περισσότερες τιμές κυμαίνονται ανάμεσα στα 190ms και 196ms. Παρατηρούμε ότι η τιμή του δείγματος 3 αποκλίνει αρκετά από τον μέσο όρο.



Διάγραμμα 5: Μετρήσεις χρόνων παραγωγής αποδείξεων με MiMC7, k = a

Στο διάγραμμα 5 παρουσιάζονται τα 10 διαφορετικά δείγματα χρόνου για την παραγωγή αποδείξεων με τη συνάρτηση MiMC7 για k=α. Οι περισσότερες τιμές κυμαίνονται ανάμεσα στα 194ms και 198ms. Παρατηρούμε ότι οι τιμές των δειγμάτων 3 και 8 αποκλίνουν αρκετά από τον μέσο όρο.

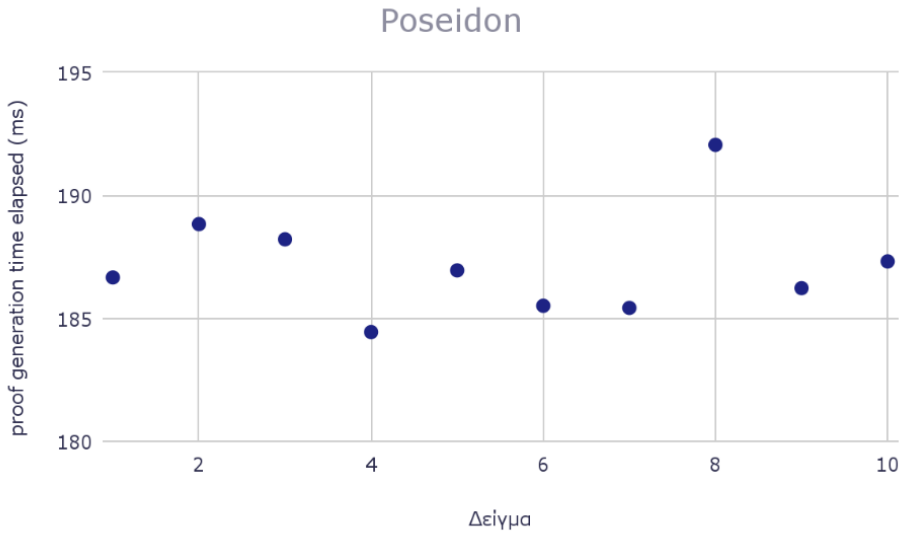
4.3.2 Pedersen



Διάγραμμα 6: Μετρήσεις χρόνων παραγωγής αποδείξεων με Pedersen

Στο διάγραμμα 6 παρουσιάζονται τα 10 διαφορετικά δείγματα χρόνου για την παραγωγή αποδείξεων με τη συνάρτηση Pedersen. Οι περισσότερες τιμές κυμαίνονται ανάμεσα στα 213ms και 220ms. Παρατηρούμε ότι η τιμή του δείγματος 3 αποκλίνει αρκετά από τον μέσο όρο.

4.3.3 Poseidon



Διάγραμμα 7: Μετρήσεις χρόνων παραγωγής αποδείξεων με Poseidon

Στο διάγραμμα 7 παρουσιάζονται τα 10 διαφορετικά δείγματα χρόνου για την παραγωγή αποδείξεων με τη συνάρτηση Poseidon. Οι περισσότερες τιμές κυμαίνονται ανάμεσα στα 185ms και 190ms. Παρατηρούμε ότι η τιμές είναι σχετικά συγκεντρωμένες γύρω από τον μέσο όρο και δεν παρουσιάζουν μεγάλη διακύμανση.

hash	proof generation time elapsed (ms)
poseidon	187,1615654
mimc5 k = 2	193,6555334
mimc7 k = 2	197,2750942
pedersen	218,9537774
mimc5 k = a	193,8229378
mimc7 k = a	197,3190736

Πίνακας 2 : Μέσος όρος χρόνου δημιουργίας αποδείξεων για κάθε hash

Παρατηρούμε ότι η πιο αργή παραγωγή αποδείξεων είναι αυτή για την οποία χρησιμοποιείται η συνάρτηση κατακερματισμού Pedersen. Οι υπόλοιπες συναρτήσεις δίνουν παρόμοιο χρόνο παραγωγής αποδείξεων με την πιο γρήγορη να είναι η Poseidon.

