ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΥΠΟΛΟΓΙΣΤΩΝ

ΔΗΜΙΟΥΡΓΙΑ ΜΙΑΣ ΑΠΟΚΕΝΤΡΩΜΕΝΗΣ ΕΦΑΡΜΟΓΗΣ ΒΑΣΙΣΜΕΝΗΣ ΣΤΟ ETHEREUM BLOCKCHAIN

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΛΕΞΑΝΔΡΟΣ Δ. ΑΘΑΝΑΣΟΠΟΥΛΟΣ

Επιβλέπων: Νικόλαος Παπασπύρου
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2019

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΥΠΟΛΟΓΙΣΤΩΝ

ΔΗΜΙΟΥΡΓΙΑ ΜΙΑΣ ΑΠΟΚΕΝΤΡΩΜΕΝΗΣ ΕΦΑΡΜΟΓΗΣ ΒΑΣΙΣΜΕΝΗΣ ΣΤΟ ETHEREUM BLOCKCHAIN

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΑΛΕΞΑΝΔΡΟΣ Δ. ΑΘΑΝΑΣΟΠΟΥΛΟΣ

Επιβλέπων: Νικόλαος Παπασπύρου
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 22η Ιουλίου 2019.

.....	.....	.....
Νικόλαος Παπασπύρου Καθηγητής Ε.Μ.Π.	Αριστείδης Παγουρτζής Αν. Καθηγητής Ε.Μ.Π.	Γεώργιος Γκούμας Επίκ. Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2019

.....

Αλέξανδρος Δ. Αθανασόπουλος
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικών Υπολογιστών Ε.Μ.Π.

Περίληψη

Σκοπός αυτής της διπλωματικής εργασίας είναι η δημιουργία μιας εφαρμογής, μέσω της οποίας θα μπορεί οποιαδήποτε εταιρεία ηλεκτρονικών παιχνιδιών να εισάγει στο ρεπερτόριο των αγώνων μεταξύ των παιχτών της, τους ανταγωνιστικούς αγώνες πραγματικών χρημάτων. Για το σκοπό αυτό δημιουργήσαμε μια αποκεντρωμένη εφαρμογή, η οποία βασίζεται στην πλατφόρμα του Ethereum, που προσφέρει ασφάλεια και αξιοπιστία σε όλους τους χρήστες της. Η εφαρμογή αυτή περιλαμβάνει τη δημιουργία διάφορων tokens, τα οποία χρησιμοποιούνται ως ανταλλακτικό μέσο για όλες τις συναλλαγές.

Το βασικό token της εφαρμογής, ονόματι gaming token, αφορά τους επενδυτές της εφαρμογής και αποτελεί, στην ουσία μέσο αναπαράστασης των μετοχών της. Ακολουθεί το ERC-20 πρωτόκολλο, το οποίο είναι και το πιο ευρέως χρησιμοποιούμενο πρωτόκολλο για token στο Ethereum με κάποιες προσθήκες, που έχουν σκοπό να τελειοποιήσουν τη λειτουργία του πάνω στη συγκεκριμένη εφαρμογή.

Για κάθε παιχνίδι που γίνεται μέλος της εφαρμογής, παρέχεται ένα νέο token, το οποίο αφορά μόνο το συγκεκριμένο παιχνίδι. Τα tokens αυτού του είδους δεν είναι διαθέσιμα για επενδύσεις και χρησιμοποιούνται μόνο για τις συναλλαγές εντός του παιχνιδιού, το οποίο αφορούν.

Τα συμβόλαια που αφορούν την εφαρμογή αυτή γράφτηκαν σε solidity, με τις προσθήκες κάποιου κώδικα EVM, όπου αυτό χρειαζόταν. Χρησιμοποιήθηκε το web3 για την επικοινωνία του frontend (διαδικτυακή εφαρμογή) με το backend (Ethereum blockchain).

Μας απασχόλησαν όλες οι λειτουργίες του Ethereum blockchain, όπως ο αλγόριθμος κρυπτογράφησής του, δηλαδή ο SHA-3, ο τρόπος αποθήκευσης των δεδομένων, δηλαδή τα Modified Merkle Patricia tries και ο τρόπος πραγματοποίησης των εξορύξεων (proof of work), ώστε να υπάρχει μια πλήρης ιδέα σχετικά με τις διαδικασίες που ακολουθούνται σε κάθε βήμα του backend της εφαρμογής. Δεν έχει χρησιμοποιηθεί βάση δεδομένων, αλλά γίνεται χρήση των δεδομένων αποκλειστικά μέσω του blockchain, ώστε η εφαρμογή αυτή να είναι πλήρως αποκεντρωμένη.

Λέξεις κλειδιά

Ethereum, blockchain, αποκεντρωμένη εφαρμογή, Έξυπνα συμβόλαια, token, ERC-20, gaming token, συναλλαγές, επενδύσεις, solidity, web3, frontend, backend, SHA-3, Modified Merkle Patricia Tries, εξόρυξη, proof of work, βάση δεδομένων.

Abstract

The purpose of this diploma thesis is to create an application, through which any gaming company can add into the repertoire of matches between its players, competitive real money games. To that end, we created a decentralized application based on the Ethereum platform, that offers security and reliability to all its users. This application includes the creation of various tokens, which are used as a replacement for all transactions.

The application's main token, called gaming token, is to be used by application investors and is essentially a means of representing the application's shares. This token is an ERC-20 protocol token, which is the most widely used token protocol in Ethereum, with some additions designed to make it the perfect match for this application.

For each game, that is part of the application, a new token is provided, which only applies to that game. Tokens of this kind are not available for investment, but are only used for the in-game transactions of the game they refer.

The smart contracts related to this application were written in solidity, with the additions of some EVM code where it was necessary. Web3 was used to make communication between the frontend (web application) and the backend (Ethereum blockchain), possible.

We have been involved in all parts of Ethereum blockchain, such as its encryption algorithm, SHA-3, the way data is stored, namely Modified Merkle Patricia Tries, and the way mining takes place (proof of work), so that we have a complete idea of the process followed in each step of the application's backend. No database has been used, but all data is used exclusively through the Ethereum blockchain, so as this application to be fully decentralized.

Keywords

Ethereum, blockchain, decentralized application, smart contracts, token, ERC-20, gaming token, transactions, investment, solidity, web3, frontend, backend, SHA-3, Modified Merkle Patricia Tries, mining, proof of work, database.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή της διπλωματικής μου κ. Νικόλαο Παπασπύρου γιατί μου έδωσε την ευκαιρία να ασχοληθώ με πολλά ενδιαφέροντα μαθήματα και ένα ενδιαφέρον θέμα διπλωματικής. Επίσης, ευχαριστώ τον καθηγητή κ. Αριστείδη Παγουρτζή για την προθυμία να βοηθήσει στην εκπόνηση της διπλωματικής μου εργασίας. Τέλος, ευχαριστώ όλους τους φίλους μου, την κοπέλα μου, Ιωάννα και την οικογένειά μου για όλες τις υπέροχες στιγμές που περάσαμε μαζί, καθώς και για τη στήριξή τους όλα αυτά τα χρόνια.

Αλέξανδρος Δ. Αθανασόπουλος,
Αθήνα, 12 Ιουλίου 2019

Περιεχόμενα

Κατάλογος πινάκων

Κατάλογος σχημάτων

1 Εισαγωγή

1.1 Σκοπός

Σκοπός αυτής της διπλωματικής εργασίας είναι η δημιουργία ενός Decentralized Application (DAPP), το οποίο σχετίζεται με τα ηλεκτρονικά παιχνίδια. Πιο συγκεκριμένα, αφορά τη δημιουργία ενός token (ERC20 protocol), το οποίο βασίζεται στο Ethereum Blockchain και μέσω του οποίου μια εταιρεία ηλεκτρονικών παιχνιδιών, μπορεί να εισάγει στο matchmaking της, τον πιο ανταγωνιστικό τρόπο παιχνιδιού, δηλαδή αυτόν στον οποίο οι παίχτες ανταγωνίζονται μεταξύ τους με πραγματικά χρήματα. Επίσης, το token αυτό είναι διαθέσιμο στην αγορά για τους διάφορους επενδυτές. Θα μελετήσουμε τον τρόπο δημιουργίας ενός DAPP, τόσο από τη σκοπιά της επικοινωνίας με το Ethereum Blockchain μέσω έξυπνων συμβολαίων (smart contracts), που αποτελεί το backend της εφαρμογής, όσο και από τη σκοπιά της επικοινωνίας του χρήστη με το web application, το οποίο αποτελεί το frontend της εφαρμογής. Επιπλέον, θα μελετήσουμε τα διάφορα εργαλεία και γλώσσες προγραμματισμού, οι οποίες καθιστούν δυνατή τη δημιουργία ενός DAPP. Τέλος, θα αναφερθεί η οικονομική φύση ενός blockchain application, καθώς και πώς βάσει όλων των προαναφερθέντων δημιουργήθηκε το συγκεκριμένο DAPP.

1.2 Ιστορία του blockchain και των κρυπτονομισμάτων

Το blockchain είναι μια συνεχώς αυξανόμενη λίστα από αρχεία, τα λεγόμενα blocks. Κάθε block είναι συνδεδεμένο με το προηγούμενό του, περιέχοντας έναν κρυπτογραφημένο κατακερματισμό (cryptographic hash) του. Συνολικά, το block περιέχει αυτόν τον κατακερματισμό, μία χρονοσφραγίδα και δεδομένα, τα οποία σε ένα blockchain κρυπτονομίσματος, είναι δεδομένα συναλλαγών, αποθηκευμένα συνήθως σαν ένα Merkle Tree. Προφανώς υπάρχει ένα block, το πρώτο, που ονομάζεται genesis block, το οποίο δεν περιέχει κρυπτογραφημένο κατακερματισμό του προηγούμενού του, καθώς δεν υπάρχει προηγούμενο. Πλεονέκτημα του blockchain είναι ότι επιτρέπει σε χρήστες από διάφορες μεριές του κόσμου να φτάσουν σε ομοφωνία, χωρίς ποτέ να έχουν επικοινωνήσει μεταξύ τους.

Η ιδέα του blockchain ξεκινά το μακρινό 1991, όπου δύο ερευνητές, ο Stuart Haber και ο W. Scott Stornetta, παρουσίασαν μια λύση στο πρόβλημα ψηφιακών δεδομένων με χρονοσφραγίδες. Συγκεκριμένα, σκοπός τους ήταν να είναι πάντα ασφαλή και σε σωστή χρονική σειρά. Έτσι, δημιούργησαν μια κρυπτογραφημένη αλυσίδα από blocks, κάθε ένα εκ των οποίων περιείχε ένα χρονοσφραγισμένο ψηφιακό έγγραφο. Το 1992, όταν και εισήγαγαν τα Merkle Trees για πρώτη φορά σε πραγματικό έργο, το σύστημά τους έγινε ακόμη πιο αποδοτικό καθώς μπορούσε πλέον να περιέχει σε κάθε block περισσότερα του ενός ψηφιακά έγγραφα. Ωστόσο, η τεχνολογία αυτή δεν χρησιμοποιήθηκε ευρέως και έτσι υπήρξε ένα μεγάλο χρονικό κενό έως ότου ξαναπραγματοποιηθεί κάποιο βήμα προς το blockchain, έτσι όπως το γνωρίζουμε στις μέρες μας. Το 2004, ο Harold Thomas Finney, δημιούργησε το RPoW (Reusable Proof of Work), ώστε να λύσει το πρόβλημα της διπλής σπατάλης, δηλαδή το πρόβλημα όπου ένα ψηφιακό αντικείμενο αξίας μπορούσε να σπαταληθεί παραπάνω από μία φορά. Συγκεκριμένα, το σύστημα έπαιρνε ένα μη ανταλλάξιμο Hash-Cash token βασιζόμενο στην απόδειξη εργασίας (proof of work) και δημιουργούσε ένα άλλο κρυπτογραφικά υπογεγραμμένο (RSA-signed) token, το οποίο μπορούσε να μεταφερθεί από ένα χρήστη σε έναν άλλο. Το μεγάλο βήμα του

RPoW όμως δεν ήταν αυτό. Ήταν το πρώτο σύστημα το οποίο χρησιμοποιούσε για ασφάλεια, κάτι παρόμοιο με το σημερινό blockchain. Συγκεκριμένα, είχε έναν ασφαλή server, μέσω του οποίου χρήστες από όλο τον κόσμο μπορούσαν να επαληθεύσουν την ορθότητα και ακεραιότητα μιας πραγματικής συναλλαγής.

Και κάπως έτσι φτάνουμε στη σύνδεση του blockchain με τα κρυπτονομίσματα. Στα τέλη του 2008, δημοσιεύτηκε σε μια λίστα αλληλογραφίας κρυπτογράφων, ένα 'white paper', το οποίο εισήγαγε ένα αποκεντρωμένο διομότιμο (peer-to-peer) ηλεκτρονικό χρηματικό σύστημα το πασίγνωστο πλέον Bitcoin, υπογεγραμμένο με το ψευδώνυμο 'Satoshi Nakamoto'. Βασιζόταν στον αλγόριθμο του RPoW , αλλά αντί να περιέχει κάποιον ασφαλή server, το πρόβλημα της διπλής σπατάλης αποφευγόταν μέσω ενός αποκεντρωμένου διομότιμου πρωτοκόλλου. Αυτό ήταν και το πρώτο κρυπτονόμισμα, το οποίο δημιουργήθηκε στην ιστορία. Το πρώτο block δημιουργήθηκε από τον/τους 'Satoshi Nakamoto' στις 9 Ιανουαρίου του 2009, το οποίο είχε έπαθλο 50 Bitcoin. Στις 12 Ιανουαρίου του 2009, ο Harold Thomas Finney, ο δημιουργός του RPoW ήταν ο πρώτος αγοραστής καθώς αγόρασε από τον/τους 'Satoshi Nakamoto' 10 Bitcoins. Αυτή ήταν η πρώτη συναλλαγή κρυπτονομίσματος στην ιστορία.

Το 2013, ο συνιδρυτής του περιοδικού 'Bitcoin Magazine', Vitalic Buterin, ισχυρίστηκε ότι το Bitcoin χρειάζεται κάποια βοηθητική γλώσσα προγραμματισμού, ώστε να καθίσταται δυνατή στους διάφορους προγραμματιστές η δημιουργία Αποκεντρωμένων Εφαρμογών, δηλαδή Decentralized Applications (DAPPS). Λόγω, όμως, του ότι δεν κατάφερε να έρθει σε συμφωνία με την υπόλοιπη κοινότητα, ξεκίνησε με τη βοήθεια του Anthony Di Iorio και κάποιων επενδυτών, την ανάπτυξη ενός άλλου κρυπτονομίσματος, βασιζόμενου και πάλι στο blockchain, του Ethereum. Εισήγαγε ένα καινούριο χαρακτηριστικό, τα λεγόμενα έξυπνα συμβόλαια (smart contracts), δηλαδή προγράμματα τα οποία αναπτύσσονται και εκτελούνται στο ίδιο το Ethereum Blockchain. Μέσω των έξυπνων συμβολαίων ο δρόμος για τη δημιουργία DAPPS άνοιξε για τα καλά. Περισσότερες πληροφορίες για τα έξυπνα συμβόλαια (smart contracts), καθώς και για τις αποκεντρωμένες εφαρμογές (DAPPS) δίνονται σε επόμενη ενότητα.

Πλέον, όλο και περισσότερα κρυπτονομίσματα βασιζόμενα στο blockchain δημιουργούνται, ενώ επίσης το blockchain χρησιμοποιείται ευρέως και σε ποικίλες άλλες εφαρμογές.

1.3 Ιστορία των έξυπνων συμβολαίων (smart contracts)

Έξυπνο συμβόλαιο (smart contract) είναι ένας κώδικας προγράμματος ηλεκτρονικού υπολογιστή, ο οποίος μπορεί να εκτελεστεί και μέσω αυτής της εκτέλεσης να επιβάλει τη διαπραγμάτευση μιας συμφωνίας ή σύμβασης χωρίς τη χρήση κάποιου εξωτερικού συνεργάτη (συμβολαιογράφου κλπ.). Τα έξυπνα συμβόλαια των κρυπτονομισμάτων χρησιμοποιούν την τεχνολογία του blockchain, ώστε να φτάσουν στο επιθυμητό αποτέλεσμα. Η όλη διαδικασία είναι αυτοματοποιημένη και μπορεί να λειτουργήσει ως συμπλήρωμα ή υποκατάστατο για νομικές συμβάσεις, όπου οι όροι της σύμβασης καταγράφονται σε μια γλώσσα υπολογιστών ως σύνολο οδηγιών.

Η όλη ιδέα για χρήση έξυπνων συμβολαίων πρωτοπροτάθηκε από έναν επιστήμονα υπολογιστών και κρυπτογράφο, τον Nick Szabo, το μακρινό 1994. Συγκεκριμένα, είχε αναφέρει ότι θα ήταν χρήσιμο να υπάρχει ένα σύνολο υποσχέσεων, καθορισμένων σε ψηφιακή μορφή, συμπεριλαμβανομένων ίσως κάποιων βασικών πρωτοκόλλων, μέσω του

οποίου τα μέλη του να μπορούν να αλληλεπιδρούν με τις υποσχέσεις αυτές. Αν και αποτελεί καθοριστικό βήμα για τη δημιουργία έξυπνων συμβολαίων, καθώς η ιδέα παραμένει ίδια και στη σημερινή εκδοχή τους, η εξέλιξή τους πάγωσε λόγω της έλλειψης κάποιας απαραίτητης τεχνολογίας, μέσω της οποίας θα μπορούσαν να εφαρμοστούν. Με τον ερχομό των κρυπτονομισμάτων, το 2008, και συγκεκριμένα λόγω της δημιουργίας τους μέσω του blockchain, η βάση για τη χρησιμοποίηση των έξυπνων συμβολαίων δημιουργήθηκε. Αν και η πρόταση του Vitalic Buterin, όπως προαναφέρθηκε δεν προχώρησε για το Bitcoin, ο ίδιος, αποφάσισε να δημιουργήσει το δικό του κρυπτονόμισμα, Ethereum, το οποίο είχε ως βασική πρωτοπορία τη χρήση έξυπνων συμβολαίων. Εδώ αξίζει να αναφερθεί ότι και το Bitcoin έχει τη δυνατότητα δημιουργίας έξυπνων συμβολαίων στο blockchain του, απλά όχι με τόσες δυνατότητες όπως το blockchain του Ethereum.

Σήμερα, πολλά νέα κρυπτονομίσματα παρέχουν τη δυνατότητα έξυπνων συμβολαίων. Το πιο βασικό, όμως, παραμένει το Ethereum. Τα έξυπνα συμβόλαια, όπως γίνεται εύκολα αντιληπτό, ανοίξαν το δρόμο προς τη δημιουργία συμβάσεων και συναλλαγών μεταξύ ανώνυμων ατόμων, χωρίς την παροχή της ασφάλειας και εμπιστευτικότητας από κάποιο τρίτο πρόσωπο, αλλά από την τεχνολογία του Blockchain. Τέλος, κατέστησαν δυνατή τη δυνατότητα εύκολης δημιουργίας tokens, καθώς και αποκεντρωμένων εφαρμογών (DAPPS).

1.4 Ιστορία των tokens κρυπτονομισμάτων

Για να μπορέσουμε να εξηγήσουμε τι ακριβώς είναι ένα token κρυπτονομίσματος, πρέπει πρώτα να εξηγήσουμε την έννοια ενός κρυπτονομίσματος. Καταρχάς, τα κρυπτονομίσματα είναι περιουσιακά στοιχεία με τη μορφή ψηφιακών νομισμάτων, όπως ακριβώς και τα φυσικά χρήματα, τα οποία είναι υπαρκτά όμως, μόνο μέσα στο δικό τους blockchain. Οι συναλλαγές των ψηφιακών αυτών νομισμάτων μπορούν να γίνονται από άτομο σε άτομο, χωρίς όμως να μετακινούνται όπως τα φυσικά χρήματα όταν κάποιος στέλνει ή λαμβάνει. Όλα αυτά τα ψηφιακά νομίσματα καταχωρούνται ως δεδομένα σε μία γιγαντιαία παγκόσμια βάση δεδομένων, το λεγόμενο blockchain. Η ορθότητα αυτής της βάσης δεδομένων ελέγχεται και επαληθεύεται από υπολογιστές σε όλον τον κόσμο, με τρόπο, ο οποίος θα μελετηθεί εκτενώς στο δεύτερο κεφάλαιο. Οπότε, όταν μιλάμε για κρυπτονόμισμα (coin) μπορούμε στο περίπου να το φανταζόμαστε ως αληθινό χρήμα, απλά σε ψηφιακή μορφή.

Τα tokens κρυπτονομισμάτων, από την άλλη, δεν έχουν το δικό τους blockchain, αλλά ζουν και υπάρχουν στο blockchain ενός κρυπτονομίσματος, ενώ η αξία τους καθορίζεται βάσει του εκάστοτε κρυπτονομίσματος. Ένα token μοιάζει με ένα προϊόν (π.χ. αυτοκίνητο), το οποίο έχει αξία σε χρήματα, αλλά δεν είναι χρήματα. Η δημιουργία ενός token απαιτεί σπατάλη του κρυπτονομίσματος, μέσα στου οποίου το blockchain θέλει να δημιουργηθεί, όπως ακριβώς και η αγορά ενός προϊόντος απαιτεί σπατάλη χρημάτων.

Τα tokens όπως και τα coins αγοράζονται σε διάφορα site συναλλάγματος, είτε απευθείας από το δημιουργό, είτε από κάποιον προηγούμενο αγοραστή. Με λίγα λόγια, όταν ένα token δημιουργείται, η πόσοτητα των διαθέσιμων tokens και το σύνολο των δημιουργημένων tokens δίνονται σε κάποιο λογαριασμό, συνήθως στο δημιουργό τους. Τα tokens, στην ουσία, αρχικά δεν έχουν αξία, αλλά την αποκτούν όταν ξεκινά η πώλησή τους. Η αρχική έξοδος των tokens στην αγορά είναι γνωστή ως ‘Initial Coin Offering’

(ICO), όπου ο αρχικός κάτοχος των tokens αποφασίζει πόσα tokens θα δοθούν προς πώληση στην αγορά, την τιμή τους, καθώς και αν η τιμή θα παραμένει σταθερή ή αν αυτή θα αυξάνεται, βάσει της ζήτησής του στην αγορά. Αφότου τελειώσει η ICO, η αξία του εκάστοτε token μπορεί να αλλάξει προς το καλύτερο ή προς το χειρότερο, σύμφωνα με τη γνωστή αρχή της προσφοράς και της ζήτησης, που ισχύει στο χρηματιστήριο. Γενικά, τα tokens μπορεί να αναπαριστούν οτιδήποτε μέσα στο οικοσύστημα στο όποίο ανήκουν (blockchain του κρυπτονομίσματος), ενώ μπορεί να έχουν παραπάνω του ενός ρόλου μέσα στο οικοσύστημά τους. Έχουν κάποιο λόγο ύπαρξης και αντιπροσωπεύουν κάποιο προϊόν ή υπηρεσία. Για να μπορέσουμε να εξηγήσουμε περαιτέρω τα tokens, θα πρέπει να αναφέρουμε τους τύπους των tokens, ενώ για να γίνει αυτό θα πρέπει να αναφερθεί ένα ιστορικό γεγονός.