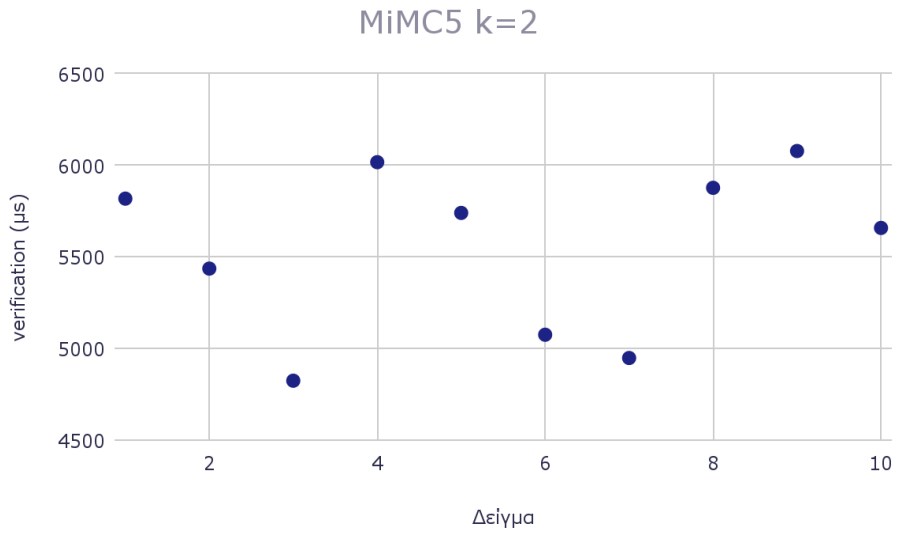
Αξίζει να σημειωθεί ότι για τις συναρτήσεις κατακερματισμού MiMC η τιμή k δεν επηρεάζει σημαντικά τον χρόνο παραγωγής απόδειξης, αφού οι μέσοι χρόνοι παραγωγής απόδειξης με MiMC5 με k = 2, MiMC5 με κ=α και MiMC7 με k = 2, MiMC7 με κ=α διαφέρουν για λιγότερο από 1ms.

Φαίνεται ότι οι χρόνοι παραγωγής αποδείξεων ακολουθούν την περιπλοκότητα των κυκλωμάτων που αντιστοιχούν στις συναρτήσεις κατακερματισμού.

4.4 Verification

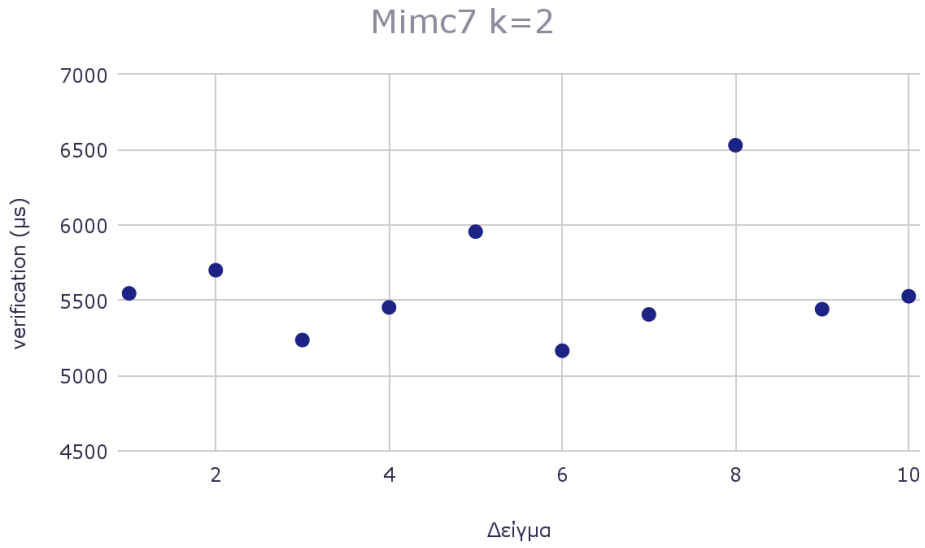
Στα ακόλουθα διαγράμματα παρουσιάζονται οι χρόνοι που απαιτούνται για την επικύρωση αποδείξεων για τις διαφορετικές συναρτήσεις κατακερματισμού. Για κάθε συνάρτηση επαναλαμβάνουμε το πείραμα επικύρωσης απόδειξης δέκα φορές χρησιμοποιώντας την ίδια απόδειξη. Λόγω της τυχαιότητας που εμφανίζεται στη διαδικασία, αλλά και της κατάστασης του μηχανήματος στο οποίο εκτελείται η διαδικασία παρατηρούμε ότι δεν έχουμε σταθερή τιμή χρόνου για όλα τα πειράματα, αλλά υπάρχουν διακυμάνσεις. Στον πίνακα 3 παρουσιάζεται ο μέσος όρος των αποτελεσμάτων για κάθε μια από τις συναρτήσεις.

4.4.1 MiMC



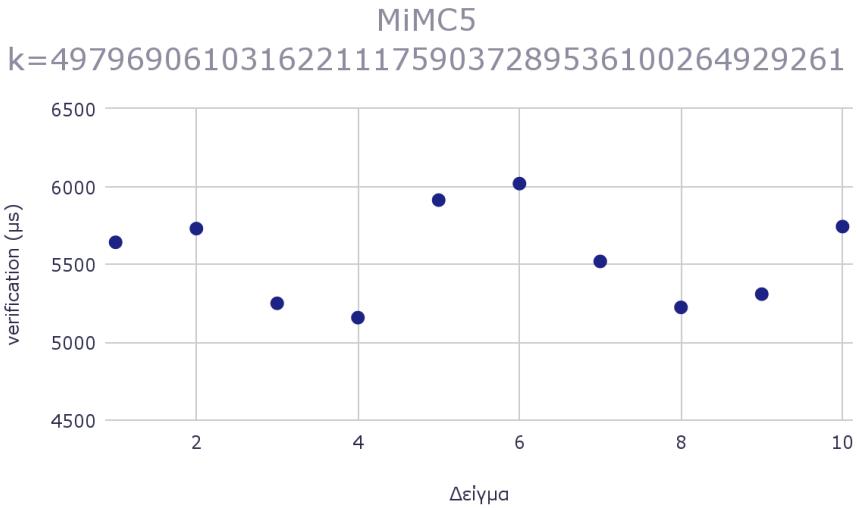
Διάγραμμα 8: Μετρήσεις χρόνων επαλήθευσης αποδείξεων με MiMC5, k = 2

Στο διάγραμμα 8 παρουσιάζονται τα 10 διαφορετικά δείγματα χρόνου για την επαλήθευση αποδείξεων με τη συνάρτηση MiMC5 με k=2. Οι περισσότερες τιμές κυμαίνονται ανάμεσα στα 5000μs και 6000μs. Παρατηρούμε ότι η τιμές παρουσιάζουν σχετικά έντονη διακύμανση.



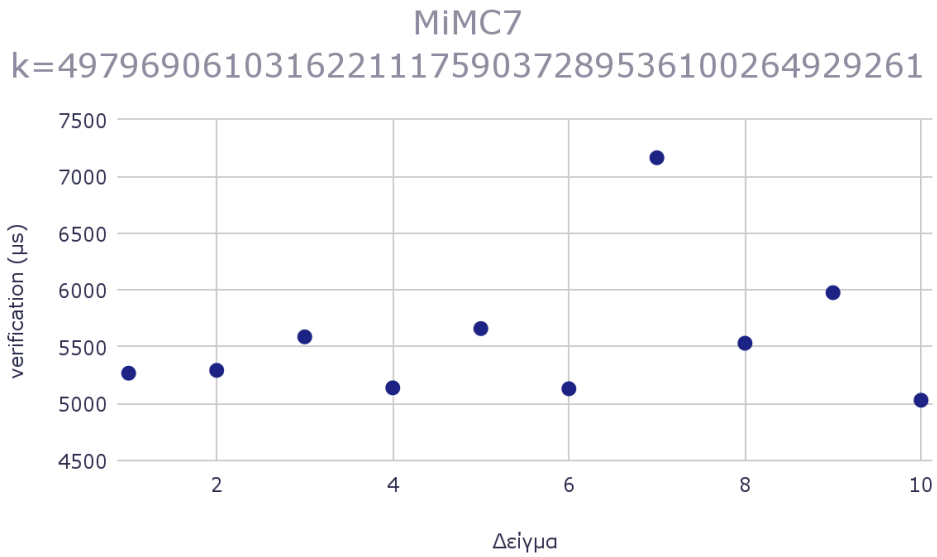
Διάγραμμα 9: Μετρήσεις χρόνων επαλήθευσης αποδείξεων με MiMC7, k = 2

Στο διάγραμμα 9 παρουσιάζονται τα 10 διαφορετικά δείγματα χρόνου για την επαλήθευση αποδείξεων με τη συνάρτηση MiMC7 με k=2. Οι περισσότερες τιμές κυμαίνονται ανάμεσα στα 5000μs και 6000μs. Παρατηρούμε ότι η τιμές είναι σχετικά συγκεντρωμένες γύρω από τον μέσο όρο και δεν παρουσιάζουν μεγάλη διακύμανση, εκτός του όγδοου δείγματος.