Περιληπτικά, το 1946 δύο εταιρείες με έδρα τη Φλώριντα, προσέφεραν συμβάσεις ακινήτων για εκτάσεις γης. Όμως, προσέφεραν ταυτόχρονα τη δυνατότητα, οι αγοραστές να μεταβιβάσουν εκ νεόυ τους αγρούς στις εταιρείες, οι οποίες είχαν τη γνώση να τους αναπτύξουν και ως αντάλλαγμα θα τους επιστρέφονταν μέρος των κερδών. Ωστόσο, οι εταιρείες κατηγορήθηκαν ότι η επένδυση αυτή ήταν παράνομη και οδηγήθηκαν στα δικαστήρια. Εκεί, πάρθηκε μια απόφαση ορόσημο για τα δεδομένα των επενδύσεων. Αναπτύχθηκε μια δοκιμή μέσω της οποίας καθορίζεται αν μια συναλλαγή είναι ή όχι σύμβαση επένδυσης, γνωστή ως Howey Test. Για να περάσει μια συναλλαγή το Howey Test και να θεωρηθεί σύμβαση επένδυσης πρέπει να πληρεί όλες τις παρακάτω προϋποθέσεις:

- Να είναι επένδυση χρημάτων
- Να υπάρχει προσδοκία για κέρδη από την επένδυση
- Η επένδυση των χρημάτων να είναι σε μία κοινή επιχείρηση
- Οποιοδήποτε κέρδος να προέρχεται από τις προσπάθειες κάποιων εργατών ή κάποιου εξωτερικού συνεργάτη

Ας έρθουμε τώρα στις κατηγορίες των tokens. Υπάρχουν τέσσερις κατηγορίες tokens, εκ των οποίων οι βασικές είναι έντονα σκιαγραφισμένες (bold):

- **Tokens εγγυήσεων (Security Tokens):**
  Τα περισσότερα tokens που εκδίδονται είναι tokens της συγκεκριμένης κατηγορίας. Το πρόσωπο που επενδύει στα tokens αυτά το κάνει με προσδοκία κάποιου κέρδους. Πρόκειται για τα tokens, που περνούν το Howay Test, οπότε αντιμετωπίζονται από τη νομοθεσία όπως και οι κανονικές χρηματοοικονομικές εγγυήσεις. Ορίζονται ως συμβάσεις επενδύσεων που αντιπροσωπεύουν νόμιμη ιδιοκτησία ενός φυσικού ή ψηφιακού στοιχείου που έχει επαληθευτεί μέσα στο blockchain. Η αξία του token είναι άρρηκτα συνδεδεμένη με την αντίστοιχη της εταιρείας (ονομαστική αξία). Τα security tokens είναι το πιο κοντινό είδος token στις πραγματικές μετοχές, λόγω και της νομιμότητάς τους.

- **Tokens χρησιμότητας (Utility Tokens):**
  Λέγονται και applications tokens, καθώς χρησιμοποιούνται κατά κόρον για τη δημιουργία αποκεντρωμένων εφαρμογών (DAPPS). Χρησιμοποιούνται ώστε να παρέχουν στους αγοραστές, κάποιο προϊόν ή υπηρεσία. Το συγκεκριμένο είδος token δεν έχει σχέση με την ονομαστική αξία της εταιρείας και πολλές φορές δεν υπάρχει καν εταιρεία. Δημιουργείται ένα εσωτερικό οικονομικό περιβάλλον στο blockchain της εφαρμογής, όπου και χρησιμοποιείται. Βασίζεται στο proof-of-

stake, δηλαδή όσα περισσότερα tokens έχει κάποιος, τόσο μεγαλύτερη επίδραση έχει η γνώμη του στη μελλοντική ανάπτυξη της εφαρμογής. Οπότε, κάποιος που επενδύει σε τέτοιες εφαρμογές, εκτός του κέρδους, λαμβάνει μέρος και στις μελλοντικές αποφάσεις. Το GamingToken, δηλαδή το token που δημιουργήθηκε στα πλαίσια της συγκεκριμένης διπλωματικής εργασίας, ανήκει στην κατηγορία των Utility Tokens.

- Tokens πληρωμών (Payment Tokens):
  Ο μοναδικός λόγος ύπαρξης των συγκεκριμένων tokens είναι για την πληρωμή κάποιου προϊόντος ή υπηρεσίας.

- Tokens μετοχικών κεφαλαίων (Equity Tokens):
  Τα tokens αυτά αντιπροσωπεύουν μετοχές των εταιρείων στις οποίες ανήκουν, ωστόσο λίγα τέτοια tokens έχουν δημιουργηθεί, διότι δεν είναι ακόμη ξεκάθαρο τι είναι νόμιμο και τι όχι σε τέτοιου είδους περιπτώσεις.

Η πρώτη ICO πραγματοποιήθηκε τον Ιούλιο του 2013 από τη Mastercoin, του J.R. Willet, ο οποίος είχε προτείνει τη δημιουργία μιας τέτοιας προσφοράς προς το κοινό, το ίδιο έτος, σε ένα συνέδριο για το Bitcoin. Η δημιουργία και πώληση των tokens, αυξήθηκε σημαντικά μερικά χρόνια μετά τη δημιουργία του Ethereum, το blockchain του οποίου φιλοξενεί μέχρι και σήμερα το 80% των tokens της αγοράς.

Οι ICOs έγιναν δημοφιλείς το 2017, όπου υπήρξαν περισσότερες από 18 ICOs μέχρι τα μέσα του έτους. Μάλιστα, το Μάιο του 2017, στην ICO της Brave, παρήχθησαν περισσότερα από 35 εκατομμύρια δολάρια, σε λιγότερο από 30 δευτερόλεπτα! Από το Νοέμβριο του ίδιου έτους υπήρχαν περίπου 50 διαφορετικά tokens που έβγαιναν προς πώληση κάθε μήνα. Στα τέλη του 2017, το 2% του συνολικού κεφαλαίου (IPOs & ICOs) βρισκόταν στις ICOs, όμως το μεγαλύτερο μέρος του κεφαλαίου αυτού αντιστοιχούσε σε 20 μόνο ICOs. Σύμφωνα με έρευνα του 2018, περίπου οι μισές ICOs του 2017 είχαν αποτύχει.

Πλέον, πολλά tokens βγαίνουν στην αγορά, αλλά και πολλά αποτυγχάνουν, με το κεφάλαιο όμως να έχει συνολικά ανοδική συμπεριφορά, ενώ κατά κύριο λόγο τα tokens αυτά συνδέονται με κάποιο DAPP.

1.5 Ιστορία των Αποκεντρωμένων Εφαρμογών (Decentralized Applications)

Οι αποκεντρωμένες εφαρμογές αποτελούν ένα από τα πολλά επαναστατικά μοντέλα που έφερε μαζί της η έλευση της τεχνολογίας του blockchain. Μέσω των εφαρμογών αυτών, πολλές από τις σχέσεις εξουσίας που διέπουν την καθημερινότητά μας τείνουν να αλλάξουν. Ειδικότερα, η ανάγκη για μεσάζοντες στις διάφορες συναλλαγές, όποιας μορφής και αν είναι αυτές, καταργείται και έτσι δίνεται η δυνατότητα στους χρήστες να εμπορευτούν ο ένας από τον άλλον άμεσα κάποιο προϊόν ή υπηρεσία. Εδώ να αναφερθεί, ότι αποκεντρωμένες εφαρμογές αποτελούν και τα ίδια τα κρυπτονομίσματα, γεγονός που θα γίνει αντιληπτό στη συνέχεια.

Αν και δεν υπάρχει μια γενική τομή στην οποία να συμφωνούν όλοι, όσον αφορά τον ορισμό των αποκεντρωμένων εφαρμογών, μια δημοσίευση του 2014, από τον David A. Johnston, ονόματι ‘The General Theory of Decentralized Applications, Dapps’ περιέχει

τον πιο αποδεκτό ορισμό για τις εφαρμογές αυτές. Ο ορισμός αυτός βασίζεται σε τρεις τομές:

1. Οι εφαρμογές αυτές πρέπει να είναι ανοιχτού κώδικα και να μην υπάρχει κάποια κεντρική εξουσία πάνω τους. Επίσης, τα δεδομένα δραστηριοτήτων τους πρέπει να παρέχονται δημόσια σε μέρος που οποιοσδήποτε αλληλεπιδρά με αυτές μπορεί να τα ελέγξει.
2. Οι εφαρμογές πρέπει να εκδίδουν κάποιο token, η διανομή του οποίου πρέπει να καθίσταται με σαφή τρόπο. Επίσης, τα tokens πρέπει να αποτελούν αναπόσπαστο και απαραίτητο κομμάτι της εφαρμογής.
3. Οι εφαρμογές αυτές μπορούν να αλλάξουν, ώστε να βελτιωθούν, αλλά αυτό πρέπει να προκύψει μέσω της συναινετικής πλειοψηφίας των χρηστών.

Οι αποκεντρωμένες εφαρμογές έκαναν την εμφάνισή τους, μαζί με το blockchain. Μάλιστα, το Bitcoin είναι και το ίδιο μια αποκεντρωμένη εφαρμογή, όπως και όλα τα κρυπτονομίσματα που ακολούθησαν. Πλέον, η αξιοπιστία και η ασφάλεια που παρέχουν οι εφαρμογές αυτές τις κάνουν ολοένα και δημοφιλέστερες μεταξύ των χρηστών. Στο γεγονός αυτό συντέλεσαν, βέβαια, η ανικανότητα πολλών ‘μεσαζόντων’ εφαρμογών να δημιουργήσουν ένα αίσθημα ασφάλειας στους χρήστες τους, καθώς και η κατάργηση των επιπρόσθετων εξόδων από τις αποκεντρωμένες εφαρμογές σε τρίτα πρόσωπα μιας συναλλαγής.

1.6 Σύνοψη της εργασίας

Η υπόλοιπη εργασία οργανώνεται ως εξής:

- Στο κεφάλαιο 2 παρουσιάζεται πλήρως η χρήση του blockchain στη λειτουργία ενός κρυπτονομίσματος. Για το σκοπό αυτό παρουσιάζονται πρώτα οι βασικοί αλγόριθμοι που χρησιμοποιούνται, έπειτα, το πώς καθορίζεται η αξία και η ποσότητα των κρυπτονομισμάτων, ενώ τέλος, ο τρόπος με τον οποίο πραγματοποιούνται οι διάφορες συναλλαγές.
- Ολόκληρο το κεφάλαιο 3 αφορά συγκεκριμένα το blockchain του Ethereum, πάνω στο οποίο είναι χτισμένη η αποκεντρωμένη εφαρμογή της συγκεκριμένης εργασίας. Παρουσιάζονται κάποια χρήσιμα βασικά χαρακτηριστικά, ενώ έπειτα, γίνεται μια εκτενής αναφορά στα έξυπνα συμβόλαια (smart contracts) της πλατφόρμας του συγκεκριμένου κρυπτονομίσματος. Στη συνέχεια, αναφέρονται τα διάφορα εργαλεία και γλώσσες προγραμματισμού, μέσω των οποίων καθίσταται δυνατή η δημιουργία του πίσω μέρους μια τέτοιας εφαρμογής (backend), ενώ παρομοίως, και η δημιουργία του μέρους, με το οποίο αλληλεπιδρά ο χρήστης (frontend). Τέλος, παρουσιάζονται η δημιουργία και οι βασικές δομές των tokens στο Ethereum.
- Το κεφάλαιο 4 αποτελεί την παρουσίαση της εφαρμογής που πραγματεύεται η συγκεκριμένη εργασία, δηλαδή του Gaming Token. Συγκεκριμένα, δίνεται η γενική ιδέα την εφαρμογής, τα διάφορα οικονομικά χαρακτηριστικά της, ενώ, εν κατακλείδι ο τρόπος με τον οποίο έχει χτιστεί η εφαρμογή, από το πίσω μέρος (backend), μέχρι και αυτό που εμφανίζεται στον χρήστη (frontend).

- Το κεφάλαιο 5, αποτελεί την κατάληξη της εργασίας, στην οποία παρουσιάζονται τα συμπεράσματά της, καθώς και μελλοντικές βελτιώσεις που είναι δυνατόν να πραγματοποιηθούν.

2 Χρήση του Blockchain στα Κρυπτονομίσματα

2.1 Εισαγωγή

Η τεχνολογία του blockchain αποδείχτηκε αρωγός αυτής των ψηφιακών νομισμάτων. Όλα τα κρυπτονομίσματα έχουν για βάση τους το blockchain και 'ζουν' μέσα σε αυτό. Υπάρχουν βασικοί λόγοι που καθιστούν το blockchain, ως το πλέον απαραίτητο και ασφαλέστερο εργαλείο δημιουργίας κρυπτονομισμάτων.

Αρχικός και βασικός λόγος χρήσης του blockchain είναι ο τρόπος με τον οποίο ανιχνεύονται και αποθηκεύονται τα διάφορα δεδομένα. Όπως αναφέρθηκε και στην πρώτη ενότητα, το blockchain είναι μια αλυσίδα από στοιχεία, τα οποία περιέχουν διάφορα δεδομένα, τα blocks. Αν ένα block δημιουργηθεί, αποτελεί μέρος του blockchain και δεν αλλάζει ποτέ. Όταν τα δεδομένα κάποιου block πρέπει να αλλάξουν, δεν αλλάζουμε το ήδη δημιουργημένο block, αλλά αυτά τα δεδομένα προστίθενται σε ένα καινούριο block της αλυσίδας αυτής, με δικιά τους χρονοσφραγίδα. Υπακούει, δηλαδή, στο γενικό λογιστικό κανόνα (general financial ledger), στον οποίο δεν καταργείς ή αλλάζεις τα ήδη υπάρχοντα δεδομένα, αλλά δημιουργείς μια νέα καταχώρηση, ώστε να γνωρίζεις όλη την προϊστορία. Παραδείγματος χάριν, αν δύο άτομα ισχυρίζονταν ότι τους ανήκει μία ιδιοκτησία, βάσει του κανόνα αυτού, θα υπήρχαν καταχωρημένοι οι ιδιοκτήτες από την αρχή ύπαρξης της ιδιοκτησίας, γεγονός που αυτομάτως θα έλυνε το πρόβλημα. Ωστόσο, το blockchain λειτουργεί λίγο διαφορετικά στο πώς αποθηκεύει τα δεδομένα αυτά.

Πριν το blockchain, τα δεδομένα αυτά ήταν συγκεντρωμένα σε ένα βιβλίο, σε έναν server ή οπουδήποτε, που, όμως, δεν είχαν όλοι πρόσβαση. Ένας, λοιπόν, ακόμη λόγος χρήσης της τεχνολογίας αυτής είναι ότι λειτουργεί αποκεντρωτικά (decentralized) και κατανεμημένα (distributed). Όταν κάποιος γίνεται κάτοχος ενός κρυπτονομίσματος, ταυτόχρονα γίνεται κάτοχος και του ίδιου του blockchain. Δηλαδή, όλοι όσοι αποτελούν μέρος αυτού του συστήματος δεν έχουν στη διάθεσή τους μόνο τις συναλλαγές και τα δεδομένα που τους αφορούν, αλλά όλα τα δεδομένα και όλες τις συναλλαγές που έχουν ποτέ δημιουργηθεί. Για να εισαχθεί ένα block στο blockchain, ακολουθείται μια διαδικασία αποκρυπτογράφησης του κρυπτογραφημένου ίχνους του block από κάποια άτομα, που ονομάζονται εξορύκτες (miners). Όποιος αποκρυπτογραφήσει πρώτος το block 'κερδίζει', ενημερώνει τους υπόλοιπους miners ότι αποκρυπτογράφησε το ίχνος και ότι το block είναι έτοιμο να εισαχθεί στο blockchain. Αν το μεγαλύτερο ποσοστό των υπόλοιπων miners εγκρίνουν το γεγονός αυτό, τότε το block προστίθεται στο blockchain, και ο miner που 'κέρδισε', λαμβάνει μια ανταμοιβή για την υπηρεσία του, η οποία είναι κάποιο ποσό του κρυπτονομίσματος, στου οποίου το blockchain έγινε η διαδικασία.

Τελευταίο βασικό λόγο χρήσης του blockchain αποτελεί η αποφυγή μεσαζόντων στις διάφορες συναλλαγές. Πριν το blockchain, δύο άτομα για να κανονίσουν μια συναλλαγή χρησιμοποιούσαν ένα άλλο άτομο ή εταιρεία ως μεσάζοντας, ο οποίος αποθήκευε τα οικονομικά τους ή εταιρικά τους στοιχεία, ώστε να υπάρξει ασφάλεια και μυστικότητα στη συναλλαγή αυτή. Ο μεσάζοντας, μετά από κάποιο χρονικό διάστημα, αποφάσιζε και κανόνιζε με τα άτομα τις διάφορες λεπτομέρειες τις συναλλαγής και έπειτα αυτή πραγματοποιούνταν, με το μεσάζοντα να παίρνει ένα σύνολο χρημάτων για την παροχή των υπηρεσιών του. Πλέον, με το blockchain, η αποφυγή του μεσάζοντα είναι γεγονός, καθώς όλοι έχουν όλα τα δεδομένα, ενώ ο καθένας μπορεί να παρουσιάσει στον άλλον όσα δεδομένα επιθυμεί, διατηρώντας ταυτόχρονα τη μυστικότητα των υπόλοιπων

δεδομένων. Οι δύο πλευρές αποφασίζουν ποια κριτήρια πρέπει να ικανοποιηθούν και μόλις αυτό συμβεί, πραγματοποιείται αυτόματα η συναλλαγή. Με τον τρόπο αυτό, γλιτώνουν πολύτιμο χρόνο και χρήματα.

Αυτή είναι μια γενική εικόνα στο πώς και γιατί τα κρυπτονομίσματα κάνουν χρήση της τεχνολογίας του blockchain. Αλλά, ας κοιτάξουμε λίγο βαθύτερα στο πώς όλα αυτά πραγματοποιούνται.

2.2 Χρήσιμοι αλγόριθμοι και δομές δεδομένων

2.2.1 Proof of Work

Η ιστορία του αλγορίθμου Proof of Work (PoW) ξεκινάει αρκετά πίσω, το 1993, όταν οι Cynthia Dwork και Moni Naor, σε μία δημοσίευσή τους, παρουσίασαν αυτόν τον αλγόριθμο, χωρίς να τον ονοματίσουν, και ανέφεραν συγκεκριμένα (μεταφρασμένο στα ελληνικά):

Η βασική ιδέα είναι να ζητηθεί από τον χρήστη να υπολογίσει μια μέτριας δυσκολίας, αλλά όχι δύσχρηστη συνάρτηση, προκειμένου να αποκτήσει πρόσβαση σε έναν πόρο, αποτρέποντας έτσι την επιπόλαιη χρήση του.

Ο αλγόριθμος αυτός πήρε την ονομασία του, 6 χρόνια αργότερα, το 1999. Το 2006, ο Harold Thomas Finney, δημιούργησε τον αλγόριθμο RPoW (Reusable Proof of Work), όπως αναφέρθηκε και στην πρώτη ενότητα, ώστε να λύσει το πρόβλημα της διπλής σπατάλης (double spending problem). Ο αλγόριθμος αυτός μετά χρησιμοποιήθηκε στο Bitcoin, για να λύσει το ίδιο πρόβλημα, ενώ συνεχίζει να χρησιμοποιείται ευρέως μέχρι σήμερα στην πλειοψηφία των κρυπτονομισμάτων.

Στην ουσία, ο Reusable Proof of Work δεν είναι αλγόριθμος, αλλά ιδέα, η οποία βασίζεται σε κάποιον αλγόριθμο υπολογισμού συνάρτησης. Proof of Work, σε γενικά πλάισια, σημαίνει ότι για να ανταμειφθείς με το να έχεις πρόσβαση σε έναν πόρο, πρέπει να υπολογίσεις την αποκρυπτογράφηση του κρυπτογραφημένου στίγματός του. Μάλιστα, για να έχεις πρόσβαση και σε άλλον πόρο πρέπει πάλι να κάνεις το ίδιο. Οπότε, έτσι, αποφεύγεται το double spending problem, διότι, αν σπαταλήσεις ένα αντικείμενο αξίας, για να έχεις πρόσβαση σε έναν πόρο, θα χρειαστεί κάποιος χρόνος ώσπου να μπορέσεις να το κάνεις, οπότε αν προσπαθήσεις με το ίδιο αντικείμενο αξίας να αποκτήσεις πρόσβαση και σε άλλον πόρο, θα χρειαστεί πάλι να υπολογίσεις, και ο χρόνος είναι αρκετά μεγαλύτερος, από αυτόν που χρειάζεται, ώστε το αντικείμενο αξίας να έχει φύγει ήδη από την κατοχή σου.

Στα κρυπτονομίσματα η ιδέα αυτή χρησιμοποιείται για την ασφάλεια των συναλλαγών και μάλιστα, υπάρχει μηχανισμός ρύθμισης, ώστε ο χρόνος απο τη δημιουργία ενός block μέχρι τη δημιουργία του επόμενου block, να διατηρείται μεγάλος. Στην ενότητα 2.4, η ιδέα αυτή, μαζί με όλες τις επακόλουθες, συνδέονται, ώστε να δώσουν το συνολικό σχήμα του πώς ακριβώς το blockchain χρησιμοποιείται στα κρυπτονομίσματα.

2.2.2 SHA-3 (Keccak-256)

2.2.2.1 *Συναρτήσεις Κατακερματισμού (Hash Functions)*

Μία συνάρτηση κατακερματισμού (hash function), είναι μια συνάρτηση, η οποία χρησιμοποιείται για να μετατρέψει (κρυπτογραφήσει) μία οποιουδήποτε μήκους συμβολοσειρά, την οποία δέχεται ως είσοδο (input), σε μία νέα, σταθερού μήκους συμβολοσειρά, την οποία προσδίδει ως έξοδο (hash value). Πολλές συναρτήσεις που κάνουν τη συγκεκριμένη δουλειά υπάρχουν στις μέρες μας, όμως κάποιοι παράγοντες πρέπει να ληφθούν υπόψιν, ώστε να αποφασίσουμε ποια από όλες θα διαλέξουμε, για να κάνουμε μια συγκεκριμένη εργασία.

Αρχικά, κάθε είσοδος, οποιαδήποτε και αν είναι αυτή, πρέπει να μετατρέπεται από τη συνάρτηση σε μοναδική έξοδο και το αντίθετο, δηλαδή από κάθε έξοδο πρέπει να είναι δυνατό να προκύψει μόνο μία είσοδος. Πρέπει, μάλιστα, να είναι αδύνατο να προκύψει ίδια έξοδος από δύο διαφορετικές εισόδους, ή το αντίθετο.

Η ταχύτητα υπολογισμού της εξόδου (hash value) αποτελεί, επιπλέον πολύ σημαντικό παράγοντα. Πρέπει να είναι σχετικά εύκολο και γρήγορο να υπολογιστεί η συγκεκριμένη τιμή, δεδομένης της εισόδου.

Τέλος, η ασφάλεια αποτελεί το σημαντικότερο παράγοντα. Η επιστροφή δεδομένης της εξόδου στην είσοδο, πρέπει να είναι υπερβολικά δύσκολη, έως και αδύνατη, ώστε να μην μπορεί κάποιος σε μικρό χρόνο να μάθει το αποκρυπτογραφημένο μήνυμα. Επίσης, μια μικρή αλλαγή στην είσοδο πρέπει να επιφέρει τεράστιες αλλαγές στην έξοδο.

2.2.2.2 *Πορεία προς τη χρήση του SHA-3*

Οι πιο δημοφιλείς hash functions κατηγοριοποιούνται ως ‘MD4 Family’, όπου τα αρχικά MD σημαίνουν ‘Message Digest’ (Σύνοψη Μηνυμάτων). Οι συναρτήσεις Message Digest δημιουργήθηκαν με σκοπό να προστατεύουν την ακεραιότητα ενός τμήματος δεδομένων (μηνύματος), ανιχνεύοντας τυχούσες αλλαγές σε οποιοδήποτε τμήμα του. Είναι ένα είδος κρυπτογραφίας που χρησιμοποιεί τιμές κατακερματισμού (hash values), με σκοπό να ενημερώσουν τον κάτοχο πνευματικών δικαιωμάτων του μηνύματος για οποιαδήποτε αλλαγή στο μήνυμα. Θα ξεκινήσουμε από τον MD4, διότι οι προκάτοχοί του δεν είχαν χαρακτηριστικά, τα οποία μας ενδιαφέρουν.

Η MD4 προτάθηκε από τον Ronald Rivest, ερευνητή του MIT, το 1990, με μήκος τιμής κατακερματισμού εξόδου (hash value) 128-bit. Αποδείχθηκε αδύναμη σε επιθέσεις συγκρούσεων (collision attacks), αλλά αποτελεί βάση μεταγενέστερων συναρτήσεων, όπως της MD5. Η MD5 προτάθηκε και πάλι από τον Ronald Rivest, το 1991, ενώ ήταν βελτιωμένη έκδοση της MD4. Είχε, όμως, τη βασική αδυναμία ότι δεν τηρούσε τον κανόνα των συναρτήσεων κατακερματισμού, όπου δε γίνεται δύο διαφορετικές είσοδοι, να δώσουν την ίδια έξοδο (message collision). Οι αδυναμίες της έγιναν γνωστές κυρίως το 1995, όμως το 2012 υπήρξε η πιο γνωστή επίθεση, ονόματι ‘Flame Malware’. Παρ’ όλα αυτά, η συνάρτηση αυτή χρησιμοποιείται ακόμη και σήμερα.

Εδώ να αναφερθεί ότι, προφανώς, όταν υπάρχουν n-bit εξόδου, η συνάρτηση μπορεί να βγάλει 2^n διαφορετικές εξόδους, γεγονός που σημαίνει ότι το collision ποτέ δεν μπορεί

πλήρως να αποφευχθεί, καθώς για ($2^n + 1$) εισόδους κάποιες δύο θα έχουν την ίδια τιμή κατακερματισμού. Το θέμα είναι να μην μπορεί κάποιος να υπολογίσει δύο τέτοιες εισόδους σε υπαρκτό χρόνο και σε καμία περίπτωση η λύση-επίθεση να έχει μικρότερη πολυπλοκότητα από $2^{n/2}$, επίθεση η οποία έχει την ονομασία 'Birthday Attack'. Επίσης, καταλαβαίνουμε ότι όσο μεγαλύτερο εύρος εξόδων έχει μία συνάρτηση κατακερματισμού, τόσο λιγότερο επιρρεπής είναι σε τέτοιου είδους επιθέσεις. Η MD5, παραδείγματος χάριν, έδινε τη δυνατότητα ακόμη και σε μικρά υπολογιστικά συστήματα να βρουν δύο τέτοιες εισόδους σε ελάχιστο χρόνο, ίσως κάποιες ώρες. Οι μεταγενέστερες συναρτήσεις καθιστούν την πραγματοποίηση της εύρεσης αυτής ολοένα και πιο χρονοβόρα, οπότε είναι ολοένα και πιο ασφαλείς.

Το 1995, από την Εθνική Υπηρεσία Ασφαλείας (NSA) των Ηνωμένων Πολιτειών, προτάθηκε ο πρώτος αλγόριθμος της κατηγορίας των SHA (Secure Hash Algorithm), ο SHA-1, με μήκος τιμής κατακερματισμού εξόδου 160-bit. Αποτέλεσε, όμως, ακόμα έναν αλγόριθμο που είχε τη βασική αδυναμία του 'εύκολου' message collision. Δέχτηκε πάμπολλες επιθέσεις ανά τα χρόνια. Το 2005 αποδείχθηκε ακαδημαϊκά ότι με λιγότερες από 2^{80} διεργασίες μπορούσε να βρεθεί ίδια τιμή εξόδου για διαφορετικές τιμές εισόδου, όμως το 2017 βρέθηκε ο γρηγορότερος τρόπος σύγκρουσης με περίπου $2^{63.1}$ διεργασίες.

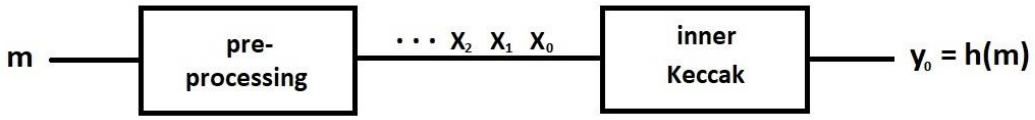
Το 2001, ένας νέος αλγόριθμος, ο SHA-2 προτάθηκε και πάλι από την Εθνική Υπηρεσία Ασφαλείας των Ηνωμένων Πολιτειών και αποτελούσε ανανεωμένη έκδοση του SHA-1. Το μήκος της τιμής κατακερματισμού της εξόδου, μπορούσε να επιλεχθεί, μεταξύ των 224-bit, 248-bit, 384-bits και 512-bit. Ο αλγόριθμος αυτός, ακόμη και σήμερα είναι ασφαλής, δηλαδή δεν έχει βρεθεί πιθανός τρόπος επίθεσης σε αυτόν. Ωστόσο, επειδή είναι άρρηκτα συνδεδεμένος με τον SHA-1, καθώς βασίζονται στην ίδια λογική, δημιούργησε ένα αίσθημα ανασφάλειας στην κοινή γνώμη η εύρεση του 2005 για τον SHA-1, και γι' αυτό φτάνουμε στην καθιέρωση του SHA-3.

Τον Νοέμβριο του 2007 η Εθνική Υπηρεσία Ασφαλείας των Ηνωμένων Πολιτειών έκανε ανοιχτή κλήση στην παγκόσμια κοινότητα κρυπτογραφίας για καταχώρηση αλγορίθμων, με σκοπό την επιλογή του καλύτερου αλγορίθμου, που θα ονομαζόταν SHA-3. Πέρασαν συνολικά 64 υποβολές, τον Οκτώμβριο του 2008, ενώ το 2010 επιλέχθηκαν για την επόμενη φάση του διαγωνισμού μόλις 5 αλγόριθμοι. Στις 2 Οκτωβρίου του 2012, τελικά, επιλέχθηκε ένας αλγόριθμος ονόματι Keccak-256, ο οποίος έγινε ο SHA-3. Ο αλγόριθμος αυτός διαθέτει ποικιλία μηκών τιμής εξόδου. Συγκεκριμένα, έχει εξόδους 224-bit, 256-bit, 384-bit και 512-bit. Δεν έχει υπάρξει κάποια απειλή ούτε για τον SHA-3 μέχρι στιγμής.

Σχεδόν όλα τα κρυπτονομίσματα καθώς και όλος ο κόσμος του Ίντερνετ χρησιμοποιούν σε τεράστιο ποσοστό τους αλγορίθμους SHA-2 και SHA-3 ως συναρτήσεις κατακερματισμού. Το Bitcoin χρησιμοποιεί τον SHA-2 (έκδοση των 256-bit), διότι δημιουργήθηκε πριν την καθιέρωση του SHA-3. Το Ethereum, καθώς και τα νεότερα κρυπτονομίσματα χρησιμοποιούν κατά κύριο λόγο τον SHA-3 (έκδοση των 256-bit). Στα πλαίσια της συγκεκριμένης διπλωματικής, θα αναλύσουμε τον SHA-3, ως αλγόριθμο που χρησιμοποιείται στο blockchain του Ethereum, πάνω στο οποίο έχει χτιστεί η συγκεκριμένη εργασία. Η ανάλυσή του επιλέχθηκε να δοθεί στη συγκεκριμένη ενότητα, ώστε ο αναγνώστης να έχει μια ιδέα για το πώς περίπου καθορίζεται ένας τέτοιος αλγόριθμος.

2.2.2.3 *Ανάλυση του SHA-3 (Keccak)*

Ο Keccak σχεδιάστηκε και προτάθηκε στον διαγωνισμό της NSA από τους Guido Bertoni, Joan Daemen, Michaël Peeters και Gilles Van Assche. Όσον αφορά τη διαδικασία, πριν μπούμε στον Keccak, δίνεται το μήνυμα που θέλουμε να κρυπτογραφηθεί (m) και σε μια πρώιμη φάση προεπεξεργασίας (pre-processing) χωρίζεται σε μικρότερα μηνύματα συγκεκριμένου μεγέθους (fixed-size blocks). Έπειτα, ο inner-Keccak είναι έτοιμος να λειτουργήσει, ώστε να δώσει την έξοδο.



Σχήμα 2.1: Εξωτερική όψη του αλγορίθμου του Keccak

Για την εκτέλεση του Keccak λαμβάνονται υπόψιν κάποιες παράμετροι που καθορίζουν την ακριβή λειτουργία του. Αυτές οι παράμετροι είναι οι εξής:

1. Κατάσταση (state) b
   Αντιπροσωπεύει το μέγεθος των μηνυμάτων που θα πραγματεύονται οι συναρτήσεις του Keccak.

2. Αριθμός γύρων (number of rounds) n
   Αντιπροσωπεύει τον αριθμό των φορών επεξεργασίας μέσω των συναρτήσεων του Keccak, ενός μηνύματος.

3. Μήκος εξόδου (output length)
   Αντιπροσωπεύει το μήκος της τιμής κατακερματισμού (hash value) που θα δωθεί ως έξοδος από τον αλγόριθμο.

4. Μέγεθος block (block size) r
   Αντιπροσωπεύει το μέγεθος των τμημάτων, στα οποία θα πρέπει να χωριστεί το μήνυμα στο pre-processing, ώστε τα τμήματα αυτά να αποτελούν σωστές εισόδους στα διάφορα τμήματα του Keccak.

5. Χωρητικότητα (capacity) c
   Αντιπροσωπεύει τον ελεύθερο χώρο που θα απομείνει στην κατάσταση b, αφότου ένος μέρος αυτού καταληφθεί από το εκάστοτε block.

Αν και η κατάσταση b και ο αριθμός των γύρων n, προτάθηκαν με πολλές διαφορετικές τιμές από τους δημιουργούς του αλγορίθμου, για SHA-3 επιλέχθηκε το κομμάτι του Keccak, για το οποίο ισχύει b = 1600 bits και n = 24 γύροι.

Από τις υπόλοιπες παραμέτρους, αρχικά επιλέγεται το μήκος εξόδου και αναλόγως με την επιλογή αυτή, καθώς και το μέγεθος της κατάστασης (b) καθορίζονται το μέγεθος

του block (r) και η χωρητικότητα (c). Στον παρακάτω πίνακα παρουσιάζονται αναλυτικά οι τιμές για τον SHA-3.

Πίνακας 2.1: Τιμές μεταβλητών του SHA-3

Μήκος εξόδου	Κατάσταση (b)	Αριθμός γύρων (n)	Μέγεθος block (r)	Χωρητικότητα (c)
224 bits	1600 bits	24	1152 bits	448 bits
256 bits	1600 bits	24	1088 bits	512 bits
384 bits	1600 bits	24	832 bits	768 bits
512 bits	1600 bits	24	576 bits	1024 bits

Παρατηρούμε ότι το άθροισμα του μεγέθους των block (r) και της χωρητικότητας (c) είναι πάντα σταθερό και ίσο με το μέγεθος της κατάστασης (b).

$$b = r + c$$

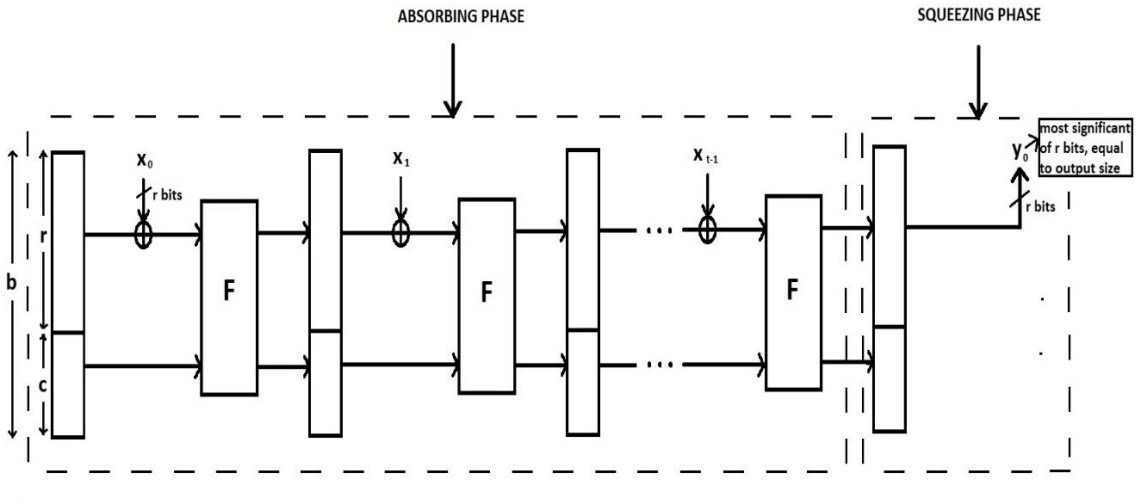
Επίσης, παρατηρούμε ότι για b = 1600 bits, η χωρητικότητα είναι πάντα διπλάσια του μήκους εξόδου.

Όσον αφορά την κατασκευή του εσωτερικού (inner) Keccak, αποτελεί ‘κατασκευή σφουγγαριού’, από την άποψη ότι στην αρχή παίρνει μήνυμα οποιουδήποτε μεγέθους, το οποίο δίνεται ως είσοδος για να πραγματοποιηθεί μια φάση του αλγορίθμου, γνωστή ως φάση απορρόφησης (absorbing phase) και στη συνέχεια ο αλγόριθμος δίνει μια έξοδο συγκεκριμένου πάντα μεγέθους, σε μια φάση γνωστή ως φάση στυψίματος (squeezing phase).

Έστω, ότι δίνεται ένα μήνυμα (m) μεγέθους k bits προς κρυπτογράφηση. Το μήνυμα αυτό διαιρείται σε t κομμάτια μεγέθους r (μέγεθος block) bits το καθένα, όπου το t προκύπτει ως εξής:

$$t = \lceil k / r \rceil$$

Αν δεν διαιρούνται ακριβώς στο τελευταίο block προστίθενται μηδενικά. Το γενικό σχήμα του inner Keccak φαίνεται παρακάτω:



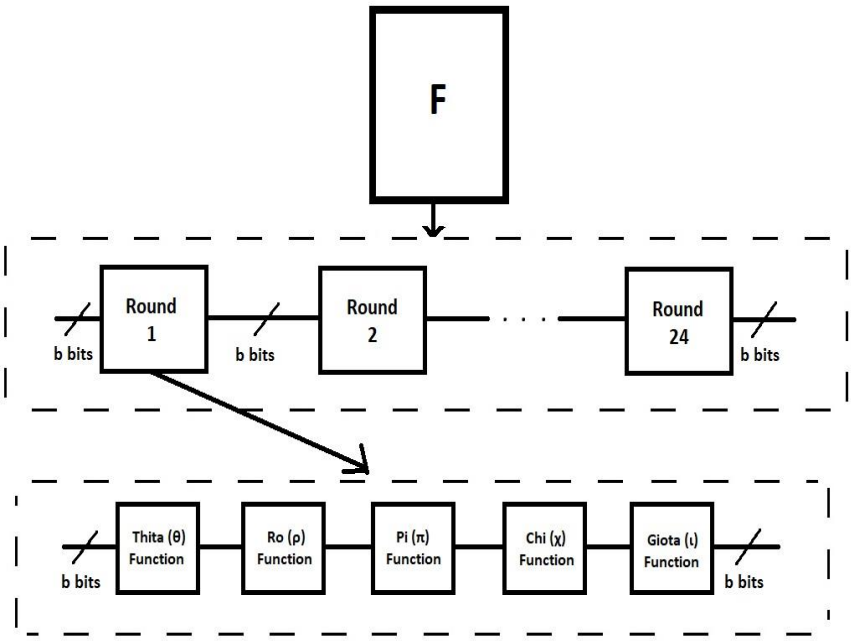
Σχήμα 2.2: Inner Keccak

Όπως φαίνεται στο σχήμα, τα δεδομένα χωρίζονται σε r και c bits στο κάθε βήμα. Αρχικά όλα τα bits έχουν την τιμή 0, ενώ έπειτα αρχίζει η φάση της απορρόφησης, όπου τα τμήματα του μηνύματος δίνονται ανά r bits. Πραγματοποιείται η πράξη xor μεταξύ των υπαρχόντων και των νέων bits. Στη συνέχεια, τα δεδομένα δίνονται στη συνάρτηση F, που αποτελεί την ουσία του αλγορίθμου και αποτελείται από 5 βήματα-συναρτήσεις, οι οποίες εκτελούνται για πολλούς γύρους. Η έξοδος της συναρτήσεως αυτής δίνεται ως είσοδος στο επόμενο βήμα και η διαδικασία επαναλαμβάνεται μέχρις ότου και το τελευταίο μέρος του μηνύματος υποστεί επεξεργασία. Ως έξοδος, δεν δίνονται και τα 1600 bits που παράγει ο αλγόριθμος, αλλά επιλέγονται τα πιο σημαντικά (most significant bits) του r ‘κομματιού’ του, έως ότου συμπληρωθεί το μήκος της τιμής κατακερματισμού εξόδου που έχει επιλεγεί. Ας δούμε, όμως, τώρα, πώς λειτουργεί η συνάρτηση F.

Εντός της συνάρτησης F, κάθε φορά που εισέρχεται ένα μέρος του μηνύματος, εκτελούνται n = 24 γύροι, καθένας εκ των οποίων αποτελείται από την εκτέλεση 5 συναρτήσεων. Σε κάθε γύρο δίνεται είσοδος b bits και εξάγεται έξοδος πάλι b bits, η οποία λειτουργεί ως είσοδος στον επόμενο γύρο, με τη διαδικασία να επαναλαμβάνεται μέχρι να ολοκληρωθούν οι γύροι. Οι δημιουργοί του Keccak, μάλιστα, έδωσαν ελληνικά ονόματα στις συναρτήσεις που εκτελούνται σε κάθε γύρο. Οι συναρτήσεις είναι με σειρά εκτέλεσης οι:

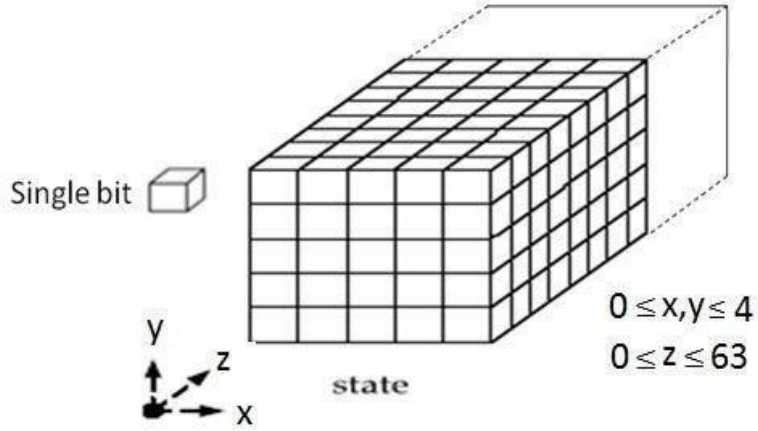
- Theta (θ) Function
- Rho (ρ) Function
- Pi (π) Function
- Chi (χ) Function
- Iota (ι) Function

Σχηματικά, το εσωτερικό της συνάρτησης F φαίνεται παρακάτω:



Σχήμα 2.3: Τα βήματα της συνάρτησης του Keccak

Για να εξηγήσουμε τα επιμέρους βήματα, ας δούμε πρώτα πώς αναπαρίσταται η κατάσταση (state). Τα 1600 bits, από τα οποία αποτελείται, μπορούν να αναπαρασταθούν ως ένας τρισδιάστατος πίνακας 5 x 5 x 64. Η μορφή αυτή φαίνεται στο παρακάτω σχήμα:



Σχήμα 2.4: Αναπαράσταση του state του Keccak ,

Mahendra Vucha, State Matrix (A) of SHA-3, www.researchgate.net/figure/State-Matrix-A-of-SHA-3-The-block-diagram-of-SHA-3-consists-of-four-functional-blocks_fig1_301335735, accessed 1st July 2019.

Theta (θ) step

Στο βήμα αυτό, καθένα εκ των 1600 bits αντικαθίσταται από το XOR άθροισμα 11 bits. Το πρώτο είναι το ίδιο το bit, τα επόμενα 5 είναι αυτά που ανήκουν στη στήλη που βρίσκεται μία θέση 'αριστερά' του (ως προς x), ενώ τα τελευταία 5 είναι αυτά που ανήκουν στη στήλη που βρίσκεται μία θέση 'δεξιά' του (ως προς x) και μία θέση 'πίσω' του (ως προς z). Δηλαδή, για να βρούμε το ανανεωμένο bit A(x, y), έχουμε:
$C(x) = A(x,0) \oplus A(x,1) \oplus A(x,2) \oplus A(x,3) \oplus A(x,4)$

$D(x) = C((x-1) \bmod 5) \oplus \mathrm{rot}(C((x+1) \bmod 5), 1)$

$A(x, y) = A(x, y) \oplus D(x)$

Rho (ρ) και Pi(π) steps

Στα βήματα αυτά, το state αντιμετωπίζεται ως ένας δισδιάστατος πίνακας 5 x 5, ο οποίος περιέχει λέξεις 64 bit. Ας θεωρήσουμε είσοδο A(x,y), όπου x, y = 0,1,2,3,4 και έξοδο B(x,y), όπου x, y = 0,1,2,3,4.

Στο Rho (ρ) βήμα κάθε λέξη A(x,y) περιστρέφεται (rotate) κατά έναν συγκεκριμένο αριθμό, ο οποίος καθορίζεται στον κάτωθι πίνακα, έστω t:

Πίνακας 2.2: Αριθμοί για το βήμα Rho (ρ) του αλγορίθμου του Keccak

	x = 3	x = 4	x = 0	x = 1	x = 2
y = 2	25	39	3	10	43
y = 1	55	20	36	44	6
y = 0	28	27	0	1	62
y = 4	56	14	18	2	61
y = 5	21	8	41	45	15

Από την περιστροφή προκύπτει:

TEMP(x, y) = rot(A(x, y), t(x, y))

Στο Pi(π) βήμα γίνεται ανταλλαγή των περιστρεφόμενων λέξεων βάσει του τύπου:

B(y, (2x + 3y) mod 5) = TEMP(x, y)

Chi (χ) step

Στο βήμα αυτό και πάλι το state αντιμετωπίζεται ως ένας δισδιάστατος πίνακας 64-bit λέξεων. Έστω είσοδος B(x,y), όπου x, y = 0,1,2,3,4 και έξοδος C(x,y), όπου x, y = 0,1,2,3,4. Πραγματοποιείται η πράξη:

$C(x,y) = B(x,y) \oplus [\, B((x+1) \bmod 5, y) \vee B((x+2) \bmod 5, y)]$

Iota (ι) step

Στο τελευταίο αυτό βήμα, στη λέξη C(0,0) προστίθεται ένας σταθερός όρος, αναλόγως με τον αριθμό του γύρου, έστω i στον οποίο βρισκόμαστε, με πρώτο γύρο για i = 0, βάσει του παρακάτω πίνακα, έστω RC:

Πίνακας 2.3: Δεκαεξαδικές τιμές των αριθμών του βήματος Iota (ι) του αλγορίθμου του Keccak

Γύρος (i)	Αριθμός σε δεκαεξαδική μορφή (RC(i))	Γύρος (i)	Αριθμός σε δεκαεξαδική μορφή (RC(i))
0	0x0000000000000001	12	0x000000008000808B
1	0x0000000000008082	13	0x800000000000008B
2	0x800000000000808A	14	0x8000000000008089
3	0x8000000080008000	15	0x8000000000008003
4	0x000000000000808B	16	0x8000000000008002
5	0x0000000080000001	17	0x8000000000000080
6	0x8000000080008081	18	0x000000000000800A
7	0x8000000000008009	19	0x800000008000000A
8	0x000000000000008A	20	0x8000000080008081
9	0x0000000000000088	21	0x8000000000008080
10	0x0000000080008009	22	0x0000000080000001
11	0x000000008000000A	23	0x8000000080008008

Οπότε τελικά έχουμε:

C(0,0) = C(0,0) + RC(i) , όπου i = 0,1,2,…,23

Έπειτα, συνεχίζουμε μέχρι να τελειώσουν όλοι οι γύροι για όλα τα blocks μέχρι το τέλος του μηνύματος. Αυτός ήταν ο αλγόριθμος του Keccak, ο πιο βασικός αλγόριθμος διαδικτυακής κρυπτογράφησης, καθώς χρησιμοποιείται σχεδόν σε όλο το διαδίκτυο.