Διάγραμμα 10: Μετρήσεις χρόνων επαλήθευσης αποδείξεων με MiMC5, k = a

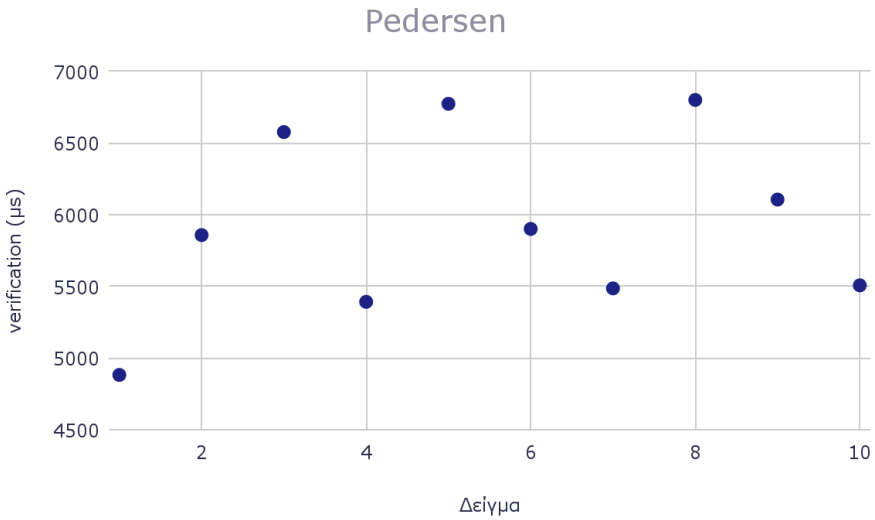
Στο διάγραμμα 10 παρουσιάζονται τα 10 διαφορετικά δείγματα χρόνου για την επαλήθευση αποδείξεων με τη συνάρτηση MiMC5 με k=α. Οι περισσότερες τιμές κυμαίνονται ανάμεσα στα 5000μs και 6000μs. Παρατηρούμε ότι η τιμές είναι σχετικά συγκεντρωμένες γύρω από τον μέσο όρο και δεν παρουσιάζουν μεγάλη διακύμανση.



Διάγραμμα 11: Μετρήσεις χρόνων επαλήθευσης αποδείξεων με MiMC7, k = a

Στο διάγραμμα 11 παρουσιάζονται τα 10 διαφορετικά δείγματα χρόνου για την επαλήθευση αποδείξεων με τη συνάρτηση MiMC7 με k=α. Οι περισσότερες τιμές κυμαίνονται ανάμεσα στα 5000μs και 5600μs. Παρατηρούμε ότι η τιμές είναι σχετικά συγκεντρωμένες γύρω από τον μέσο όρο και δεν παρουσιάζουν μεγάλη διακύμανση, εκτός του έβδομου δείγματος.

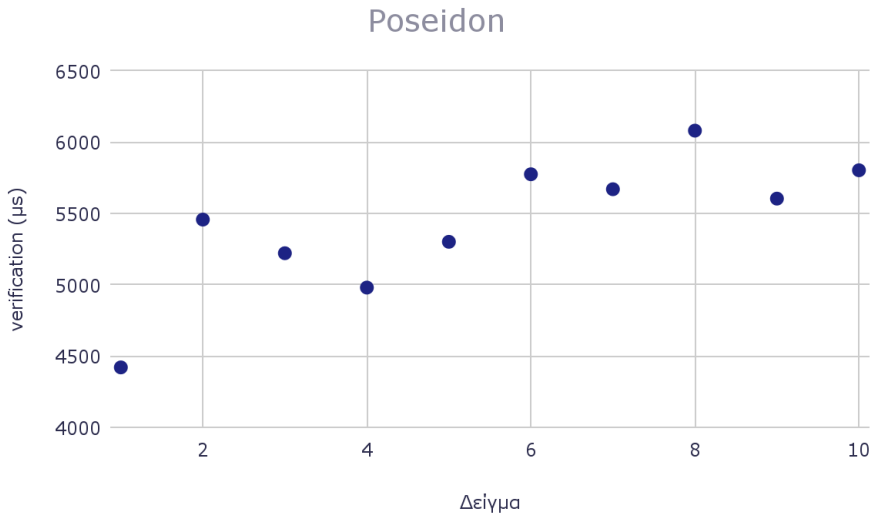
4.4.2 Pedersen



Διάγραμμα 12: Μετρήσεις χρόνων επαλήθευσης αποδείξεων με Pedersen

Στο διάγραμμα 12 παρουσιάζονται τα 10 διαφορετικά δείγματα χρόνου για την επαλήθευση αποδείξεων με τη συνάρτηση Pedersen. Οι περισσότερες τιμές κυμαίνονται ανάμεσα στα 5400μs και 6100μs. . Παρατηρούμε ότι η τιμές παρουσιάζουν σχετικά έντονη διακύμανση.

4.4.3 Poseidon



Διάγραμμα 13: Μετρήσεις χρόνων επαλήθευσης αποδείξεων με Poseidon

Στο διάγραμμα 13 παρουσιάζονται τα 10 διαφορετικά δείγματα χρόνου για την επαλήθευση αποδείξεων με τη συνάρτηση Poseidon. Οι περισσότερες τιμές κυμαίνονται ανάμεσα στα 5000μs και 6000μs. Παρατηρούμε ότι η τιμές είναι σχετικά συγκεντρωμένες γύρω από τον μέσο όρο και δεν παρουσιάζουν μεγάλη διακύμανση, εκτός του πρώτου δείγματος.

hash	parsing and verification (μs)	verification (μs)
poseidon	8663,3	5431,2
mimc5 k = 2	9181,3	5547,1
mimc7 k = 2	9213,6	5596
pedersen	9554,6	5928,4
mimc5 k = a	9349,2	5550,7
mimc7 k = a	8915,5	5578,4

Πίνακας 3 :Μέσος όρος χρόνου επαλήθευσης για κάθε hash

Παρατηρούμε από τον Πίνακα 3 ότι ο χρόνος επαλήθευσης αποδείξεων είναι αρκετά μικρός για όλες τις περιπτώσεις συναρτήσεων κατακερματισμού που χρησιμοποιούνται. Αυτό είναι ένα σημαντικό χαρακτηριστικό για τον σχεδιασμό ενός δικτύου Blockchain, καθώς είναι απαραίτητο

για την κλιμάκωσή του: Όσο οι χρήστες του δικτύου αυξάνονται, αυξάνεται έντονα και ο αριθμός τον εκτελούμενων συναλλαγών με αποτέλεσμα να απαιτούνται όλο και περισσότερες επιβεβαιώσεις αποδείξεων.

Είναι διακριτό ότι η Pedersen είναι πιο αργή από τις υπόλοιπες συναρτήσεις κατακερματισμού, ενώ τα ταχύτερα αποτελέσματα λαμβάνουμε με τη χρήση της Poseidon. Αυτό έρχεται σε συμφωνία με την περιπλοκότητα των κυκλωμάτων που αντιστοιχούν στις συναρτήσεις κατακερματισμού, αλλά και με το θεωρητικό υπόβαθρο της εργασίας αυτής.

Τέλος, οι χρόνοι του MiMC5 και MiMC7 φαίνεται ότι είναι ανεξάρτητοι από την τιμή της σταθεράς k, όπως συνέβη και με τους χρόνους παραγωγής της απόδειξης.

Κεφάλαιο 5

Συμπεράσματα και Μελλοντικές Προεκτάσεις

5.1 Συμπεράσματα

Δημιουργήσαμε ένα ολοκληρωμένο πληροφοριακό σύστημα, το οποίο αξιολογεί την αξιοπιστία ενός αλγορίθμου μηχανικής μάθησης μέσω επαλήθευσης αποδείξεων μηδενικής γνώσης σε περιβάλλοντα αλυσίδων-κορμού. Χρησιμοποιήσαμε με επιτυχία το πρωτόκολλο απόδειξης μηδενικής γνώσης για να εξασφαλίσουμε τη μυστικότητα ιδιωτικών παραμέτρων του δικτύου μας, διατηρώντας παράλληλα την αξιοπιστία και τον αποκεντρωμένο χαρακτήρα του. Η χρήση της πλατφόρμας Hyperledger Fabric μας επέτρεψε να φτιάξουμε μια προσαρμοσμένη αρχιτεκτονική δικτύου blockchain με ZKP.

Πιο συγκεκριμένα, αφού έχει γίνει η επικύρωση και ο έλεγχος των αλγορίθμων μηχανικής μάθησης, πραγματοποιείται η δημιουργία αποδείξεων μηδενικής γνώσης εκτός του δικτύου σύμφωνα με το πρωτόκολλο zk-SNARK. Ύστερα, γίνεται η υποβολή των απαραίτητων για την

επαλήθευση αρχείων στο δίκτυό μας και επαληθεύεται από το έξυπνο συμβόλαιο η εγκυρότητα της απόδειξης.

Η επαλήθευση με ZKP είναι σημαντική σε ένα τέτοιο σύστημα, καθώς μέσω αυτής γίνεται η απόδειξη και επιβεβαίωση της ορθής εκτέλεσης των αλγορίθμων μηχανικής μάθησης ενώ ταυτόχρονα προστατεύεται η ταυτότητά τους. Με αυτόν τον τρόπο, οι αλγόριθμοι που επιβεβαιώνονται καθίστανται αξιόπιστοι από την πλευρά του αποκεντρωμένου συστήματος. Αυτό δε θα ήταν εφικτό με τη χρήση μόνο συναρτήσεων κατακερματισμού για την επαλήθευση καθώς η ταυτότητα των αλγορίθμων θα προδιδόταν.