Τέλος, να σημειωθεί ότι για την κρυπτογράφηση στο blockchain του Ethereum χρησιμοποιείται η εκδοχή με μέγεθος τιμής κατακερματισμού εξόδου (hash value) 256-bits, οπότε μέγεθος block ή blocksize (r) 1088-bits, ενώ η χωρητικότητα ή capacity (c) είναι μεγέθους 512-bits.

2.2.3 Merkle Tree

Το Merkle Tree είναι μία δομή δεδομένων (data structure), η οποία αναπαρίσταται ως ένα δέντρο. Κάθε φύλλο (leaf) του δέντρου αναπαριστά ένα κρυπτογραφημένο μήνυμα, κάθε κλαδί (branch) του αναπαριστά τη συνδυασμένη τιμή κατακερμαρισμού εξόδου (hash value) των παιδιών της, βάσει ενός αλγορίθμου κατακερματισμού (hash algorithm), ενώ η ρίζα του αποτελείται από μία μόνο τιμή, η οποία αναπαριστά τη συνολική συνδυαστική τιμή κατακερματισμού όλου του δέντρου.

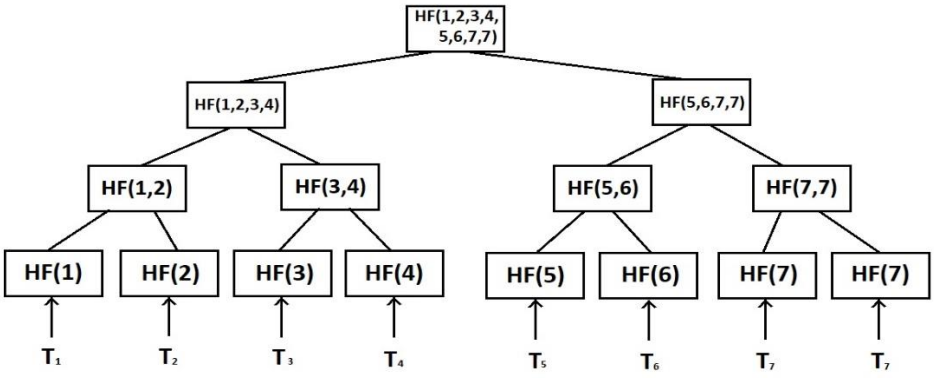
Η ιστορία του ξεκινάει το 1979, όταν ο Ralph Merkle (εξού και η ονομασία), ενώ φοιτούσε στο Stanford University, έβγαλε μια ακαδημαϊκή δημοσίευση με τίτλο ‘A

Certified Digital Signature'. Στη δημοσίευση αυτή ανέφερε μία εξαιρετικά αποδοτική μέθοδο για τη δημιουργία αποδείξεων, την οποία δεν ονομάτισε . Μέσω της μεθόδου αυτής, οι υπολογιστές θα μπορούσαν να επαληθεύσουν δεδομένα πολύ γρηγορότερα απ' ό,τι στην εποχή εκείνη. Το 1992, χρησιμοποιήθηκαν για να εισάγονται περισσότερες της μίας συναλλαγές στο πρόβλημα χρονοσφραγίδων, που παρουσιάζεται στην πρώτη ενότητα, ενώ μετά υπήρξε μια περίοδος παύσης της χρήσης τους. Το 2008, όμως, με την έλευση του Bitcoin, το οποίο χρησιμοποιούσε στη δημιουργία των block του μόνο Merkle Trees, καθιερώθηκαν σε όλη την κοινότητα κρυπτογραφίας και χρησιμοποιούνται ευρέως μέχρι σήμερα, σε τέτοιου και άλλου είδους εφαρμογές κρυπτογράφησης. Ας εμβαθύνουμε, όμως, στην ακριβή χρήση των δέντρων αυτών στα κρυπτονομίσματα, ώστε να αντιληφθούμε την πλήρη εικόνα τους.

Το blockchain ενός κρυπτονομίσματος αποτελείται από χιλιάδες block, ενώ κάθε block αποτελείται από μερικές χιλιάδες συναλλαγές. Κάθε block χρησιμοποιεί ως μέσο αναπαράστασης όλων των συναλλαγών που βρίσκονται σε αυτό, ένα μόνο Merkle Tree και μάλιστα μόνο τη ρίζα του (Merkle Root).

Τα Merkle Trees χρησιμοποιούν πάντα ζευγάρια μηνυμάτων-συναλλαγών. Αν ο αριθμός τους είναι περιττός (odd), τότε δημιουργείται ένα αντίγραφο της τελευταίας συναλλαγής και γίνεται ζευγάρι με τον εαυτό της, καθώς και όταν κατά πλάτος ενός βήματος, το σύνολο των κλαδιών που προκύπτουν είναι περιττός, τότε και πάλι δημιουργείται αντίγραφο του τελευταίου κλαδιού και ζευγαρώνει με τον εαυτό του. Σε κάθε γύρο το σύνολο των νέων κλαδιών ή φύλλων περνάει σε ζευγάρια από μια συνάρτηση κατακερματισμού (hash function), από την οποία προκύπτει το επόμενο επίπεδο, το οποίο περιέχει τη συνδυασμένη τιμή κατακερματισμού των δύο και η διαδικασία συνεχίζεται, έως ότου προκύψει η ρίζα του δέντρου. Για να αντιληφθούμε καλύτερα τη λειτουργία των Merkle Trees, ας αναφέρουμε ένα παράδειγμα.

Έστω ότι έχουμε 7 συναλλαγές που πρέπει να μπούνε σε ένα block (T_1, T_2, …, T_7) και ότι απλά χρησιμοποιούμε μια hash function (HF). Το δέντρο που θα προκύψει είναι το εξής:



Σχήμα 2.5: Παράδειγμα Merkle Tree

Οπότε, όλα τα δεδομένα όλων των συναλλαγών είναι αποθηκευμένα μέσα στην τιμή κατακερματισμού της ρίζας του δέντρου. Όμως, το Merkle Tree δεν είναι δέντρο αναζήτησης, δηλαδή για να αποδείξει κάποιος ότι μία συναλλαγή περιέχεται σε αυτό πρέπει να ψάξει όλο το δέντρο. Επίσης, για να αποδειχτεί ότι κάθε συναλλαγή δεν είναι ψεύτικη, πάλι πρέπει να ψαχτεί όλο το δέντρο. Ωστόσο, αν κάποιος θέλει να αλλάξει σκοπίμως μια συναλλαγή, ώστε να προκαλέσει πρόβλημα στο blockchain, αυτό γίνεται αμέσως αντιληπτό, καθώς η ρίζα του δέντρου αλλάζει τελείως. Οπότε, τα Merkle Trees προσφέρουν ασφάλεια, αλλά έχουν πολύ μεγάλο χρόνο αναζήτησης. Για το λόγο αυτό χρησιμοποιείται ένα τέχνασμα, ώστε κάποιος που απλά θέλει να ελέγξει μόνο αν μια συναλλαγή που τον αφορά περιέχεται στο συγκεκριμένο block και όχι όλες τις συναλλαγές, να μπορεί να το κάνει με μικρό κόστος σε χρόνο και χώρο. Συγκεκριμένα, δίνονται μόνο οι απαραίτητες πληροφορίες που θα χρειαστεί σε κάθε επίπεδο του δέντρου, ώστε βάζοντας τη συναλλαγή να μπορεί να σχηματίσει τη ρίζα του δέντρου και να δει αν αυτή ταυτίζεται με την πραγματική. Αν ναι, η συναλλαγή περιέχεται όντως στο block.

Παραδείγματος χάρη, αν κάποιος θέλει να ελέγξει αν περιέχεται στη συναλλαγή T_6, μια συναλλαγή που τον αφορά, ή αν η συναλλαγή T_6 είναι έγκυρη, αρκεί να του δοθούν οι κόμβοι HF(5) , HF(7,7) , HF(1,2,3,4), του παραπάνω σχήματος. Έπειτα, μέσω του HF(6) που ήδη έχει βρίσκει τον HF(5,6), μέσω του HF(5,6) και του HF(7,7) βρίσκει τον HF(5,6,7,7) και, τέλος, μέσω του HF(1,2,3,4) και του HF(5,6,7,7) βρίσκει τη ρίζα και τη συγκρίνει με την πραγματική.

Καταλήγοντας, θα ήθελα να αναφέρω ότι το blockchain του Ethereum δεν περιέχει απλώς ένα Merkle Tree, αλλά ένα σύνολο από τέτοια δέντρα, ονόματι Merkle Patricia Trie για να εξυπηρετήσει τις ανάγκες του, με τρόπο που θα μελετηθεί στο κεφάλαιο 3.

2.3 Εξόρυξη (Mining)

Το mining είναι το σημαντικότερο κομμάτι των κρυπτονομισμάτων και είναι αυτό που στην ουσία προσδίδει αξία στα κρυπτονομίσματα. Κάθε χρήστης κρυπτονομίσματος, όπως αναφέραμε, έχει ένα αντίγραφο του blockchain αυτού. Σε πολλά κρυπτονομίσματα, βέβαια, ο χρήστης έχει την επιλογή να γίνει ‘lightweight client’, δηλαδή χρήστης, ο οποίος δεν έχει όλο το blockchain, αλλά ένα μέρος του, το οποίο και αφορά μόνο τις συναλλαγές του. Αυτή η επιλογή δίνεται επειδή ο όγκος των δεδομένων αυξάνεται συνεχώς και μπορεί να μην υπάρχει ο απαραίτητος αποθηκευτικός χώρος στο hardware του χρήστη. Εκτός αυτών των χρηστών, όλοι οι υπόλοιποι χρήστες, μπορούν να συμβάλλουν στο mining του εκάστοτε κρυπτονομίσματος και να γίνουν miners.

Miners είναι χρήστες, οι οποίοι δημιουργούν τα νέα blocks του blockchain, με σκοπό την πληρωμή τους. Για να πληρωθεί κάποιος miner πρέπει να είναι ο πρώτος από όλους τους υπόλοιπους που θα βρει μια τιμή κατακερματισμού εξόδου (hash value) για ένα block, που θα πληρεί ορισμένα κριτήρια. Όταν κάποιος miner ισχυριστεί ότι βρήκε μια τέτοια τιμή, στέλνει την πληροφορία σε όλους του υπόλοιπους miners. Αν η πλειοψηφία αυτών δεχτεί το block αυτό ως έγκυρο, τότε η προσθήκη του στην αλυσίδα είναι γεγονός και όλοι οι χρήστες το προσθέτουν στην τοπική τους αλυσίδα, ενώ ο miner που πρότεινε το block πληρώνεται ένα πόσο του εκάστοτε κρυπτονομίσματος. Για το σκοπό αυτό οι miners καταναλώνουν υπολογιστική ισχύ των μηχανημάτων τους, η οποία μεταφράζεται σε ηλεκτρικό ρεύμα.

Από αυτά τα ποσά που πληρώνονται οι miners για τη δημιουργία των blocks, προκύπτει και το σύνολο των κρυπτονομισμάτων που υπάρχει στην αγορά. Η συνολική παροχή ενός κρυπτονομίσματος προκύπτει από την υπολογιστική ισχύ που έχει σπαταληθεί στο mining του. Δηλαδή, όταν ένα κρυπτονόμισμα βγαίνει στην αγορά, αρχικά η συνολική παροχή είναι μηδενική, δηλαδή δεν υπάρχει ακόμα κανένα κρυπτονόμισμα. Με τη δημιουργία του πρώτου block, ο miner-δημιουργός του block παίρνει την αμοιβή που δίνεται και σιγά σιγά όσο μεγαλώνει η αλυσίδα τόσα περισσότερα είναι τα κρυπτονομίσματα που παράγονται. Εδώ να διευκρινιστεί ότι το πόσα κρυπτονομίσματα θα δίνονται σε κάθε νέο block είναι προγραμματισμένο από τον σχεδιαστή του εκάστοτε κρυπτονομίσματος, ενώ ο αριθμός αυτός συνήθως μειώνεται όσο αυξάνεται ο αριθμός των block. Αυτό συμβαίνει, διότι όσο περισσότερα blocks περιέχει ένα blockchain, τόσο πιο ασφαλές από επιθέσεις είναι, καθώς έχει περισσότερους miners και είναι δυσκολότερο να φτάσουν σε ομοφωνία για κακόβουλο block, οπότε τόσο λιγότερη αξία έχει και η δημιουργία νέων blocks. Τέλος, αξίζει να αναφερθεί ότι πολλά κρυπτονομίσματα βγαίνουν στην αγορά, με ήδη υπάρχουσα παροχή (pre-mined), μέρος της οποίας δίνεται προς αγορά στο κοινό, μέσω ICO (Initial Coin Offering). Έτσι συνέβη και στην περίπτωση του Ethereum.

Ας δούμε, όμως, πώς ακριβώς πραγματοποιείται το mining. Όταν δημιουργείται μια συναλλαγή (transaction) πηγαίνει αυτόματα σε μία ‘πισίνα’ ανεξακρίβωτων συναλλαγών (pool of unconfirmed transactions). Κάθε miner ξεχωριστά διαλέγει όσες και όποιες συναλλαγές θέλει από αυτή την ‘πισίνα’, που δεν ξεπερνούν όμως σε χώρο ένα συγκεκριμένο μέγεθος, και δημιουργεί ένα block. Το block αυτό περιέχει συνήθως το σύνολο των συναλλαγών που επιλέχθηκαν ως ένα Merkle Tree, το hash value του προηγούμενου block, καθώς και έναν τυχαίο αριθμό (nonce). Πριν συνεχιστεί η διαδικασία, ο miner ελέγχει αν οι συναλλαγές που έχει εντάξει στο block του δεν παραβαίνουν κάποιον όρο (π.χ. δεν σπαταλώνται τα ίδια χρήματα δύο φορές), ώστε να έχει όσο το δυνατόν μεγαλύτερες πιθανότητες να δεχτούν το block ως έγκυρο οι υπόλοιποι. Όταν ο miner επιλέξει τις συναλλαγές, δύο πράγματα είναι σταθερά, το hash value του προηγούμενου block, καθώς και οι συναλλαγές που επέλεξε. Το τρίτο, δηλαδή το nonce, μεταβάλλεται, ώσπου η σύμπτηξη των τριών αυτών δεδομένων περάσει από τη συνάρτηση κατακερματισμού που χρησιμοποιείται στο blockchain του εκάστοτε κρυπτονομίσματος και δώσει αποτέλεσμα, που πληρεί ορισμένες προϋποθέσεις. Άλλωστε, όπως αναφέραμε, μια μικρή αλλαγή στην είσοδο επιφέρει τεράστιες αλλαγές στην έξοδο μιας συνάρτησης κατακερματισμού (π.χ. Keccak).

Οι προϋποθέσεις που πρέπει να συναντώνται, ώστε κάποιο block να είναι έγκυρο και να το αποδεχτούν οι υπόλοιποι miners, προκύπτουν από μία τιμή που ονομάζεται δυσκολία (difficulty). Όσο μεγαλύτερη είναι αυτή η τιμή, τόσο πιο δύσκολο είναι να βρεθεί μία έγκυρη έξοδος και τόσο περισσότερες τιμές για το nonce πρέπει να δοκιμαστούν. Μάλιστα, για αποφυγή παραβάσεων, τα περισσότερα blockchain επιδιώκουν να υπάρχει ένας σταθερός χρόνος μεταξύ της δημιουργίας δύο block (π.χ. Bitcoin, 10 λεπτά). Για να το επιτύχουν αυτό, δεν απαγορεύουν το mining για τόσο χρόνο, αλλά αυτόματα τα blockchain τους μεταβάλλουν τη δυσκολία, μικραίνοντας την τιμή της αν τα block αργούν να δημιουργηθούν και αντίστοιχα, μεγαλώνοντάς την, αν τα block δημιουργούνται ταχύτερα. Αναλυτικά, για τη δυσκολία του Ethereum, θα μιλήσουμε στο τρίτο κεφάλαιο. Για να υπάρχει μια εικόνα στο τι μπορεί να σημαίνει δυσκολία, να αναφερθεί ότι στο blockchain του Bitcoin, η δυσκολία (difficulty) καθορίζει από πόσα μηδενικά πρέπει να ξεκινά η τιμή κατακερματισμού της εξόδου. Δηλαδή, αν το difficulty

είναι 8, θα πρέπει να μεταβάλλεται το nonce από τον miner, μέχρι να βρεθεί μία έξοδος της μορφής 00000000xxxx… .

Όταν ένας miner βρει ένα block, με έγκυρη υπογραφή (τιμή κατακερματισμού), στέλνει σε όλους τους υπόλοιπους το block και την υπογραφή. Αν η πλειοψηφία των miners, αφού ελέγξουν το block, αποφασίσουν ότι είναι έγκυρο, τότε φτάνουν σε ομοφωνία και εντάσσουν όλοι το block αυτό, στην τοπική τους αλυσίδα. Εδώ είναι που υπάρχει η έννοια του proof-of-work που μελετήσαμε. Ο miner που βρήκε το έγκυρο block απέδειξε στην ουσία στους υπόλοιπους την εργασία του, η οποία μεταφράζεται σε υπολογιστική ισχύ που σπατάλησε, ώσπου να σχηματίσει ένα block, χωρίς παραβάσεις στις συναλλαγές του και να βρει ένα nonce που να τηρεί τα κριτήρια που εισάγει το difficulty. Αφού ολοκληρωθεί η διαδικασία ένταξης του νέου αυτού block στο blockchain, ο miner που το πρότεινε πληρώνεται για τη συνεισφορά του το σύνολο των κρυπτονομισμάτων που αντιστοιχεί στη δημιουργία αυτή, καθώς και καθορίζει πού θα πάνε κάποιοι μικροί φόροι που δίνονται από αυτούς που πυροδότησαν τις συναλλαγές που περιέχει το block, τους οποίους θα μελετήσουμε στην αμέσως επόμενη υποενότητα. Πρώτα, όμως, για να σχηματιστεί ολική εικόνα του mining, θα δούμε πώς ένας miner κερδίζει στον ‘αγώνα’ αυτό.

Αν και η διαδικασία εύρεσης ενός miner-νικητή φαντάζει τελείως τυχαία, στην πραγματικότητα δεν είναι. Μάλιστα, βασίζεται σε πιθανότητες, οι οποίες προκύπτουν από την υπολογιστική δύναμη του κάθε miner. Δηλαδή, αν το μηχάνημα mining ενός miner υπολογίζει τιμές κατακερματισμού εξόδου 10 φορές πιο γρήγορα από ενός άλλου, τότε θα έχει και 10 φορές μεγαλύτερη πιθανότητα να στεφθεί νικητής, διότι, όταν ο δεύτερος θα έχει δοκιμάσει μία τιμή nonce, ο πρώτος θα έχει ήδη δοκιμάσει δέκα τέτοιες τιμές. Η διαδικασία αυτή οδηγεί, με την πάροδο του χρόνου, σε όλο και μεγαλύτερη σπατάλη ηλεκτρικής ενέργειας.

Όταν ένα block προστίθεται στο blockchain, όλοι οι miners και όχι μόνο ο miner-νικητής, ξεκινούν από την αρχή αγώνα για την εύρεση του επόμενου block. Άρα, η υπολογιστική ισχύς που σπαταλήθηκε από όλους τους υπόλοιπους miners, εκτός του νικητή, πάει χαμένη. Δεν μπορούν να συνεχίσουν να εξετάζουν αριθμούς nonce για το block που είχαν φτιάξει και έψαχναν αποτύπωμα για δύο λόγους:

i. Μπορεί κάποιες από τις συναλλαγές που είχαν στο block να γίναν ήδη αποδεκτές από το block που δημιουργήθηκε, οπότε το δεύτερο από τα τρία δεδομένα που δίνονται στη συνάρτηση κατακερματισμού, δηλαδή αυτό των συναλλαγών, άλλαξε (τελείως διαφορετικό Merkle Tree).
ii. Έχει αλλάξει το πρώτο από τα τρία δεδομένα, δηλαδή το hash value του προηγούμενου block, καθώς πλέον προηγούμενο είναι αυτό, που μόλις προστέθηκε.

Συμπεραίνουμε ότι η διαδικασία πρέπει απαραίτητα να ξεκινήσει από την αρχή για όλους, οπότε όντως τεράστιος όγκος υπολογιστικής ισχύος σπαταλάται άσκοπα, ενώ όσο περισσότεροι γίνονται οι miners, τόσο μεγαλύτερη η σπατάλη. Επιπλέον, αυτό οδηγεί στη συσσώρευση πολλών miner στα λεγόμενα mining pools. Τα mining pools είναι δίκτυα κάτω από τα οποία miners ‘ενώνουν’ την ισχύ τους, ώστε να μεγιστοποιήσουν την πιθανότητα για κέρδος, αν και αυτό όταν έρχεται, είναι αισθητά μικρότερο, καθώς, όποιος από αυτούς και αν κερδίσει, τα κέρδη μοιράζονται ισότιμα, αναλόγως με την υπολογιστική ισχύ που ο καθένας προσφέρει στο δίκτυο. Το γεγονός αυτό αυξάνει την

ανασφάλεια, καθώς αν ένα τέτοιο δίκτυο υπερβεί την πλειοψηφία των miners ενός κρυπτονομίσματος, θα μπορεί να καθορίζει ποια blocks θα προστίθενται και να φτάνει μόνο του σε ομοφωνία, ακόμα και κακόβουλων συναλλαγών, κάνοντας στην ουσία επίθεση στο κρυπτονόμισμα, τη λεγόμενη '51% attack'. Τα τρία μεγαλύτερα τέτοια δίκτυα, παραδείγματος χάριν, για το Bitcoin, αν ενωθούν, θα μπορούσαν να εξαπολύσουν μια τέτοια επίθεση. Εν κατακλείδι, οι δημιουργοί των κρυπτονομισμάτων, οδηγούνται στην αναζήτηση ενός άλλου τρόπου ομοφωνίας, διαφορετικού από το proof-of-work, ώστε να αποφευθεί η σπατάλη τόσο μεγάλου όγκου υπολογιστικής ισχύος, καθώς και ο κίνδυνος για μια τέτοιου είδους επίθεση. Ήδη, πολλά κρυπτονομίσματα έχουν αποφασίσει και δουλεύουν πάνω στην αλλαγή σε proof-of-stake, μεταξύ των οποίων και το Ethereum. Το proof-of-stake θα μελετηθεί στο τελευταίο κεφάλαιο.

2.4 Συναλλαγές (Transactions)

Έχει αναφερθεί σε πολλά σημεία του κειμένου, η αξία της χρήσης του blockchain, λόγω του ότι δε χρειάζονται μεσάζοντες στις συναλλαγές. Μια συναλλαγή μεταξύ δύο χρηστών μπορεί να πραγματοποιηθεί χωρίς να σπαταληθεί χρόνος και να δοθούν χρήματα σε κάποιον μεσάζοντα, ακόμα και αν είναι ανώνυμοι μεταξύ τους, καθώς, γνωρίζουν ότι αν υπάρξει κάποια παράβαση στη συναλλαγή αυτή, οι miners δε θα φτάσουν ποτέ σε ομοφωνία (αν και υπάρχει τρόπος μέσω της λεγόμενης '51% attack'). Η πραγματικότητα, όμως, είναι ότι οι χρήστες σπαταλούν κάποιο μικρό ποσό για τη συναλλαγή τους αυτή, γνωστό ως transaction fee.

Το transaction fee μπορεί να είναι ακόμα και μηδενικό και καθορίζεται από διάφορους παράγοντες. Πρώτο και κύριο παράγοντα αποτελεί ο όγκος των δεδομένων που πρέπει να εκτελεστούν, για να πραγματοποιηθεί μια συναλλαγή. Όσο μεγαλύτερος ο όγκος των δεδομένων, τόσο μεγαλύτερος και ο φόρος. Επιπλέον, ρόλο παίζει και πόσο γρήγορα θέλει ο χρήστης να πραγματοποιηθεί η συναλλαγή. Οι miners, ως επί το πλείστον, εισάγουν στα block τους συναλλαγές, οι οποίες έχουν μεγάλο φόρο, ώστε να μεγιστοποιήσουν το κέρδος τους. Οπότε, όσο μεγαλύτερος ο φόρος, τόσο μεγαλύτερες και οι πιθανότητες η συναλλαγή να εισαχθεί στα block των miners, άρα και τόσο γρηγορότερα γίνεται μέρος της αλυσίδας, δηλαδή πραγματοποιείται. Πολλοί άλλοι παράγοντες μπορεί να επηρεάσουν τους φόρους συναλλαγής, αλλά διαφέρουν από κρυπτονόμισμα σε κρυπτονόμισμα, αναλόγως με τη χρήση του.

Η πορεία μιας συναλλαγής προς την πραγματοποίηση εξηγήθηκε πλήρως στο mining. Οπότε, σε αυτό το σημείο, έχουμε μια σφαιρική εικόνα για τα βασικά χαρακτηριστικά λειτουργείας του blockchain ενός οποιουδήποτε κρυπτονομίσματος. Το κρυπτονόμισμα που μας ενδιαφέρει, όμως, στα πλαίσια της διπλωματικής είναι το Ethereum, το οποίο θα μελετήσουμε και εκτενώς στην επόμενη ενότητα, έχοντας όμως όλες τις βάσεις της ενότητας αυτής.

3 Ethereum

3.1 Εισαγωγή

Το Ethereum είναι το δεύτερο κρυπτονόμισμα που εμφανίστηκε στις αγορές μετά το Bitcoin. Αν και κυρίως η ιδέα προήρθε από τον Vitalik Buterin, ανακοινώθηκε επίσημα με μια μεγάλη λίστα δημιουργών. Τον Δεκέμβριο του 2013, ο Antonio Di Iorio έγραψε 'το Ethereum ιδρύθηκε από τους Vitalik Buterin, τον εαυτό μου, τον Charles Hoskinson, τον Mihai Alisie και τον Amir Chetrit'. Οι Joseph Joseph Lublin, Gavin Wood και Jeffrey Wilke προστέθηκαν στις αρχές του 2014 ως ιδρυτές. Η τυπική ανάπτυξή του ξεκίνησε στις αρχές του 2014, από μια ελβετική εταιρεία, την Ethereum Switzerland GmbH (EthSuisse).

Η βασική ιδέα ήταν το Ethereum, να μην αποτελεί απλώς κρυπτονόμισμα, αλλά στην ουσία να αποτελεί πλατφόρμα, μέσω της οποίας να μπορούν να δημιουργούνται διάφορες εφαρμογές με βάση το blockchain (αποκεντρωμένες εφαρμογές). Μάλιστα, αρχικά, όταν και ο Vitalik Buterin είχε προτείνει την ιδέα αυτή, προοριζόταν να πραγματοποιηθεί στο blockchain του Bitcoin. Εκτός, όμως, από την ασυνεννοησία στην κοινότητα, ένας άλλος βασικός λόγος που οδήγησε στη δημιουργία ενός νέου κρυπτονομίσματος ήταν ότι η γλώσσα του Bitcoin είναι Turing-incomplete. Η υποστήριξη της ιδέας απαιτούσε μια Turing-complete γλώσσα. Έτσι, τέθηκε η ιδέα να χρησιμοποιούνται έξυπνα συμβόλαια και έπειτα ξεκίνησε η δημιουργία του λογισμικού μέσω του οποίου θα εκτελούνταν. Το έργο αυτό ανέλαβε ο Gavin Wood, επικεφαλής της εταιρείας, ο οποίος παρουσίασε το 'yellow paper', που καθόριζε την εικονική μηχανή (virtual machine) του Ethereum (EVM).

Έπειτα, δημιουργήθηκε ένα ελβετικό μη κερδοσκοπικό ίδρυμα για το Ethereum, το Stiftung Ethereum. Η ανάπτυξη του ιδρύματος αυτού χρηματοδοτήθηκε από το κοινό στην πρώτη ICO (Initial Coin Offering) που πραγματοποιήθηκε την περίοδο Ιουλίου-Αυγούστου του 2014, όπου οι επενδυτές μπορούσαν να αγοράσουν ether, δηλαδή το νόμισμα του Ethereum, χρησιμοποιώντας, όμως, Bitcoins για την αγορά του. Συνολικά, δόθηκαν 72 εκατομμύρια ether, τα οποία είχαν γίνει pre-mined, δηλαδή δε γεννήθηκαν από το mining κάποιου block. Τον Μάρτιο του 2017, ανακοινώθηκε η δημιουργία του Enterprise Ethereum Alliance (EOX), που περιείχε 30 ιδρυτικά μέλη και αποτελούσε μη κερδοσκοπικό οργανισμό εταιρειών που χρησιμοποιούσαν το blockchain, ενώ σήμερα περιέχει πάνω από 150 εταιρείες-μέλη.

Με λίγα λόγια, η δημιουργία του Ethereum, αποτέλεσε βάση για τη δημιουργία χιλιάδων αποκεντρωμένων εφαρμογών, καθώς χρησιμοποιεί τα έξυπνα συμβόλαια (smart contracts), τα οποία γράφονται σε γλώσσα Turing-Complete. Πάμε, όμως, να μελετήσουμε αρχικά, πώς το Ethereum διαφέρει από το βασικό πρότυπο ενός κρυπτονομίσματος που παρουσιάστηκε στο κεφάλαιο 2.

3.2 Βασικά Χαρακτηριστικά

Ο τρόπος, με τον οποίο το blockchain του Ethereum έχει κατασκευαστεί, διαφέρει πολύ από το κλασσικό πρότυπο ενός κρυπτονομίσματος, ώστε να μπορεί να αποτελέσει πλατφόρμα, μέσω της οποίας να δημιουργούνται όλων των ειδών projects με βάση το

blockchain. Μάλιστα, πολλά από τα δεδομένα δεν είναι αποθηκευμένα στο ίδιο το blockchain, αλλά Modified Merkle Patricia Tries εξωτερικά αυτού. Το μόνο που αποθηκεύεται στο blockchain είναι η ρίζα (root) αυτού του δέντρου, με τρόπο ώστε να μην επηρεάζεται από την αλλαγή των δεδομένων. Το project του, όμως, έχει διαφορές ακόμα και στην πληρωμή των miners. Πάμε, πρώτα, να δούμε από τι αποτελείται ένα block στο Ethereum blockchain.

3.2.1 Blocks

Τα blocks στο Ethereum είναι τελείως διαφορετικά από τα blocks των τεσσάρων πεδίων ενός κλασσικού κρυπτονομίσματος. Στην πραγματικότητα, περιέχουν 15 πεδία τα οποία είναι:

- Previous hash
  Το hash value του προηγούμενου block, μέσω του οποίου ενώνονται και δημιουργείται η αλυσίδα.

- Nonce
  Ένας τυχαίος αριθμός 64-bits, ο οποίος χρησιμοποιήθηκε για το mining του block, και συγκεκριμένα, αυτός που ταίριαξε, ώστε να δοθεί έξοδος που συναντά τα κριτήρια που είχαν τεθεί. Αποδεικνύει, σε συνδυασμό με το mix hash που θα δούμε, ότι ένα επαρκές σύνολο υπολογιστικής ισχύος χρησιμοποιήθηκε για τη δημιουργία του συγκεκριμένου block.

- Timestamp
  Μια βαθμωτή τιμή ίση με την έξοδο της εντολής time του Unix, κατά τη γέννηση του block.

- Uncle’s hash
  Το hash value ενός block, για το οποίο θα μιλήσουμε στην επόμενη υποενότητα.

- Beneficiary
  Η διεύθυνση 160-bit στην οποία έχουν καταβληθεί όλα τα τέλη που συλλέγονται από την επιτυχημένη εξόρυξη αυτού του μπλοκ. Με λίγα λόγια, η διεύθυνση του miner, ο οποίος δημιούργησε το block.

- Logs bloom
  Ένα φίλτρο, ονομαζόμενο Bloom, το οποίο αποτελείται από πληροφορίες ευρετηρίου (διεύθυνση καταγραφής και θέματα καταγραφής), από τις οποίες αποτελούνται οι διάφορες συναλλαγές του συγκεκριμένου block.

- Difficulty
  Μια βαθμωτή τιμή που αντιστοιχεί στο επίπεδο δυσκολίας εύρεσης του συγκεκριμένου block. Προκύπτει από το επίπεδο δυσκολίας του προηγούμενου block και το timestamp αυτού του block.

- Extra Data
  Ένας αυθαίρετος πίνακας bytes, ο οποίος περιέχει δεδομένα σχετικά με αυτό το block. Πρέπει να αποτελείται από 32 bytes ή λιγότερα.

- Block Number
  Μία βαθμωτή τιμή ίση με τον αριθμό των προηγούμενων block στην αλυσίδα. Το genesis block, δηλαδή το πρώτο block έχει την τιμή αυτή ίση με 0.

- Gas Limit
  Μια βαθμωτή τιμή ίση με το συγκεκριμένο όριο δαπάνης gas ανά block. Για το gas θα μιλήσουμε εκτενώς σε επόμενη ενότητα, αλλά αποτελεί τη μονάδα μέτρησης υπολογιστικής ισχύος για την εκτέλεση των έξυπνων συμβολαίων (smart contracts).

- Gas Used
  Μια βαθμωτή τιμή ίση με το σύνολο του gas, που δαπανήθηκε για την εκτέλεση όλων των συναλλαγών, που περιέχονται στο συγκεκριμένο block.

- Mix hash
  Μια τιμή κατακερματισμού (hash value) 64-bits, η οποία σε συνδυασμό με το nonce, αποδεικνύει ότι έχει πραγματοποιηθεί επαρκής υπολογισμός για τη δημιουργία του συγκεκριμένου block.

- State Root
  Αποτελεί την τιμή κατακερματισμού (hash value), με χρησιμοποίηση του Keccak-256-bits, της ρίζας ενός δέντρου, ονόματι state trie, αφού όλες οι συναλλαγές έχουν εκτελεστεί και έχει έρθει σε τελική μορφή. Για το δέντρο αυτό, όπως και τα υπόλοιπα δύο θα μιλήσουμε στη συνέχεια.

- Transaction Root
  Αποτελεί την τιμή κατακερματισμού (hash value), με χρησιμοποίηση του Keccak-256-bits, της ρίζας ενός δέντρου, ονόματι transaction trie, το οποίο περιέχει κάθε συναλλαγή η οποία συμπεριλήφθηκε στο συγκεκριμένο block.

- Receipt Root
  Αποτελεί την τιμή κατακερματισμού (hash value), με χρησιμοποίηση του Keccak-256-bits, της ρίζας ενός δέντρου, ονόματι receipt trie, το οποίο περιέχει όλες τις αποδείξεις των συναλλαγών που συμπεριλήφθηκαν στο συγκεκριμένο block.

Ας δούμε τώρα εξονυχιστικά το nonce και το difficulty. Στο Ethereum blockchain υπάρχει ένας συγκεκριμένος τεχνητός χρόνος από τη δημιουργία ενός block, μέχρι τη δημιουργία του επόμενου. Ο χρόνος αυτός είναι πάντα περίπου 15 δευτερόλεπτα και υπάρχει ώστε να δημιουργείται ασφάλεια. Για τη διατήρηση του χρόνου αυτού στα ίδια επίπεδα ακόμη και όταν η υπολογιστική ισχύς που χρησιμοποιείται αλλάζει, χρησιμοποιείται το difficulty και το nonce. Συγκεκριμένα, χρησιμοποιείται ένας αλγόριθμος, ο οποίος αναλόγως με την κινητικότητα, δημιουργεί ένα στόχο για τους miners που ασχολούνται με την εύρεση του νεόυ block. Θέτει μια τιμή 256-bits, κάτω από την οποία πρέπει να βρίσκεται η τιμή κατακερματισμού (hash value) του συγκεκριμένου block. Η τιμή κατακερματισμού του block προκύπτει από τα data του block, το hash value του προηγούμενου block και το nonce. Τα δύο πρώτα είναι πάντα σταθερά, ενώ το τρίτο αλλάζει ώστε να φτάσει ένας miner να βρει μια τιμή μικρότερη της τιμής-στόχου. Δηλαδή, όσο μεγαλύτερο είναι το difficulty, τόσο περισσότερες δοκιμές nonce πρέπει να γίνουν, ώστε να φτάσει ένας miner στο επιθυμητό αποτέλεσμα.

Ωστόσο, υπάρχουν ορισμένες περιπτώσεις που η επιθυμία αυτή δεν εκπληρώνεται. Δύο blocks, δηλαδή, μπορεί να είναι έγκυρα και να δημιουργηθούν ακριβώς στον ίδιο χρόνο, γεγονός που ονομάζεται σύγκρουση block ή αλλιώς block clash. Όμως, μόνο το ένα από αυτά θα καταφέρει να γίνει μέρος του blockchain, ακόμα και αν τα δεδομένα στο άλλο block είναι τεχνικά έγκυρα. Στο Ethereum τα blocks που 'χάνουν' ονομάζονται Uncle's blocks. Σε αντίθεση με τα περισσότερα κρυπτονομίσματα, που τα block αυτά απλά χάνονται και δε γίνεται κανείς να αναφερθεί σε αυτά, στο Ethereum μπορεί να γίνει αναφορά σε αυτά από μερικά από τα επόμενα blocks, και μάλιστα, αν και τα δεδομένα τους δε χρησιμοποιούνται, μια μικρότερη ανταμοιβή για τον miner των συγκεκριμένων blocks δίνεται τότε. Αυτό γίνεται, διότι αποτελεί συχνό φαινόμενο η σύγκρουση blocks, στο Ethereum, καθώς ο χρόνος μεταξύ δύο block είναι πολύ μικρός. Οπότε, θέλει να ενθαρρύνει τους miners να συνεχίσουν να προσδίδουν αξία στο Ethereum, ακόμα και μετά από τέτοια συμβάντα και επίσης, επιτυγχάνει τη δημιουργία ασφάλειας, καθώς αναγνωρίζεται η εργασία που έγινε για τη δημιουργία τέτοιου είδους block.

3.2.2 Έξυπνα συμβόλαια (Smart Contracts)

Όλη η διαδικασία δημιουργίας της πλατφόρμας του Ethereum βασίστηκε στο πώς θα μπορεί να αποτελέσει τη βάση για τη δημιουργία αποκεντρωμένων εφαρμογών. Γέφυρα του blockchain με τη δημιουργία αποκεντρωμένων εφαρμογών αποτέλεσε η χρήση των έξυπνων συμβολαίων.

Τα έξυπνα συμβόλαια έχουν δύο χαρακτηριστικά, τα οποία τα καταστούν κατάλληλα για να εκπληρώσουν το συγκεκριμένο σκοπό. Το πρώτο είναι ότι είναι αμετάβλητα (immutable), δηλαδή αν ένα έξυπνο συμβόλαιο εκτελεστεί, ο κώδικάς του ποτέ δεν μπορεί να αλλάξει. Το δεύτερο χαρακτηριστικό είναι ότι είναι αποκεντρωμένα (decentralized), δηλαδή όταν κάποιος θέλει να εκτελέσει μια συνάρτηση ενός έξυπνου συμβολαίου, θα πρέπει, για να γίνουν αλλαγές, όντως, στην κατάστασή του, η πλειοψηφία των miners του Ethereum, να εγκρίνει τη συναλλαγή. Αυτά τα δύο χαρακτηριστικά προσφέρουν ασφάλεια, καθώς αν συμφωνηθεί κάτι, ούτε αλλάζει ποτέ, ούτε τα κριτήρια που έχουν οριστεί, μπορούν να προσπεραστούν, καθώς δε θα υπάρξει έγκριση.

Το πρότυπο που χρησιμοποιεί το Ethereum για να προσεγγίσει τη χρήση μια Turing-Complete γλώσσας μοιάζει με αυτό μιας εικονικής μηχανής (virtual machine). Για το λόγο αυτό, η γλώσσα στην οποία γράφονται τα έξυπνα συμβόλαια στο Ethereum ονομάζεται EVM (Ethereum Virtual Machine). Η γλώσσα αυτή μοιάζει με κώδικα μηχανής (assembly).

Για να μπορέσει κάποιος να εκτελέσει ή να καλέσει ένα έξυπνο συμβόλαιο θα πρέπει, να πληρώσει ένα ποσό, ίσο με αυτό που θα δαπανηθεί σε υπολογιστική ενέργεια από τους miners για την πραγματοποίηση του συγκεκριμένου σκοπού. Η μονάδα μέτρησης της ενέργειας αυτής ονομάζεται gas και η ποσότητα του gas, η οποία σπαταλάται για τη χρήση κάθε εντολής της EVM είναι καθορισμένη από τους δημιουργούς του Ethereum.

Το gas που χρησιμοποιείται μεταφράζεται σε Ether και τα χρήματα αυτά πηγαίνουν στον miner, ο οποίος εισήγαγε τη συναλλαγή αυτή στο block του. Οπότε στο Ethereum έχουμε άλλον έναν τρόπο, μέσω του οποίου ένας miner μπορεί να βγάλει κέρδος. Η τιμή του gas, όμως δεν είναι προκαθορισμένη. Κάθε χρήστης, ο οποίος δημιουργεί ή εκτελεί ένα

έξυπνο συμβόλαιο αποφασίζει τι ποσό θα πληρώσει ανά gas (gas price) που θα χρησιμοποιηθεί για τη συναλλαγή αυτή, αναλόγως με το πόσο γρήγορα θέλει να εκτελεστεί. Αυτό συμβαίνει, διότι, όπως είπαμε οι miners κυνηγάνε όσο το δυνατόν μεγαλύτερο κέρδος για την υπολογιστική ενέργεια που σπαταλούν, οπότε προφανώς θα εισάγουν ευκολότερα στο block τους συναλλαγές-εκτελέσεις συμβολαίων με μεγάλη τιμή gas. Επίσης, ο χρήστης καθορίζει και τη μέγιστη ποσότητα gas (gas limit), την οποία είναι διατεθειμένος να σπαταλήσει για την εκτέλεση μιας τέτοιας συναλλαγής.

Ο χρήστης πληρώνει ποσό σε Wei (1 Ether = 10^{15} Wei) ίσο με το γινόμενο gas price x gas limit. Αν το αίτημα εκτελεστεί επιτυχώς, δηλαδή συναντώνται τα κριτήρια που το συμβόλαιο ορίζει και ο miner το 'εξορύξει', τότε, αν δε χρησιμοποιηθεί όλο το gas, που ορίζεται από το gas limit, τα χρήματα που απομένουν επιστρέφονται στον χρήστη. Αν, όμως, το αίτημα δεν εκτελεστεί επιτυχώς, καθώς δε συναντώνται τα προαπαιτούμενα κριτήρια, σπαταλάται όλο το ποσό, ως φόρος, στον χρήστη που έκανε εσφαλμένη κλίση συμβολαίου. Αυτό είναι και άλλο ένα χαρακτηριστικό που προσφέρει ασφάλεια, καθώς οι χρήστες πρέπει να είναι σίγουροι κατά την κλίση ενός συμβολαίου, αλλιώς χάνουν άσκοπα σημαντικά ποσά.

Τα έξυπνα συμβόλαια αποτελούν σημαντικό μέρος της συγκεκριμένης εργασίας και θα μας απασχολήσουν σε όλο το υπόλοιπο κομμάτι της. Το πώς ο τρόπος που χρησιμοποιούνται στο Ethereum προσεγγίζει εικονική μηχανή θα γίνει αντιληπτό στις επόμενες υποενότητες.

3.2.3 Modified Merkle-Patricia Trie

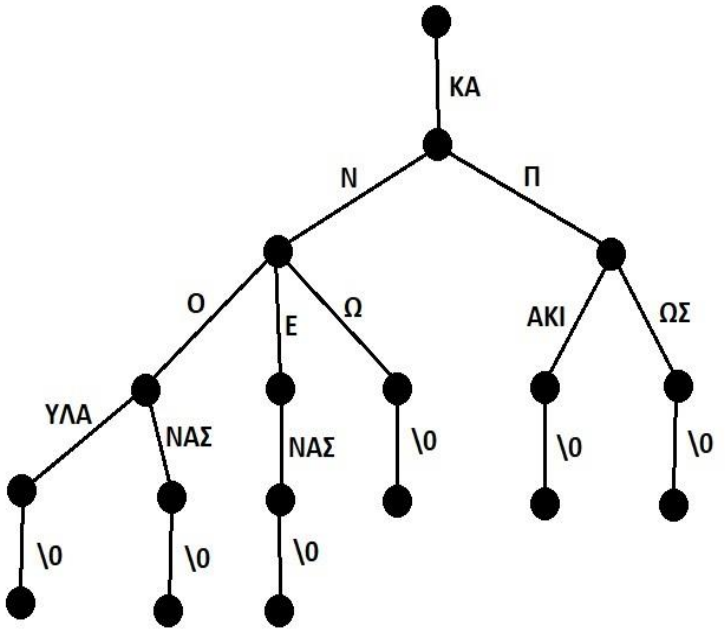
Τα πεδία που δεν αναφέρθηκαν και χρειάζονται ιδιαίτερη επεξήγηση είναι τα δέντρα που χρησιμοποιούνται για την αποθήκευση των διάφορων δεδομένων. Το block, όπως είδαμε, περιέχει τις ρίζες τριών δέντρων ως πεδία του. Στην πραγματικότητα, όμως, τα δέντρα που χρησιμοποιούνται στο Ethereum είναι 4. Όλα, όμως, είναι ίδιου είδους, Modified Merkle-Patricia Tries.