Συγκρίναμε διαφορετικές συναρτήσεις κατακερματισμού και εξαγάγαμε τους χρόνους που απαιτούνται για τη δημιουργία κυκλωμάτων σε ασφαλές περιβάλλον αλλά και τη δημιουργία και την επαλήθευση αποδείξεων. Από τη σύγκριση αυτή συμπεράναμε ότι η πολυπλοκότητα του αριθμητικού κυκλώματος συνδέεται με την απόδοση των συναρτήσεων κατακερματισμού κατά την εφαρμογή τους στο πρωτόκολλο zk-SNARK. Μάλιστα, δείξαμε ότι η συνάρτηση Pedersen έχει τους πιο αργούς χρόνους σε αντίθεση με τη συνάρτηση Poseidon που έχει τους γρηγορότερους. Επιπλέον, η τιμή της σταθεράς k της συνάρτησης MiMC δε φαίνεται να επηρεάζει ούτε τη δημιουργία ούτε την επαλήθευση των αποδείξεων.

Εκτός αυτών, δείξαμε ότι οι συναρτήσεις Poseidon, MiMC5, MiMC7, Pedersen είναι αρκετά αποδοτικές ώστε το εν λόγω πληροφοριακό σύστημα να μπορεί να κλιμακωθεί σε δίκτυο πολλών χρηστών ή οργανισμών .

5.2 Μελλοντικές Προεκτάσεις

Στην παρούσα διπλωματική εργασία, θεωρήσαμε ότι ο αλγόριθμος συναίνεσης Proof of Authority (PoA) θα εξυπηρετούσε τις ανάγκες της υλοποίησή μας όπως αναφέρεται αναλυτικότερα στο Κεφάλαιο 3. Ωστόσο, ο αλγόριθμος αυτός προϋποθέτει την ύπαρξη κάποιου συγκεκριμένου αριθμού κόμβων οι οποίοι θεωρούνται έμπιστοι και ως εκ τούτου τους δίνεται η δυνατότητα να εγκρίνουν συναλλαγές. Αν θέλουμε να πετύχουμε τη μέγιστη αποκέντρωση του δικτύου μας, προτείνεται να γίνει η κατάλληλη μελέτη σχετικά με το ποιος αλγόριθμος, από τους

γνωστούς ή μη, ταιριάζει στο συγκεκριμένο κατασκευασθέν πληροφοριακό σύστημα. Ταυτόχρονα, ενδιαφέρον παρουσιάζει ο τρόπος με τον οποίο θα γίνεται ο έλεγχος του αλγορίθμου μηχανικής μάθησης προτού αυτός εγκριθεί.

Οι αλγόριθμοι κατακερματισμού που εξετάσαμε προσέφεραν μικρούς χρόνους για δημιουργία και επαλήθευση αποδείξεων. Κατ' επέκταση, περισσότεροι αλγόριθμοι κατακερματισμού μπορούν να εξεταστούν στα παραπάνω ερευνητικά πλαίσια.

Στο συγκεκριμένο πληροφοριακό σύστημα επιλέξαμε το πρωτόκολλο zk-SNARKs, αφού δόθηκε έμφαση την ταχύτητα, στην απόδοση του δικτύου και στο μικρό μέγεθος αποδείξεων. Ωστόσο, ενδιαφέρον παρουσιάζει και το πρωτόκολλο zk-STARK, το οποίο προσφέρει μεγαλύτερη ασφάλεια από το zk-SNARK, διότι δεν απαιτείται δημιουργία ασφαλούς περιβάλλοντος για την παραγωγή των αποδείξεων. Επιπλέον, το zk-STARK εμφανίζει ανθεκτικότητα στην κβαντική υπολογιστική δύναμη (quantum security) κάτι το οποίο ενδέχεται να απασχολήσει μελλοντικά την επιστημονική κοινότητα.

Γλωσσάριο

Ελληνικός Όρος	Ξενόγλωσσος Όρος
άδεια	permission
αλγόριθμος συναίνεσης	consensus algorithm
αλήθεια	true
αναζήτησης κλειδιών	key query
αναπαράσταση πόρων	resource representations
αναρτώ	post
ανθεκτικό σε αποτυχία	crash fault-tolerant
ανθεκτικό σε βυζαντινά λάθη	byzantine fault tolerant
ανωνυμία	anonymity
απόδειξη	prove
απόδειξη εσωτερικού γινομένου	inner-product range proof
απόδειξης μηδενικής γνώσης	zero knowledge proof
αποκεντρωμένος λογιστικός κατάλογος	decentralized ledger
αποκέντρωση	decentralization
αποκλειστικό ή	exclusive or
αριθμητικο κύκλωμα	arithmetic circuit
αρχή	authority
ασφαλής εγκατάσταση	trusted set up
γλώσσες ειδικού τομέα	domain specific languages

δημόσιο κλειδί	public key
διαγράφω	delete
διαδικασίας δοκιμής και σφάλματος	trial and error
διατηρησιμότητα	persistency
διευθυνσιοδότηση πόρων	resource addressability
δίκτυο	network
δρομολογητές	routers
εγγραφή χρήστη	register user
έκδοση	version
ελεγξιμότητα	audibility
ενσωματώσεις	integrations
εξυπηρετητής	server
έξυπνο συμβόλαιο	smart contract
εξωτερικό γέμισμα	outer padding
επαλήθευση	verification
επαληθευτής	verifier
επικυρωτής	validator
επιχείρηση	enterprise
ιδιωτικό κλειδί	private key
καλώδια	wires
κατανεμημένος λογιστικός κατάλογος	distributed ledger
κατάσταση των κόμβων	world state
κεφαλίδα	header
κοινό κλειδί	shared key
κοινοπραξία	consortium
κόμβος	node
κρυπτογραφικός	cryptographic
κρυφό κλειδί	secret key
κώδικας αλυσίδας	chaincode
λαμβάνω	get
λογιστικός κατάλογος	ledger
μάρτυρας	witness
μερίδιο	stake
μεταγλωττιστής	compiler

μεταθέσεις	permutations
Μήνυμα	message
μονάδες άμεσης λειτουργίας	plug-and-play modules
μπλοκ	block
μπλοκ γεννήσεως	genesis block
ομοιόμορφη διεπαφή	uniform interface
παγκόσμια κατάσταση	global state
παραγωγή απόδειξης	proof generation
πελάτης	client
περιβάλλον αλυσίδας-κορμού	blockchain
πληρότητα	completeness
πολιτική αποδοχής	endorsement policy
πράξεις	operations
πύλες	gates
σταθερότητα	soundness
συγκεντρωτικός κατάλογος	centralized ledger
συμπίεση διανυσμάτων	vector compression
συνάρτηση κατακερματισμού	hash function
συνένωση	concatenation
σώμα	body
τοποθετώ	put
υβριδικό	hybrid
υποβολή απόδειξης	submit proof
υπόλοιπο	modulo
χρήση υπερμέσων	hypermedia utilization
χρονοσφραγίδα	timestamp
χώροι εργασίας	workspaces
ψευδο-εξυπηρετητές	mock-servers

Συντομεύσεις – Αρκτικόλεξα

API	Application Programming Interface
ZKP	Zero Knowledge Proof
ML	Machine Learning
AI	Artificial Intelligence
DLT	Distributed Ledger Technology
PoW	Proof of Work
PoS	Proof of Stake
DPoS	Delegated Proof of Stake
PBFT	Practical Byzantine Fault Tolerance
PoET	Proof of Elapsed Time
PoB	Proof of Burn
CFT	Crash Fault Tolerant
BFT	Byzantine Fault Tolerant
DSL	Domain Specific Language
MIT	Massachusetts Institute of Technology
Zk-SNARK	Zero-Knowledge Succinct Non-Interactive Argument of Knowledge
zk-STARK	Zero-Knowledge Scalable Transparent Argument of Knowledge
HMAC	Hash-Based Message Authentication Code
HTTPS	Hypertext Transfer Protocol Secure
SFTP	Secure File Transfer Protocol
FTPS	File Transfer Protocol Secure

MiMC	Minimal Multiplicative Complexity
Rest	Representational State Transfer
JSON	JavaScript Object Notation
XML	Extensible Markup Language
URIs	Uniform Resource Identifiers
IDMs	Identity Management Systems
GDPR	General Data Protection Regulation
CouchDB	Couch Database
NoSQL	Non Structured Query Language
CA	Certificate Authority

Βιβλιογραφία

[1] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system” 2008.

[2] C. Komalavalli, Deepika Saxena, Chetna Laroiya “Handbook of Research on Blockchain Technology” 2020.

[3] J. Abou Jaoude and R. George Saade, "Blockchain Applications – Usage in Different Domains," in IEEE Access, vol. 7, pp. 45360-45381, 2019, doi: 10.1109/ACCESS.2019.2902501.

[4] Kadam, Suvarna. Review of Distributed Ledgers: The technological Advances behind cryptocurrency. (2018)

[5] https://www.finra.org/sites/default/files/2017_BC_Byte.pdf?fbclid=IwAR13RZj0BlkTPQ2_41nshQ7U80e9potj3elLxIuGxuIwwTBXq13mUvi4uaY

[6] Zheng, Zibin & Xie, Shaoan & Dai, Hong-Ning & Chen, Xiangping & Wang, Huaimin. (2017). An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. 10.1109/BigDataCongress.2017.85.