Το Modified Merkle-Patricia Trie αποτελεί δομή δεδομένων, η οποία πρωτοχρησιμοποιήθηκε στο Ethereum. Είναι ένας συνδυασμός Merkle-Tree, που παρουσιάστηκε στην προηγούμενη ενότητα και Patricia-Trie. Το Merkle Tree, όπως είδαμε, προσφέρει ασφάλεια στις συναλλαγές, καθώς, αν έστω ένα τμήμα του αλλάξει, αλλάζει και το hash value της ρίζας του, οπότε γίνεται αμέσως αντιληπτό. Είναι, όμως, αρκετά αργό στην εύρεση ενός φύλλου του, καθώς πρέπει κανείς να διασχίσει ολόκληρη τη διαδρομή που πάει προς το φύλλο αυτό ή να δοθούν σε κάποιον έτοιμοι κάποιοι κόμβοι του δέντρου. Όταν περιέχονται συναλλαγές μόνο ενός συγκεκριμένου block, ο μεγάλος αυτός χρόνος δε γίνεται αισθητός, καθώς είναι λίγα τα δεδομένα. Το Ethereum, όμως, για να μπορέσει να εκλπηρώσει το σκοπό του χρειάζεται μεγάλο όγκο δεδομένων. Το Patricia Trie (θα μελετηθεί αμέσως μετά) έχει το πλεονέκτημα ότι είναι πολύ πιο γρήγορο στην εύρεση δεδομένων, αλλά λιγότερο ασφαλές. Οπότε, ένας συνδυασμός των δύο αυτών δομών δεδομένων αποτελεί τη λύση σε όλα τα προβλήματα.

Το Patricia Trie (trie από retrieval) είναι μια δομή δεδομένων, η οποία χρησιμοποιεί προθέματα, ώστε να κάνει την ανάκτηση των δεδομένων πολύ πιο γρήγορη. Συγκεκριμένα, παίρνει τις συμβολοσειρές ως εισόδους και τσεκάρει αν κάποιες εξ' αυτών έχουν κοινά προθέματα, αρχίζουν δηλαδή από ίδια σύμβολα. Αρχίζει, σιγά σιγά να δημιουργείται ένα δέντρο, το οποίο ξεκινά από έναν κόμβο, ο οποίος πηγαίνει με το κοινό

πρόθεμα στους επόμενους κόμβους και αυτοί με το επόμενο κοινό πρόθεμα στους επόμενους και ούτω καθεξής. Όταν φτάσει μια συμβολοσειρά στο τέλος της, πηγαίνει σε έναν τελικό κόμβο με το σύμβολο '\0', που σημαίνει τέλος συμβολοσειράς. Η διαδικασία συνεχίζεται μέχρι όλες οι συμβολοσειρές να τερματίσουν.

Για να γίνει καλύτερα αντιληπτή η όλη διαδικασία, ας πάμε σε ένα παράδειγμα. Έστω ότι έχουμε τις συμβολοσειρές ΚΑΝΕΝΑΣ, ΚΑΝΟΝΑΣ, ΚΑΝΟΥΛΑ, ΚΑΝΩ, ΚΑΠΑΚΙ, ΚΑΠΩΣ και θέλουμε να σχηματίσουμε ένα Patricia Trie. Αυτό φαίνεται στο παρακάτω σχήμα:



Σχήμα 3.1: Παράδειγμα Patricia Trie

Όπως γίνεται αντιληπτό, στα δέντρα αυτά είναι πολύ γρήγορη η διαδικασία εύρεσης κάποιας συμβολοσειράς, καθώς γνωρίζοντας τη συμβολοσειρά, μπορεί κανείς να επισκέπτεται μόνο τους κόμβους που οδηγούνται προς αυτή και έτσι να βρει εύκολα, αν η συγκεκριμένη συμβολοσειρά βρίσκεται στο δέντρο. Ωστόσο, ενώ είναι ταχύτατα στην αναζήτηση, δεν είναι ασφαλή, καθώς κάποιος μπορεί να αλλάξει κάποιο σημείο, όπως παραδείγματος χάριν να κάνει το ΚΑΝΩ, ΚΑΝΗ και αυτό να μην αλλάξει τίποτα στο υπόλοιπο δέντρο.

Το Modified Merkle Patricia Trie του Ethereum χρησιμοποιεί αυτά ακριβώς τα πλεονεκτήματα των Merkle Trees και των Patricia Tries, χωρίς τα μειονεκτήματά τους, ώστε να φτάσει στο επιθυμητό αποτέλεσμα. Το δέντρο αυτό παίρνει ως είσοδο ζευγάρια κλειδιών-τιμών (key-value pairs) στο δεκαεξαδικό (hex) σύστημα, δηλαδή ένα mapping.

Περιλαμβάνει 4 είδη κόμβων:

1. Μηδενικό κόμβο ή Null-Node
   Ένας τέτοιος κόμβος αναπαριστά την άδεια συμβολοσειρά.

2. Κόμβο-Κλαδί ή Branch-Node
   Ένας τέτοιος κόμβος αποτελείται από όλα τα σύμβολα του δεκαεξαδικού συστήματος (0,..,9,a,…,f) και μία τιμή value, η οποία χρησιμοποιείται, αν κάποιο κλειδί καταλήγει σε τέτοιο κόμβο.

3. Κόμβο-Φύλλο ή Leaf-Node
   Ένας τέτοιος κόμβος αποτελείται από ένα πρόθεμα, ένα τέλος-κλειδιού και μια τιμή. Αποτελούν την κατάληξη όσων κλειδιών δεν τελειώνουν σε κάποιο κόμβο κλαδί.

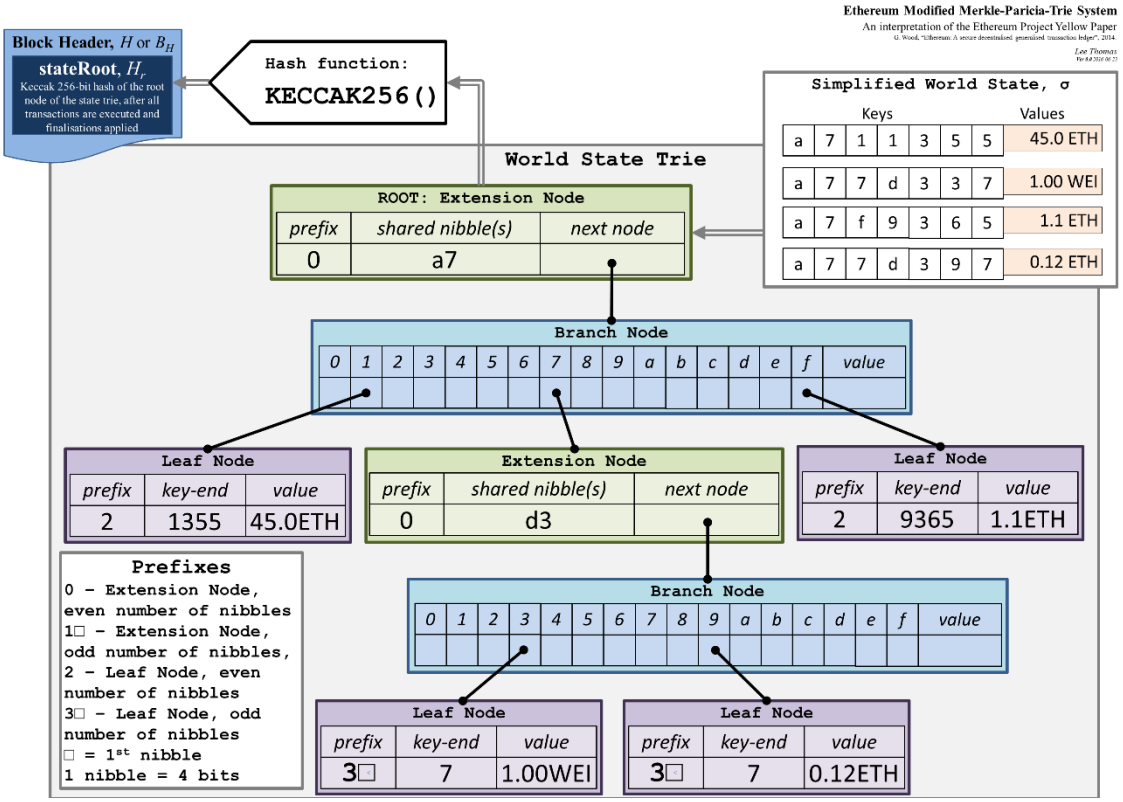
4. Κόμβο-Επέκταση ή Extension-Node
   Ένας τέτοιος κόμβος αποτελείται από ένα πρόθεμα, από κοινά nibbles και από τον επόμενο κόμβο.

Η διαδικασία δημιουργίας είναι η ακόλουθη. Μπαίνουν ως είσοδος κάποια ζευγάρια κλειδιών-τιμών. Στη συνέχεια ελέγχεται, όπως στο Patricia Trie, αν τα κλειδιά αυτά έχουν κοινά προθέματα. Δημιουργείται η ρίζα του δέντρου ως Extension-Node. Το πρόθεμα σε όλα τα παραπάνω δεν είναι ίδιο με το πρόθεμα στο Patricia Trie, αλλά τα κοινά nibbles είναι στην ουσία τα προθέματα. Δηλαδή, τα κοινά nibbles αναπαριστούν τη συμβολοσειρά που είναι κοινή στα κλειδιά που συνεχίζουν από το προηγούμενο βήμα, μέχρι το επόμενο. Τα προθέματα (prefixes), εδώ, αναπαριστούν τι είδους είναι ο συγκεκριμένος κόμβος και αν περιέχει περιττό ή άρτιο αριθμό nibbles. Υπάρχουν 5 είδη προθέματος σε ένα Modified Merkle Patricia Trie:

- 0, που σημαίνει κόμβος-επέκτασης, με άρτιο αριθμό nibbles
- 1□, που σημαίνει κόμβος-επέκτασης, με περιττό αριθμό nibbles
- 2, που σημαίνει κόμβος-φύλλο, με άρτιο αριθμό nibbles
- 3□, που σημαίνει κόμβος-φύλλο, με περιττό αριθμό nibbles
- □, που αντικατοπτρίζει το πρώτο nibble

Η ρίζα, έπειτα, συνδέεται με έναν κόμβο-κλαδί, ο οποίος ‘μοιράζει’ τα διάφορα σύμβολα του δεκαεξαδικού σε επόμενους κόμβους, οι οποίοι μπορεί να είναι οποιουδήποτε άλλου τύπου, εκτός από κόμβος-κλαδί. Αν, κάποια συμβολοσειρά-κλειδί καταλήγει σε κόμβο-κλαδί, τότε η τιμή value του κόμβου αυτού είναι και η τιμή value της συμβολοσειράς και συνδέεται με έναν μηδενικό κόμβο, ο οποίος αναπαριστά κενή συμβολοσειρά και δείχνει ότι κάποιο κλειδί κατέληξε σε κόμβο-κλαδί. Αν από κάποιο σύμβολο του δεκαεξαδικού, συνεχίζουν παραπάνω των δύο εναπομείναντων συμβολοσειρών-κλειδιών, τότε από το σύμβολο αυτό πηγαίνουμε σε έναν κόμβο-επέκτασης, ο οποίος δείχνει τα επόμενα κοινά σύμβολα των δύο συμβολοσειρών και συνεχίζει, όπως η ρίζα, ενώ αν από κάποιο σύμβολο συνεχίζει μόνο μία συμβολοσειρά-κλειδί, από το σύμβολο αυτό πηγαίνουμε σε κόμβο-φύλλο, όπου και καταλήγει το συγκεκριμένο κλειδί. Η διαδικασία αυτή συνεχίζεται, έως ότου όλες οι συμβολοσειρές-κλειδιά καταλήξουν. Στο παρακάτω σχήμα φαίνεται ένα τέτοιο δέντρο με τα ζευγάρια των κλειδιών-τιμών να δίνονται πάνω δεξιά.

Στην ουσία είναι ένα από τα τέσσερα τέτοια δέντρα που περιέχει το Ethereum, και συγκεκριμένα το state trie, το οποίο θα μελετήσουμε ακολούθως.



Σχήμα 3.2: Παράδειγμα Modified Merkle Patricia Trie (Ethereum State Trie)

Figure G. Source: Lee Thomas, https://ethereum.stackexchange.com/questions/268/ethereum-block-architecture, accessed 5th July 2019.

Όπως φαίνεται και στο σχήμα, το τροποποιημένο αυτό δέντρο είναι και ασφαλές, καθώς, αν αλλάξει οποιοδήποτε δεδομένο αλλάζει όλο το δέντρο, αλλά και γρήγορο στην εύρεση, καθώς γνωρίζοντας το κλειδί, μπορείς να βρεις ποιο μονοπάτι να ακολουθήσεις, για να φτάσεις στην τιμή.

3.2.4 States and Tries

Το Ethereum ξεπερνά το απλό μοντέλο των blocks και περιέχει, εκτός αυτών, και κάποια άλλα δεδομένα, ως Modified Merkle Patricia Tries, τα οποία βέβαια αντικατοπτρίζονται από τις ρίζες (roots) των διάφορων δέντρων και μέσα στα blocks. Τα δεδομένα αυτά, βέβαια, αλλάζουν με τις διάφορες συναλλαγές.

3.2.4.1 *Account State*

Στο Ethereum υπάρχουν δύο είδη λογαριασμών (accounts), οι εξωτερικοί λογαριασμοί ή Externally Owned Accounts (EOA) και οι λογαριασμοί συμβολαίων ή Contract Accounts. Οι εξωτερικοί λογαριασμοί είναι αυτοί τους οποίους διαθέτουν τα άτομα και οι οποίοι χρησιμοποιούνται για να στέλνουν και να λαμβάνουν Ether σε και από άλλους

εξωτερικούς λογαριασμούς, καθώς και για να εκτελούν έξυπνα συμβόλαια (smart contracts). Κάθε έξυπνο συμβόλαιο έχει και αυτό το δικό του λογαριασμό, ο οποίος είναι ο λογαριασμός συμβολαίου. Ένα Ethereum account περιέχει τέσσερα πεδία:

1) Nonce
   Είναι ένας αριθμός, ίσος με τον αριθμό των συναλλαγών ή των δημιουργιών συμβολαίων που έχουν πραγματοποιηθεί από τον συγκεκριμένο λογαριασμό.

2) Balance
   Το σύνολο των Ether που περιέχει ο συγκεκριμένος λογαριασμός, μετρημένο σε Wei (1 Ether = 1000000000000000 Wei).

3) storageRoot
   Η τιμή κατακερματισμού της ρίζας του account storage trie (θα το εξετάσουμε σε λίγο).

4) codeHash
   Η τιμή κατακερματισμού που οδηγεί στον κώδικα EVM του λογαριασμού.

Όλα τα παραπάνω πεδία, εκτός του codeHash, μπορούν να αλλάξουν, καθώς με νέες συναλλαγές δημιουργούνται νέα δεδομένα για τον λογαριασμό. Στους εξωτερικούς λογαριασμούς, το storageRoot είναι άδειο και το codeHash είναι τιμή κατακερματισμού για μία άδεια συμβολοσειρά. Στους λογαριασμούς συμβολαίων, όλα τα δεδομένα είναι αποθηκευμένα μέσα στο Account Storage Trie, το οποίο είναι ένα Merkle Patricia Trie, με κλειδιά τους λογαριασμούς συμβολαίων και τιμές τα δεδομένα του κώδικα. Η τιμή κατακερματισμού της ρίζας αυτού του δέντρου αποτελεί το πεδίο storageRoot, ενώ ο κώδικας του συμβολαίου αυτού είναι το πεδίο codeHash, αφου κρυπτογραφηθεί. Όπως γίνεται αντιληπτό, αποτέλεσμα, του ότι δεν αλλάζει το codeHash, είναι ότι ένα έξυπνο συμβόλαιο το οποίο περιέχει λάθη και έχει ήδη εκτελεστεί, δεν μπορεί να αντικατασταθεί από κάποιο νέο, καθώς τότε θα άλλαζε και το codeHash του λογαριασμού του. Αυτός είναι και ο λόγος που τα smart contracts πρέπει να τεστάρονται για τη λειτουργία τους εξονυχιστικά, πριν εκτελεστούν.

3.2.4.2 *World State*

Το World State είναι ένα mapping με κλειδί κάποιον λογαριασμό και τιμή το account state του λογαριασμού. Δηλαδή, δεδομένου ενός συγκεκριμένου λογαριασμού, δίνει τις πληροφορίες που αναφέραμε στο account state. Αναπαρίσταται ως ένα Merkle Patricia Trie, το οποίο, εκτός της ρίζας του, δεν αποτελεί μέρος του blockchain και ονομάζεται state trie. Για όλο το δίκτυο του Ethereum υπάρχει μόνο ένα state trie. Προφανώς, καθώς αλλάζουν τα δεδομένα των λογαριασμών μέσω των συναλλαγών, αλλάζει και το world state. Αν μπορούσαμε να φανταστούμε το δίκτυο του Ethereum ως έναν αποκεντρωμένο υπολογιστή, το world state θα αποτελούσε τον σκληρό του δίσκο. Οπότε, στο Ethereum δεν έχουμε όλα τα δεδομένα αποθηκευμένα μέσα στο blockchain, αλλά πολλά είναι εξωτερικά, που, όμως, οι αλλαγές σε αυτά πραγματοποιούνται μόνο αν μια συναλλαγή που τα επηρεάζει, μπει στο blockchain.

3.2.4.3 *Transaction and Transaction's Tries*

Υπάρχουν δύο κατηγορίες συναλλαγών, οι οποίες μπορούν να πραγματοποιηθούν στην πλατφόρμα του Ethereum, εκ των οποίων η μία περιέχει δύο υποκατηγορίες.

Οι κατηγορίες αυτές είναι:

A. Συναλλαγές μεταξύ ήδη υπαρχόντων λογαριασμών. Χωρίζεται σε δύο κατηγορίες, τις:

   I. Συναλλαγές μεταξύ δύο εξωτερικών λογαριασμών, δηλαδή μεταφορά αξίας από έναν κανονικό λογαριασμό σε έναν άλλο

   II. Συναλλαγές μεταξύ ενός εξωτερικού λογαριασμού και ενός λογαριασμού συμβολαίου

B. Δημιουργία ενός νέου λογαριασμού συμβολαίου, δηλαδή εκτέλεση ενός νέου smart contract

Μια συναλλαγή (Transaction) περιέχει τα εξής πεδία:

1. Nonce
   Αριθμός συναλλαγών που έχουν σταλεί από τον λογαριασμό που δημιούργησε τη συγκεκριμένη συναλλαγή.

2. gasPrice
   Αξία σε Wei του gas που σπαταλήθηκε για να πληρωθούν τα υπολογιστικά κόστη εκτέλεσης της συναλλαγής.

3. gasLimit
   Η μέγιστη ποσότητα gas, που είναι δυνατό να χρησιμοποιηθεί για την εκτέλεση της συναλλαγής.

4. to
   Αν αφορά συναλλαγή μεταξύ εξωτερικών λογαριασμών, το πεδίο αυτό είναι ο λογαριασμός στον οποίο θα μεταφερθούν τα χρήματα.
   Αν αφορά συναλλαγή μεταξύ εξωτερικού λογαριασμού και λογαριασμού συμβολαίου, δηλαδή συναλλαγή, όπου κάποιος εξωτερικός λογαριασμός κάλεσε κάποια συνάρτηση από το συμβόλαιο, το πεδίο αυτό είναι ο λογαριασμός του συμβολαίου.
   Αν αφορά τη δημιουργία ενός νέου συμβολαίου, το πεδίο αυτό είναι πάντα κενό.

5. Value
   Αν η συναλλαγή αυτή αφορά τη μεταφορά χρημάτων από ένα εξωτερικό λογαριασμό σε έναν άλλο, τότε το πεδίο αυτό είναι το ποσό σε Wei, που πρόκειται να μεταφερθεί.
   Αν η συναλλαγή αυτή αφορά την κλίση ενός συμβολαίου, τότε το πεδίο αυτό είναι το ποσό σε Wei, που θα πληρωθεί από τον λογαριασμό συμβολαίου, που λαμβάνει το μήνυμα.

Αν η συναλλαγή αυτή αφορά τη δημιουργία ενός νέου συμβολαίου, τότε το πεδίο αυτό είναι το ποσό σε Wei που θα προστεθεί στο υπόλοιπο του νεοδημιουργηθέντος συμβολαίου.

6. v,r,s
   Τιμές οι οποίες χρησιμοποιούνται στην κρυπτογραφημένη υπογραφή της συναλλαγής, ώστε να μπορεί να προσδιοριστεί ο λογαριασμός που έστειλε τη συναλλαγή.

7. Data
   Το πεδίο αυτό χρησιμοποιείται μόνο στην πρώτη κατηγορία συναλλαγών. Περιέχει τα δεδομένα εισόδου της συναλλαγής.

8. init
   Το πεδίο αυτό χρησιμοποιείται μόνο στη δεύτερη κατηγορία συναλλαγών. Περιέχει τον κώδικα EVM που χρησιμοποιήθηκε για την αρχικοποίηση του έξυπνου συμβολαίου.

Όταν μια συναλλαγή εκτελεστεί και το block, στο οποίο περιέχεται γίνει μέρος του Ethereum blockchain, δημιουργείται μία απόδειξη συναλλαγής (transaction receipt), η οποία περιέχει λεπτομέρειες για την εκτέλεση της συναλλαγής. Αυτό αναφέρεται, διότι υπάρχουν δύο Modified Merkle Patricia Tries, τα οποία δημιουργούνται για τις συναλλαγές. Τα δέντρα αυτά είναι το transaction trie και το transaction receipt trie.

Υπάρχει ένα transaction trie για κάθε block που γίνεται μέρος της αλυσίδας. Περιέχει όλες τις πληροφορίες για τις συναλλαγές του συγκεκριμένου block που αναφέρθηκαν παραπάνω. Η τιμή κατακερματισμού της ρίζας του δέντρου αυτού αποτελεί πεδίο του block, όπως είδαμε.

Υπάρχει ένα transaction receipt trie για κάθε block που γίνεται μέρος της αλυσίδας. Περιέχει αποδείξεις των συναλλαγών που συμπεριλήφθηκαν στο συγκεκριμένο block. Η τιμή κατακερματισμού της ρίζας του δέντρου αυτού αποτελεί, επίσης, πεδίο του block.

3.2.5 Σύνοψη και Mining

Συνοψίζοντας, είδαμε ότι το Ethereum αποθηκεύει δεδομένα και εντός και εκτός του blockchain του. Εκτός του blockchain χρησιμοποιούνται Merkle Patricia Tries, τα οποία κρατούν τα δεδομένα ασφαλή, αλλά ταυτόχρονα δίνουν τη δυνατότητα γρήγορης εύρεσης. Τα έξυπνα συμβόλαια αποθηκεύονται και εκτελούνται μέσα στο state trie, το οποίο είναι μοναδικό για ολόκληρο το blockchain και αλλάζει μέσω των συναλλαγών που εκτελούνται. Τα έξυπνα συμβόλαια, αν εκτελεστούν μία φορά, δεν μπορούν να αλλάξουν, όσον αφορά τη δομή του κώδικά τους, οπότε πρέπει να δημιουργηθεί εκ νέου νέο συμβόλαιο.