[7] Aithal, P & Saavedra, P & Aithal, Sreeramana & Ghoash, Surajit. (2021). Blockchain Technology and its Types-A Short Review. International Journal of Applied Science and Engineering. 9. 189-200.

[8] https://www.foley.com/en/insights/publications/2021/08/types-of-blockchain-public-private-between

[9] Strehle, Elias. (2020). Public Versus Private Blockchains.

[10] Shijie Zhang, Jong-Hyouk Lee, “Analysis of the main consensus protocols of blockchain”,ICT Express, Volume 6, Issue 2,2020,ISSN 2405-9595, https://doi.org/10.1016/j.icte.2019.08.001.

[11] Azbeg, Kebira & Ouchetto, Ouaïl & jai andaloussi, Said & Laila, Fetjah. (2020). An Overview of Blockchain Consensus Algorithms: Comparison, Challenges and Future Directions. 10.1007/978-981-15-6048-4_31.

[12] https://hyperledger-fabric.readthedocs.io/en/latest/whatis.html

[13] Azeez, Nureni & Chinazo, Onyema. (2018). ACHIEVING DATA AUTHENTICATION WITH HMAC-SHA256 ALGORITHM..

[14] Hasan, Jahid. (2019). Overview andApplications of Zero Knowledge Proof (ZKP). 8. 5.

[15] https://ethereum.org/en/zero-knowledge-proofs/

[16] Bunz, Benedikt & Bootle, Jonathan & Boneh, Dan & Poelstra, Andrew & Wuille, Pieter & Maxwell, Greg. (2018). Bulletproofs: Short Proofs for Confidential Transactions and More. 315-334. 10.1109/SP.2018.00020.

[17] G. Jeong, N. Lee, J. Kim and H. Oh, "Azeroth: Auditable Zero-Knowledge Transactions in Smart Contracts," in IEEE Access, vol. 11, pp. 56463-56480, 2023, doi: 10.1109/ACCESS.2023.3279408.

[18] Ben-Sasson, Eli et al. “Scalable, transparent, and post-quantum secure computational integrity.” *IACR Cryptol. ePrint Arch.* 2018 (2018): 46.

[19] https://docs.circom.io/

[20] Denton, Ben & Adhami, R. (2012). Modern Hash Function Construction.

[21] Rodriguez, Carlos & Baez, Marcos & Daniel, Florian & Casati, Fabio & Trabucco, Juan & Canali, Luigi & Percannella, Gianraffaele. (2016). REST APIs: A Large-Scale Analysis of Compliance with Principles and Best Practices. 21-39. 10.1007/978-3-319-38791-8_2.

[22] https://www.postman.com/product/what-is-postman/

[23] https://blog.postman.com/continuous-api-testing-with-postman/

[24] Albrecht, Martin & Grassi, Lorenzo & Rechberger, Christian & Roy, Arnab & Tiessen, Tyge. (2016). MiMC: Efficient Encryption and Cryptographic Hashing with Minimal Multiplicative Complexity. 191-219. 10.1007/978-3-662-53887-6_7.

[25] Grassi, L., Khovratovich, D., Schofnegger, M. (2023). POSEIDON2: A Faster Version of the POSEIDON Hash Function. In: El Mrabet, N., De Feo, L., Duquesne, S. (eds) Progress in Cryptology - AFRICACRYPT 2023. AFRICACRYPT 2023. Lecture Notes in Computer Science, vol 14064. Springer, Cham. https://doi.org/10.1007/978-3-031-37679-5_8

[26] Jordi Baylina1 and Marta Belles, iden3, 2Universitat Pompeu Fabra, 4-bit Window Pedersen Hash On The Baby Jubjub Elliptic Curve

[27] Tyagi, Shobha & Kathuria, Madhumita. (2022). Role of Zero-Knowledge Proof in Blockchain Security. 738-743. 10.1109/COM-IT-CON54601.2022.9850714.

[28] Bai, Tianyu, Yangsheng Hu, Jianfeng He, Hongbo Fan, and Zhenzhou An. 2022. "Health-zkIDM: A Healthcare Identity System Based on Fabric Blockchain and Zero-Knowledge Proof" *Sensors* 22, no. 20: 7716. https://doi.org/10.3390/s22207716

[29] S. Gao, Z. Peng, F. Tan, Y. Zheng and B. Xiao, "SymmeProof: Compact Zero-Knowledge Argument for Blockchain Confidential Transactions," in IEEE Transactions on Dependable and Secure Computing, vol. 20, no. 3, pp. 2289-2301, 1 May-June 2023, doi: 10.1109/TDSC.2022.3179913.

[30] https://github.com/iden3/go-circom-prover-verifier

[31] Groth, Jens. (2016). On the Size of Pairing-Based Non-interactive Arguments. 305-326. 10.1007/978-3-662-49896-5_11.