Το mining στο Ethereum πληρώνεται με τρεις τρόπους, από τη δημιουργία ενός block, από την εκτέλεση έξυπνων συμβολαίων και από τη δημιουργία ενός Uncle block. Οι φόροι που πληρώνονται στο Ethereum από τους λογαριασμούς που θέλουν να εκτελέσουν συναλλαγές είναι είτε κάποιος φόρος απλής συναλλαγής, είτε φόρος του οποίου η αξία πληρώνεται σε gas, για τη δημιουργία ή εκτέλεση έξυπνων συμβολαίων. Επίσης, το Ethereum έχει άλλον έναν τρόπο που μπορεί να πληρώσει, τα Uncle blocks.

Συγκεκριμένα, επειδή ο χρόνος μεταξύ της δημιουργίας δύο blocks είναι πολύ μικρός (περίπου 15 δευτερόλεπτα), πολλές φορές δύο miners βρίσκουν ένα έγκυρο block, σχεδόν ταυτόχρονα. Τότε, όπως είπαμε, το ένα γίνεται μέρος της αλυσίδας και το άλλο γίνεται Uncle block. Στην τοποθέτηση του μεθεπόμενου block στην αλυσίδα, ο miner μπορεί να επιλέξει να εισάγει αυτό το Uncle block στο κανονικό του block. Αν αυτό συμβεί, πληρώνεται ένα σημαντικό ποσό ο miner του Uncle block.

Τα blocks στο Ethereum προκύπτουν από τον αγώνα μεταξύ των miners, όπως ακριβώς περιγράφηκε στο κεφάλαιο 2. Εδώ, ο αλγόριθμος, μέσω του οποίου προσπαθούν να βρουν μια τιμή κρυπτογράφησης που να τηρεί τα κριτήρια, είναι ο Keccak-256-bits. Δοκιμάζουν κλασικά, διαφορετικά nonce, μέχρι να πετύχουν το αποτέλεσμα. Το ποσό που πληρώνεται ένας miner, για τη δημιουργία ενός κανονικού block, είναι περίπου 2 Ether (Ιούλιος 2019), ενώ στο ποσό αυτό προστίθενται οι φόροι των συναλλαγών, είτε αυτές είναι κανονικές συναλλαγές, είτε είναι εκτελέσεις έξυπνων συμβολαίων. Μέσα από ένα παράδειγμα θα γίνει περισσότερο αντιληπτή η διαδικασία.

Έστω ότι κατά τη διαδικασία εύρεσης του block 10000 δύο miners βρήκαν σχεδόν ταυτόχρονα δύο έγκυρα blocks, τα A10000 και B10000. Έστω ότι το block A10000 προστέθηκε στο blockchain και το B10000 έγινε Uncle Block. Ο miner του A10000 πληρώνεται 2 Ether και τους φόρους. Ο miner του B10000 περιμένει. Όταν εισαχθεί το block A10001, συνδέεται με το block A10000, οπότε και βλέπει ότι υπάρχει το B10000, επίσης. Κατά τη διαδικασία εύρεσης του A10002, οι miners συγκαταλέγουν στο block τους το B10000, ώστε να πληρωθεί ο miner του. Όταν το A10002 γίνει μέρος της αλυσίδας πληρώνεται ο miner του Uncle block, ένα ποσό περίπου ίσο με τα 7/8 του ποσού που δίνεται για ένα κανονικό block.

Δηλαδή, στο Ethereum γεννιούνται νομίσματα, όχι μόνο από τα blocks της αλυσίδας, αλλά και από τα Uncle blocks. Επίσης, όπως είπαμε οι δημιουργοί του είχαν δώσει στην αγορά κάποια pre-mined coins (72.000.000). Τέλος, το Ethereum δεν έχει, όπως άλλα κρυπτονομίσματα ένα συνολικό ποσό, το οποίο είναι και το όριο των coins που μπορούν να παραχθούν συνολικά, αλλά έχει ένα όριο στην ποσότητα των Ethers που μπορούν να παραχθούν σε έναν χρόνο, το οποίο είναι ίσο με το 25% του ποσού που δόθηκε στην ICO του. Δηλαδή 18.000.000 Ethers, το πολύ, μπορούν να παράγονται κάθε χρόνο.

Πάμε, όμως, να δούμε πώς καθορίζονται τα tokens, καθώς και τα εργαλεία με τα οποία κάποιος μπορεί να δημιουργήσει τη δικιά του αποκεντρωμένη εφαρμογή πάνω στο blockchain του Ethereum.

3.3 Αποκεντρωμένες εφαρμογές (Dapps) στο Ethereum

Το Ethereum δημιουργήθηκε με σκοπό να αποτελέσει πλατφόρμα πάνω στην οποία να δημιουργούνται αποκεντρωμένες εφαρμογές. Για να έχουν λόγο ύπαρξης οι αποκεντρωμένες εφαρμογές, συνήθως, χρησιμοποιούν το δικό τους νόμισμα-αντάλλαγμα, το οποίο ζει και έχει αξία μόνο εντός της αποκεντρωμένης εφαρμογής, το λεγόμενο token. Για τα tokens μιλήσαμε και στο πρώτο κεφάλαιο, αλλά πάμε να δούμε πώς ακριβώς λειτουργούν στο Ethereum.

3.3.1 Ethereum Tokens

Τα tokens, όπως αναφέραμε και στην πρώτη ενότητα, μοιάζουν περισσότερο με μετοχές παρά με νομίσματα. Υπάρχουν διάφορα είδη tokens, ενώ κάθε είδος token εξυπηρετεί συγκεκριμένους σκοπούς. Κάθε token στο Ethereum έχει αξία βάσει του Ether. Η αξία αυτή καθορίζεται, ανάλογα με το είδος του token από διαφορετικούς παράγοντες. Η αξία των security tokens είναι άρρηκτα συνδεδεμένη με την ονομαστική αξία της εταιρείας, η οποία τα χρησιμοποιεί. Η αξία των utility tokens, καθορίζεται μόνο από την αρχή της προσφοράς και της ζήτησης στην αγορά. Τα utility tokens είναι και αυτά που αποτελούν τα tokens των αποκεντρωμένων εφαρμογών.

Ένα token είναι στην ουσία ένα έξυπνο συμβόλαιο, το οποίο περιέχει συναρτήσεις για τις διάφορες χρήσεις του token αυτού. Από τότε που δημιουργήθηκε το Ethereum, ξεκίνησαν να δημιουργούνται τέτοια συμβόλαια. Ωστόσο, κάθε ένα από αυτά τα συμβόλαια χρησιμοποιούσε συναρτήσεις, οι οποίες προέκυπταν καθαρά και μόνο από τον δημιουργό τους, οπότε όλα ήταν τελείως διαφορετικά στη δομή τους. Το γεγονός αυτό, όπως ήταν αναμενόμενο, οδήγησε σε πολλές καταστροφές αποκεντρωμένων εφαρμογών, λόγω της ύπαρξης λαθών (bugs) στον κώδικά τους. Και όταν μιλάμε για λάθη, μιλάμε για λάθη τα οποία οδηγούσαν σε κλοπές των tokens, που μεταφράζονται σε Ethers, που με τη σειρά τους μεταφράζονται σε πραγματικά χρήματα. Η μεγαλύτερη, μάλιστα, κλοπή ήταν στον οργανισμό DAO (Decentralized Autonomous Organization), η οποία προέκυψε από την εύρεση μιας τρύπας σε έναν βρόχο (loop hole) στον κώδικα. Κλάπηκαν, σε μόλις λίγες ώρες tokens αξίας 3,6 εκατομμυρίων Ether, που μεταφράζονταν στην εποχή εκείνη (17 Ιουνίου του 2016) σε 70 εκατομμύρια δολάρια. Καταλαβαίνει κανείς πως, ενώ το χτίσιμο του Ethereum δε συντέλεσε στην κλοπή αυτή, η ανάγκη για έναν κοινό κορμό στη δημιουργία tokens γινόταν ολοένα και μεγαλύτερη.

Το πρώτο κοινά αποδεκτό πρωτόκολλο προτάθηκε στις 19 Νοεμβρίου του 2015 και ήταν το ERC-20 protocol. Ακόμα και σήμερα είναι το πιο συνηθισμένο πρωτόκολλο για τη δημιουργία ενός token και υποστηρίζεται από όλα, σχεδόν, τα site συναλλάγματος tokens. Το πρωτόκολλο αυτό περιέχει κάποιες βασικές συναρτήσεις, οι οποίες καθορίζουν τις διάφορες λειτουργίες ενός token όσον αφορά τις συναλλαγές, καθώς και κάποια συμβάντα (events), τα οποία πραγματοποιούνται σε ορισμένες περιπτώσεις. Για να θεωρηθεί ότι ένα token ακολουθεί το συγκεκριμένο πρωτόκολλο πρέπει αναγκαστικά στο έξυπνο συμβόλαιο, μέσα στο οποίο ορίζεται, να περιέχονται απαραίτητα οι βασικές αυτές συναρτήσεις, αλλά και όποιες άλλες ο χρήστης προτιμά. Οι βασικές συναρτήσεις του συγκεκριμένου πρωτοκόλλου είναι οι εξής:

- **function totalSupply() constant returns (uint256 totalSupply)**
  Η συνάρτηση αυτή επιστρέφει σε αυτόν που την κάλεσε τη συνολική ποσότητα των tokens που έχουν δημιουργηθεί.

- **function balanceOf(address account) constant returns (uint256 balance)**
  Η συνάρτηση αυτή επιστρέφει σε αυτόν που την κάλεσε την ποσότητα των tokens που περιέχει η διεύθυνση account.

- **function transfer(address to, uint256 value) returns (bool success)**
  Η συνάρτηση αυτή στέλνει από τον λογαριασμό που την κάλεσε στον λογαριασμό to ποσότητα tokens ίση με το value. Επιστρέφει μια τιμή true ή false για επιτυχία και αποτυχία, αντίστοιχα.

- **function transferFrom(address from, address to, uint256 value) returns (bool success)**
  Η συνάρτηση αυτή στέλνει από τον λογαριασμό from στον λογαριασμό to ποσότητα tokens ίση με το value. Αυτή η συνάρτηση θα επιτύχει μόνο, αν ο λογαριασμός from έχει δώσει αρμοδιότητα στον λογαριασμό που την κάλεσε να ανταλλάσσει tokens εκ μέρους του. Επιστρέφει μια τιμή true ή false για επιτυχία και αποτυχία, αντίστοιχα.

- **function approve(address spender, uint256 value) returns (bool success)**
  Η συνάρτηση αυτή δίνει την αρμοδιότητα στον λογαριασμό spender να ανταλλάσσει ποσότητα tokens ίση με το value, που ανήκουν στον λογαριασμό του χρήστη που την κάλεσε. Επιστρέφει μια τιμή true ή false για επιτυχία και αποτυχία, αντίστοιχα.

- **function allowance(address owner, address spender) constant returns (uint256 remaining)**
  Η συνάρτηση αυτή επιστρέφει σε αυτόν που την κάλεσε την ποσότητα των tokens που απομένουν, τα οποία μπορεί ο λογαριασμός spender να ανταλλάσσει εκ μέρους του λογαριασμού owner.

Τα συμβάντα (events) είναι:

- **event Transfer(address from, address to, uint256 value)**
  Το event αυτό ειδοποιεί τον χρήστη που εκτέλεσε μια συνάρτηση μεταφοράς tokens ότι η συναλλαγή ολοκληρώθηκε και του παρουσιάζει ότι ο λογαριασμός from έστειλε στον λογαριασμό to ποσότητα tokens ίση με το value.

- **event Approval(address spender, uint256 value)**
  Το event αυτό ειδοποιεί τον χρήστη που κάλεσε τη συνάρτηση approve, ότι έδωσε αρμοδιότητα στον λογαριασμό spender να ανταλλάξει ποσότητα tokens ίση με το value εκ μέρους του.

Το πρωτόκολλο αυτό, αν χρησιμοποιηθεί αυτούσιο, χωρίς καμία παραπάνω συνάρτηση ή συμβάν, μπορεί να οδηγήσει σε απώλεια χρημάτων. Αυτό συμβαίνει διότι, καταρχάς, ο παραλήπτης δεν μπορεί να αναγνωρίσει την εισερχόμενη συναλλαγή, καθώς τα συμβάντα στέλνονται σε αυτόν που εκτελεί τη συναλλαγή. Ως αποτέλεσμα, αν ο παραλήπτης είναι λογαριασμός συμβολαίου και όχι εξωτερικός λογαριασμός, τα tokens πηγαίνουν κάπου, όπου δεν υπάρχει τρόπος χρήσης τους. Πιο συγκεκριμένα, ένα έξυπνο συμβόλαιο, όπως ξέρουμε, αν εκτελεστεί, δεν μπορεί να αλλάξει ποτέ. Οπότε, αν λάβει κάποιο token και δεν έχει συνάρτηση, ώστε να το πουλήσει ή να δώσει αρμοδιότητα σε κάποιον άλλον λογαριασμό να το πουλήσει εκ μέρους του, το token αυτό ανήκει σε έναν λογαριασμό, ο οποίος δεν μπορεί να το χρησιμοποιήσει, οπότε αχρηστεύεται. Έχουν χαθεί με τον τρόπο αυτό tokens αξίας τουλάχιστον 3.000.000 δολλαρίων.

Για τον λόγο αυτό, προτάθηκε ένα νέο πρωτόκολλο το ERC223, τον Μάρτιο του 2017. Το πρωτόκολλο αυτό περιέχει μια παραπάνω συνάρτηση, η οποία με πολύ έξυπνο τρόπο λύνει το πρόβλημα. Η συνάρτηση αυτή είναι η εξής (αναφέρεται μόνο αυτή, καθώς οι άλλες δεν μας ενδιαφέρουν):

- **function tokenFallback (address from, uint value, bytes data)**
  Η συνάρτηση αυτή καλείται κάθε φορά που μία συναλλαγή πυροδοτείται. Η συναλλαγή πυροδοτήθηκε από τον λογαριασμό from με σκοπό να στείλει ποσότητα tokens ίση με το value. Το data αφορά τα δεδομένα που περιέχει ο λογαριασμός στον οποίο πηγαίνουν τα tokens. Αν το data είναι κενό, σημαίνει ότι ο λογαριασμός αυτός είναι εξωτερικός λογαριασμός, ενώ αν υπάρχουν data σημαίνει ότι είναι λογαριασμός συμβολαίου. Στην πρώτη περίπτωση, η συναλλαγή προχωρά, ενώ στη δεύτερη περίπτωση, ελέγχονται τα δεδομένα του συμβολαίου-παραλήπτη. Αν παρατηρηθεί ότι το συμβόλαιο αυτό είναι προετοιμασμένο να δεχτεί tokens, η συναλλαγή προχωράει, ενώ αν όχι, η συναλλαγή διακόπτεται και η ποσότητα των tokens επιστρέφεται στον κάτοχό της.

Γενικά, πολλά πρωτόκολλα έχουν προταθεί όλα τα χρόνια από τη δημιουργία του Ethereum, όμως, ακόμη και σήμερα, αυτό που υποστηρίζεται από τα περισσότερα site συναλλάγματος είναι το ERC-20. Συνηθίζεται, όμως, εκτός των βασικών του συναρτήσεων, να περιέχεται και κάποιος τρόπος με τον οποίο το συμβόλαιο του token να διαχειρίζεται καταστάσεις, όπως η παραπάνω. Το Gaming Token, δηλαδή το token που δημιουργήθηκε στα πλαίσια της συγκεκριμένης εφαρμογής, ακολουθεί το ERC-20 πρωτόκολλο, έχοντας όμως μία συνάρτηση διαχείρισης, όπως είναι η tokenFallback.

3.3.2 Εργαλεία για τη δημιουργία αποκεντρωμένων εφαρμογών στο Ethereum

Για τη δημιουργία και τη σωστή χρήση αποκεντρωμένων εφαρμογών (Dapps) στο Ethereum έχουν προταθεί πάμπολλα εργαλεία, τα οποία συνεχώς εξελίσσονται και αυξάνονται. Στα πλαίσια της συγκεκριμένης εργασίας θα αναφερθούν μόνο αυτά που χρησιμοποιήθηκαν για την εκπόνησή της.

Ethereum Blockchain

Το blockchain του Ethereum αποτελεί στην ουσία το περιβάλλον στο οποίο ζουν οι εφαρμογές αυτές. Μάλιστα, οι αποκεντρωμένες εφαρμογές το χρησιμοποιούν ως βάση δεδομένων και συνήθως δε χρησιμοποιούν καν κανονικές βάσεις. Αυτός είναι και ο κύριος λόγος που είναι αποκεντρωμένες. Τα συμβόλαια των αποκεντρωμένων εφαρμογών στο Ethereum, γράφονται σε EVM. Σκοπός όλων των υπόλοιπων εργαλείων είναι κάποια στιγμή η εφαρμογή να γίνει μέρος του πραγματικού Ethereum δικτύου. Απλά, αν ένα έξυπνο συμβόλαιο μπει στο δίκτυο αυτό, όπως προείπαμε, δεν αλλάζει πάλι ποτέ.

Solidity

Η solidity είναι μία υψηλού επιπέδου (high-level) γλώσσα προγραμματισμού, μέσω της οποίας μπορούν να δημιουργηθούν έξυπνα συμβόλαια. Έχει παρόμοια σύνταξη με τη scripting γλώσσα της Javascript. Φτιάχτηκε, ώστε να ενισχύσει την EVM (Ethereum Virtual Machine) και να δώσει τη δυνατότητα στους προγραμματιστές να γράφουν έξυπνα συμβόλαιο σε κανονική γλώσσα και όχι σε τύπου assembly γλώσσα. Οι διάφορες εντολές της μεταφράζονται σε EVM, μέσω της οποίας τρέχουν τα έξυπνα συμβόλαια. Είναι μια στατική scripting γλώσσα, η οποία διεξάγει τη διαδικασία της επαλήθευσης και επιβολής των περιορισμών κατά το χρόνο σύνταξης (compile time) και όχι κατά το χρόνο

εκτέλεσης (run time). Έχει και τις δυνατότητες του αντικειμενοστραφή προγραμματισμού, όπως η αφαιρετικότητα (abstraction), η κληρονομιά (inheritance), ο πολυμορφισμός (polymorphish), η κλάση (class), η διεπαφή (interface) και η ενθυλάκωση (encapsulation). Γενικά, είναι μια γλώσσα με πολλές δυνατότητες που συνδυάζει τον αντικειμενοστραφή προγραμματισμό, το scripting, αλλά και όσες λειτουργίες κρίνονται απαραίτητες για ένα συμβόλαιο που τρέχει στο Ethereum.

Ganache

Το ganache είναι ένα εργαλείο, μέσω του οποίου ο χρήστης μπορεί να δημιουργήσει το δικό του προσωπικό τοπικό blockchain, το οποίο λειτουργεί με τον ίδιο ακριβώς τρόπο που λειτουργεί το πραγματικό Ethereum blockchain, με τη διαφορά ότι δε χρησιμοποιεί miners, αλλά ο χρήστης μπορεί να εισάγει πόσος χρόνος θέλει να μεσολαβεί από τη δημιουργία ενός block, μέχρι τη δημιουργία ενός άλλου. Χρησιμοποιείται ώστε να δοκιμαστεί η λειτουργία των έξυπνων συμβολαίων ή της εφαρμογής που είναι χτισμένα πάνω στο Ethereum, πριν αυτά πυροδοτηθούν στο πραγματικό blockchain. Δίνει τη δυνατότητα στον χρήστη να εισάγει όποιες επιλογές θέλει, ώστε να προσομοιώσει όπως ακριβώς θέλει τις εφαρμογές του.

Truffle

Το Truffle είναι ένα αναπτυξιακό περιβάλλον, δοκιμαστικό πλαίσιο (testing framework) και μέσο επικοινωνίας με το Ethereum, που έχει σκοπό να βελτιώσει και να διευκολύνει τον τρόπο με τον οποίο ένας προγραμματιστής αλληλεπιδρά με αυτό. Συγκεκριμένα, είναι ένα μέσο το οποίο επικοινωνεί με το ganache και μέσω του οποίου καθίσταται δυνατή η μετατροπή των έξυπνων συμβολαίων, τα οποία είναι γραμμένα σε solidity, στη γλώσσα που αντιλαμβάνεται το blockchain. Τα κάνει compile, τα συνδέει, τα αναπτύσσει και διαχειρίζεται τον δυαδικό τους κώδικα. Μέσω του truffle, δίνεται η δυνατότητα στον προγραμματιστή να εκτελέσει όλα του τα συμβόλαια μαζί και να φτιάξει ξεχωριστά tests για το καθένα, που με μία εντολή αυτόματα θα εκτελεστούν όλα. Επίσης, μπορεί κάποιος να επικοινωνήσει άμεσα με ένα έξυπνο συμβόλαιο ενός τοπικού blockchain, δηλαδή να του αλλάξει την κατάσταση. Το truffle χρησιμοποιεί ως γλώσσα την Javascript, και κυρίως το κομμάτι των υποσχέσεων (promises). Τέλος, να αναφερθεί ότι το truffle δε χρησιμοποιείται στο πραγματικό blockchain, αλλά αν συνδυαστεί με το ganache και το web3 που θα εξεταστεί αμέσως μετά, δίνει τη δυνατότητα ακριβούς προσομοίωσης για δοκιμή στον κώδικα ενός έξυπνου συμβολαίου.

Node.js and Web3

Το node.js είναι ένα περιβάλλον ανοιχτού κώδικα, το οποίο παρέχει διάφορες βιβλιοθήκες στους χρήστες του, με μία από αυτές να είναι το web3. Το web3 είναι μια συλλογή από βιβλιοθήκες, η οποία επιτρέπει στους χρήστες να αλληλεπιδράσουν με έναν τοπικό ή και απομακρυσμένο κόμβο του Ethereum, χρησιμοποιώντας σύνδεση HTTP, WebSocket ή IPC. Έχει όλες τις δυνατότητες επικοινωνίας και με το πραγματικό blockchain. Μέσω αυτού, μπορεί κάποιος να εντοπίσει ένα συμβόλαιο, δίνοντας δύο πληροφορίες, το Abi του συμβολαίου και την διεύθυνση του λογαριασμού του και έπειτα να επικοινωνήσει μαζί του. Το web3 χρησιμοποιεί τη γλώσσα Javascript για τη χρήση του και τις εντολές του.

Metamask

Η εφαρμογή Metamask είναι η κύρια εφαρμογή, μέσω της οποίας οι χρήστες διαχειρίζονται τους λογαριασμούς τους στο Ethereum Blockchain, δηλαδή είναι μια frontend εφαρμογή. Δίνει δυνατότητα επιλογής και τοπικού δικτύου, οπότε ο

προγραμματιστής μπορεί να εξετάζει πώς το site του επικοινωνεί με τον χρήστη, ώστε να έχει σφαιρική άποψη για την αποκεντρωμένη εφαρμογή του.

4 Gaming Token

4.1 Εισαγωγή

Το Gaming Token είναι το όνομα της αποκεντρωμένης εφαρμογής που δημιουργήθηκε στα πλαίσια της συγκεκριμένης διπλωματικής εργασίας. Ο γενικός σκοπός της εφαρμογής αυτής είναι να εντάξει στα skill-based παιχνίδια τον ανταγωνισμό με πραγματικά χρήματα, τα οποία είναι μεταφρασμένα σε tokens και μέσω αυτού του γεγονότος να γίνει κερδοφόρα. Η εφαρμογή αποτελείται από δύο διαφορετικά είδη tokens, που και τα δύο είναι utility tokens. Το ένα είδος token έχει σχέση μόνο με αυτήν την εφαρμογή και πωλείται μόνο σε επενδυτές. Στο άλλο είδος, ανήκουν τα tokens, τα οποία έχει κάθε παιχνίδι το οποίο γίνεται μέρος της εφαρμογής αυτής.

Η εφαρμογή αυτή χρησιμοποιεί όλα τα πλεονεκτήματα και τις δυνατότητες που η πλατφόρμα Ethereum προσφέρει. Δε χρησιμοποιεί βάσεις δεδομένων, καθώς έχει το blockchain για τη διατήρηση των πληροφοριών και δεν ανήκει σε κανέναν για τον ίδιο λόγο. Επιπλέον, είναι απολύτως ασφαλής όσον αφορά τις συναλλαγές, καθώς για να μπορέσει κάποιος να κάνει tampering με αυτές, πρέπει να ξεγελάσει την πλειοψηφία miners, γεγονός περίπου αδύνατο, ενώ μπορεί οποιοσδήποτε χρήστης να ελέγξει πώς πραγματοποιήθηκε η συναλλαγή. Επίσης, οι συναλλαγές γίνονται αυτοματοποιημένα από τα έξυπνα συμβόλαια που έχουν καθοριστεί, οπότε δεν υπάρχει περίπτωση κανενός λάθους. Οι συναλλαγές πραγματοποιούνται άμεσα, γιατί τα tokens που ο καθένας κερδίζει ή αγοράζει μεταφέρονται στον λογαριασμό του, τη στιγμή που ένας miner εισάγει τη συναλλαγή στο block του, άρα μιλάμε για μερικά λεπτά. Οι φόροι είναι ελάχιστοι και καθορίζονται από το gas που χρησιμοποιείται για να εκτελεστεί ο κώδικας που αφορά τη συναλλαγή.

4.2 Γενική ιδέα

Η αποκεντρωμένη αυτή εφαρμογή ξεκινά με τη δημιουργία ενός token, του gaming token ή GT. Το gaming token είναι για την εφαρμογή ό,τι είναι και οι μετοχές για κάποια εταιρεία. Για την πώληση των gaming tokens χρησιμοποιείται ICO (Initial Coin Offering), με έναν όμως παράδοξο τρόπο. Αρχικά, δίνεται η δυνατότητα σε εταιρείες ηλεκτρονικών παιχνιδιών να δημιουργήσουν τα δικά τους tokens. Αυτό πραγματοποιείται, μέσω ενός έξυπνου συμβολαίου, το οποίο δημιουργεί καινούρια tokens με όνομα καθορισμένο από την εταιρεία, τα οποία αφορούν ένα συγκεκριμένο παιχνίδι της. Επίσης, η εταιρεία καθορίζει και τις μορφές παιχνιδιού, οι οποίες διατίθενται για το συγκεκριμένο σκοπό (π.χ. 2 ομάδες με 5 παίχτες η κάθε ομάδα).

Όταν μια εταιρεία ηλεκτρονικών παιχνιδιών δημιουργήσει αυτά τα tokens, ίδια ακριβώς ποσότητα gaming tokens, με αυτά που δημιουργήθηκαν, γίνονται διαθέσιμα στην ICO του gaming token για τους επενδυτές. Αυτό συμβαίνει ώστε να αποκτούν αξία τα gaming tokens, που έπειτα βγαίνουν στην αγορά και διατίθενται στους επενδυτές. Οι επενδυτές αγοράζουν με μία καθορισμένη τιμή τα gaming tokens κατά την ICO.

Η ICO, δηλαδή, δεν περιέχει έναν καθορισμένο αριθμό tokens που διατίθενται στην αγορά, αλλά ο αριθμός αυτός καθορίζεται από το πόσα tokens χρησιμοποιούνται από παιχνίδια που έγιναν μέρος της εφαρμογής. Το πόσα tokens μπορούν οι εταιρείες ηλεκτρονικών παιχνιδιών να δημιουργήσουν, από την άλλη, είναι καθορισμένα. Αυτό συμβαίνει, ώστε ο δημιουργός του Gaming Token, να μπορεί να ελέγξει πόσα από τα

gaming tokens θα βγουν στην αγορά, διότι, αφού το gaming token είναι utility token, όταν ένα επενδυτής έχει κάποιο ποσοστό από το σύνολο των gaming tokens, έχει το ίδιο ποσοστό βαρύτητας η γνώμη του στις αποφάσεις για το μέλλον της αποκεντρωμένης εφαρμογής.

Τις περιόδους που δεν διατίθενται tokens προς πώληση από την εφαρμογή, οι επενδυτές μπορούν να ανταλλάσουν tokens μεταξύ τους με αντάλλαγμα Ether, μέσω εφαρμογών αγοραπωλησίας tokens. Η τιμή του gaming token δηλαδή, μπορεί να ανέβει ή να πέσει, αναλόγως με την προσφορά και τη ζήτησή του στην αγορά.

Όταν κάποια εταιρεία έχει το δικό της token, χρησιμοποιεί τα έξυπνα συμβόλαια που της παρέχει η εφαρμογή, ώστε να μπορούν οι παίχτες να ανταγωνιστούν με πραγματικά χρήματα. Τα tokens των εταιρειών μπορούν να συναλλαχθούν μόνο εντός του έξυπνου συμβολαίου του εκάστοτε token, και με κανέναν άλλον τρόπο. Αυτό συμβαίνει, ώστε η αξία του token να παραμένει σταθερή για πάντα και ο μόνος τρόπος μέσω του οποίου οι παίχτες να μπορούν να βγάλουν κέρδος ή το αντίθετο, να είναι παίζοντας το παιχνίδι.

Οι παίχτες, μπορούν να αγοράσουν tokens, μπορούν να πουλήσουν tokens και μπορούν να προσφέρουν tokens σε άλλους παίχτες. Επίσης, μπορούν να χρησιμοποιήσουν τα tokens τους για να παίξουν το εκάστοτε παιχνίδι, με σκοπό να κερδίσουν και να αποκομίσουν κάποιο χρηματικό κέρδος. Όταν οι παίχτες έχουν αγοράσει tokens και ξεκινούν την εύρεση παιχνιδιού, η εφαρμογή ξεκινά την αναζήτηση έγκυρου παιχνιδιού. Πριν βρεθεί το παιχνίδι, επιλέγεται ένας αρχηγός για κάθε ομάδα. Όταν βρεθεί το παιχνίδι, επιλέγεται τυχαία κάποιος από τους αρχηγούς, για να αλληλεπιδράσει με το συμβόλαιο του παιχνιδιού. Όταν ο παίχτης αυτός δεχτεί, εκτελείται μια συνάρτηση του συμβολαίου, μέσω της οποίας αποσπάται το ανάλογο ποσό από όλους τους παίχτες και δημιουργείται ένα νέο συμβόλαιο μέσα στο οποίο αποθηκεύονται τα λεφτά των παιχτών και οι πληροφορίες τους.

Όταν το παιχνίδι τελειώσει, ο αρχηγός της νικήτριας ομάδας λαμβάνει ένα μήνυμα, μέσω του οποίου αλληλεπιδρά με το συμβόλαιο που δημιουργήθηκε για το παιχνίδι και οι νικητές λαμβάνουν το ποσό που τους αναλογεί, αναλόγως με το πόσες ομάδες συμμετείχαν στο συγκεκριμένο παιχνίδι και τον φόρο που λαμβάνει η εταιρεία του παιχνιδιού.

Ας εξετάσουμε, όμως, όλα τα οικονομικά δεδομένα, ώστε να δούμε γιατί η αποκεντρωμένη αυτή εφαρμογή είναι συμφέρουσα για όλους τους συμμετέχοντες. Οι κατηγορίες ατόμων που αποτελούν μέρος της εφαρμογής είναι 4 και εξετάζονται τα οικονομικά δεδομένα κάθε κατηγορίας παρακάτω:

- <u>Εταιρείες ηλεκτρονικών παιχνιδιών</u>
  Οι εταιρείες αυτές στην αρχή δημιουργούν τα tokens τους. Έπειτα, κάθε φορά που ένα παιχνίδι παίζεται, παίρνουν ένα ποσοστό από τα tokens που παίχτηκαν. Πληρώνονται αρχικά σε tokens και μετά μπορούν να κάνουν ανάληψη και να τα μετατρέψουν σε Ether. Εδώ, υπάρχουν ορισμένες λεπτομέρειες που επηρεάζουν τη συνολική λειτουργία της εφαρμογής.
  Προφανώς, όταν μια εταιρεία δημιουργεί tokens, τα tokens αυτά πηγαίνουν σε έναν εξωτερικό λογαριασμό που κατέχει η εταιρεία, ενώ η διαχείρισή τους πραγματοποιείται μέσω του έξυπνου συμβολαίου που τα αφορά. Από τον λογαριασμό αυτό, μεταφέρονται τα tokens στους λογαριασμούς των παιχτών που

τα αγοράζουν, όμως τα Ethers με τα οποία οι παίχτες αγοράζουν τα tokens πηγαίνουν στον λογαριασμό συμβολαίου του έξυπνου συμβολαίου που διαχειρίζεται τα tokens. Αυτό συμβαίνει, διότι οι παίχτες ανταλλάσσουν tokens μόνο με το συμβόλαιο αυτό και δεν πρέπει σε καμία περίπτωση κάποιος παίχτης να θέλει να πουλήσει τα tokens για να βγάλει κέρδος και να μην μπορεί. Οπότε, το συμβόλαιο αυτό πρέπει ανα πάσα στιγμή να έχει τη δυνατότητα να αγοράσει όλα τα tokens, τα οποία, πλέον, ανήκουν σε παίχτες. Άρα, στον εξωτερικό λογαριασμό της εταιρείας δίνεται η δυνατότητα να κάνει ανάληψη ενός ποσού ίσου με τα χρήματα τα οποία περισσεύουν στο συμβόλαιο, αν από τα συνολικά Ethers που κατέχει το συμβόλαιο αφαιρέσουμε τα Ethers που ισούνται με την αξία των tokens που ανήκουν σε παίχτες. Τέλος, όταν γίνεται ανάληψη χρημάτων από μια εταιρεία, ένα ποσοστό των Ethers πηγαίνει στον δημιουργό της αποκεντρωμένης εφαρμογής.

- Παίχτες
  Οι παίχτες ενός ηλεκτρονικού παιχνιδιού, αρχικά σπαταλούν Ethers, ώστε να αγοράσουν αντίστοιχα tokens του παιχνιδιού. Ανά πάσα στιγμή, μπορούν να πουλήσουν τα tokens που κατέχουν, για αξία ίδια με αυτή που τα αγόρασαν. Όταν αποφασίζουν να παίξουν ένα παιχνίδι με σκοπό να κερδίσουν παραπάνω tokens, πληρώνουν το αντίστοιχο ποσό tokens. Αν χάσουν, χάνουν τελείως το ποσό, ενώ αν κερδίσουν παίρνουν πίσω ένα μεγαλύτερο ποσό.

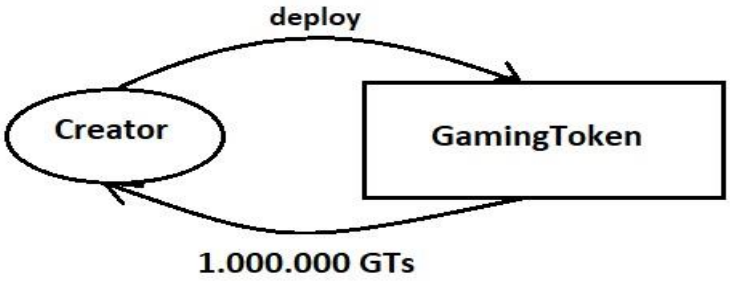
- Κάτοχος εφαρμογής
  Ο κάτοχος της αποκεντρωμένης εφαρμογής, κερδίζει Ethers, με τρεις τρόπους. Ο ένας τρόπος είναι κατά τη διάρκεια μιας ICO, όταν οι επενδυτές αγοράζουν tokens από αυτόν, οπότε και παίρνει τα Ethers ως αντάλλαγμα. Ο άλλος τρόπος είναι με πληρωμή σε μορφή μικρού φόρου, κάθε φορά που κάποια εταιρεία ηλεκτρονικών παιχνιδιών κάνει ανάληψη, δηλαδή μετατρέπει όσα tokens μπορεί και θέλει σε Ethers. Ο τελευταίος τρόπος είναι όταν μια εταιρεία θέλει, να προσθέσει παραπάνω tokens σε ένα υπάρχον παιχνίδι. Τότε, θα πρέπει να πληρώσει ένα μικρό ποσό για κάθε token, καθορισμένο από τον δημιουργό.

- Επενδυτές
  Τους επενδυτές αφορούν τα gaming tokens τα οποία είναι διαθέσιμα στην αγορά, δηλαδή αυτά που προκύπτουν, αν από τα συνολικά tokens, που έχουν δημιουργηθεί, αφαιρέσουμε αυτά, τα οποία κατέχει ο λογαριασμός του δημιουργού της εφαρμογής. Τα tokens αυτά ρέουν προς την αγορά, αρχικά κατά την ICO. Σκοπός των επενδυτών που αγοράζουν τα tokens είναι η αποκόμιση κέρδους. Αφότου ολοκληρωθεί κάποια ICO, οι επενδυτές που αγόρασαν tokens μπορούν να τα πουλήσουν μέσω διάφορων site συναλλάγματος tokens. Η τιμή των gaming tokens, όταν δεν υπάρχει ICO, καθορίζεται από την προσφορά και τη ζήτησή τους. Αν περισσότεροι πουλάνε, από όσους αγοράζουν, η τιμή πέφτει, ενώ αν περισσότεροι αγοράζουν, από όσους πουλάνε, η τιμή ανεβαίνει. Για να βγάλουν κέρδος, πρέπει να πουλήσουν τα tokens για περισσότερα Ethers, απ’ ό,τι τα αγόρασαν.

4.3 Η εφαρμογή βήμα βήμα

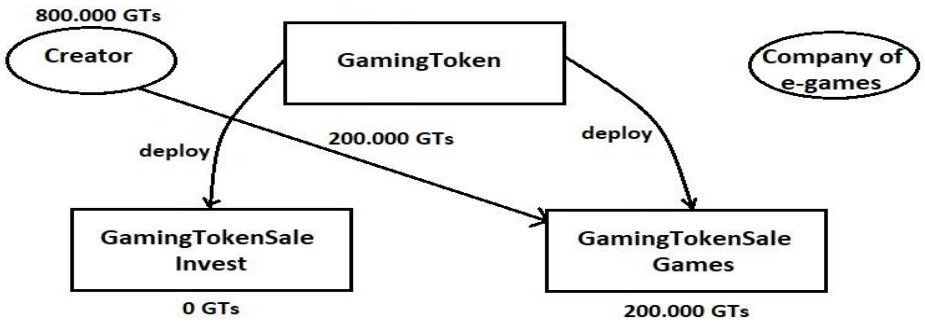
Για να γίνει περισσότερο κατανοητό πώς ακριβώς λειτουργεί η εφαρμογή, θα εξηγηθεί βήμα-βήμα, με τη βοήθεια ορισμένων σχημάτων. Τα τετράγωνα σχήματα αντιστοιχούν σε έξυπνα συμβόλαια, τα ελλειψοειδή αντιστοιχούν σε εξωτερικούς λογαριασμούς, ενώ οι ρόμβοι σε τελείως εξωτερικά του blockchain προγράμματα.

Αρχικά, όπως είπαμε δημιουργείται το έξυπνο συμβόλαιο του gaming token και ορίζεται η αρχική ποσότητα, έστω 1000000 GTs. Η ποσότητα αυτή προστίθεται στον εξωτερικό λογαριασμό του δημιουργού.



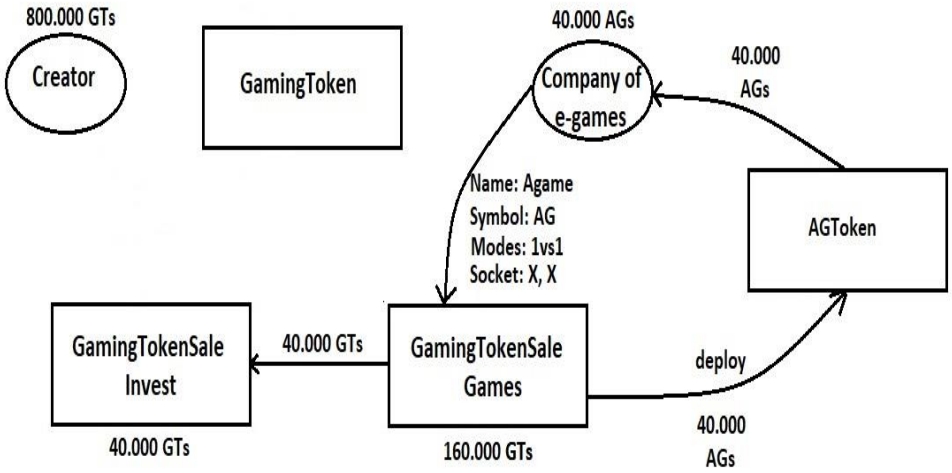
Σχήμα 4.1: Δημιουργία Συμβολαίου διαχείρισης των Gaming Tokens

Έπειτα, ο δημιουργός της εφαρμογής αποφασίζει πόσα GTs θα μπορούν να γίνουν διαθέσιμα προς το κοινό, έστω 200.000 GTs και ποια θα είναι η αξία του κάθε token σε Ether, έστω 0.001 Ether. Όμως, δεν τα δίνει με τη μία στην ICO του gaming token, αλλά προσφέρει ανάλογη ποσότητα στις εταιρείες ηλεκτρονικών παιχνιδιών, οι οποίες, όσο δημιουργούν δικά τους tokens, τόσο πυροδοτούν tokens προς την κανονική ICO. Δημιουργούνται, δηλαδή, άλλα δύο έξυπνα συμβόλαια από το βασικό συμβόλαιο, ένα για τις εταιρείες ηλεκτρονικών παιχνιδιών και ένα για τους επενδυτές. Εδώ, αξίζει να σημειωθεί ότι οι εταιρείες δεν μπορούν να δημιουργήσουν όσα tokens θέλουν, δωρεάν. Δηλαδή, από τις 200.000, ο δημιουργός καθορίζει πόσα αντιστοιχούν για κάθε παιχνίδι που εισάγεται, έστω 40.000. Αν, έπειτα, η εταιρεία θέλει να προσθέσει περισσότερα tokens σε ήδη υπάρχον παιχνίδι, θα πρέπει να πληρώσει ένα μικρό ποσό για κάθε token, το οποίο πηγαίνει στον δημιουργό.



Σχήμα 4.2: Δημιουργία των συμβολαίων της ICO

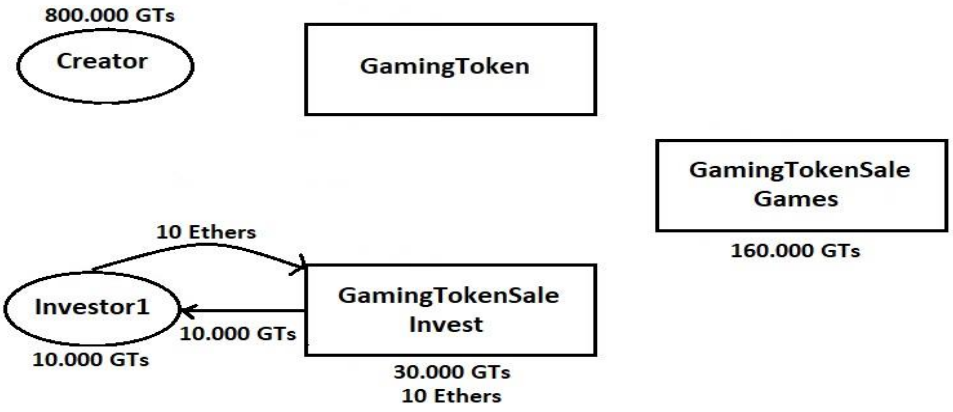
Στη συνέχεια, έστω ότι μία εταιρεία δημιουργεί δικό της token, το Agame Token ή AG. Τότε δημιουργείται ένα νέο συμβόλαιο από το συμβόλαιο GamingToken, το οποίο περιέχει όλες τις πληροφορίες για το Agame (όνομα παιχνιδιού, όνομα token, σύμβολο token, modes του παιχνιδιού, ip και port για το socket επικοινωνίας), καθώς και όλες τις συναρτήσεις που καθορίζουν τη λειτουργία του AG. Η αξία του κάθε AG είναι προκαθορισμένη και ίση με την αξία που έχει δοθεί σε κάθε GT, στην προκειμένη περίπτωση 0.001 Ether. Η συνολική ποσότητα πηγαίνει στον εξωτερικό λογαριασμό της εταιρείας, ο οποίος πυροδότησε τη συναλλαγή. Ταυτόχρονα, το συμβόλαιο που δημιουργεί τα tokens των παιχνιδιών, στέλνει όσα tokens δημιουργήθηκαν σε AGs, ως GTs, στο συμβόλαιο που πουλάει tokens στους επενδυτές.



Σχήμα 4.3: Είσοδος παιχνιδιού στην εφαρμογή

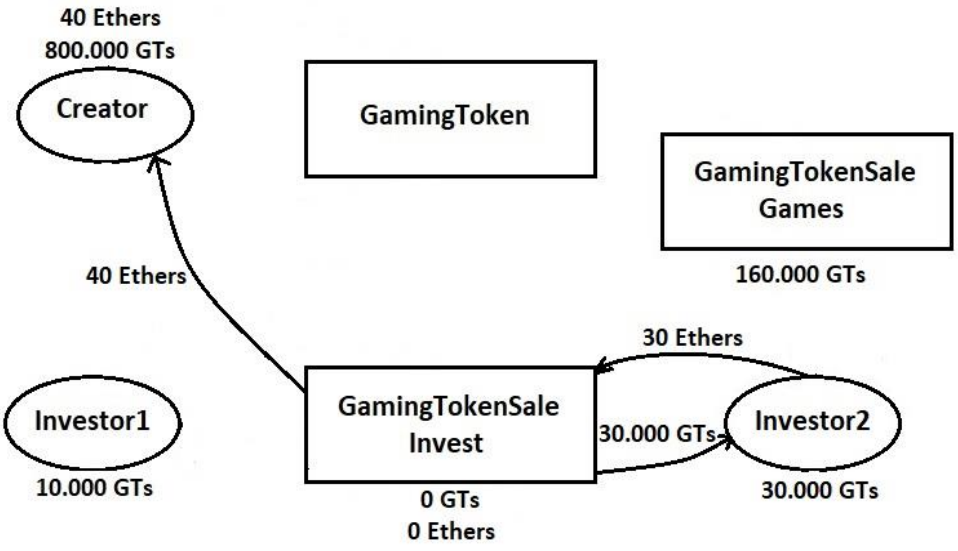
Έχουμε, λοιπόν, φτάσει στο σημείο, όπου οι διαδικασίες πώλησης έχουν πλήρως εξηγηθεί. Από αυτό το σημείο, μπορούμε να δούμε τις δύο διαδικασίες σαν δύο διαφορετικά πράγματα. Πάμε, δηλαδή, πρώτα να δούμε τη διαδικασία πώλησης στους επενδυτές και έπειτα τη διαδικασία, η οποία ακολουθείται, έως ότου παιχτεί κάποιο παιχνίδι για το Agame.

Αφότου υπάρχουν GTs προς πώληση για τους διάφορους επενδυτές, αυτοί μπορούν με αντίτιμο τα Ethers που αντιστοιχούν, να αγοράσουν κάποια GTs. Η τιμή των GTs έχει καθοριστεί στο παράδειγμά μας στα 0.001 Ether. Οπότε, έστω ότι κάποιος επενδυτής θέλει να αγοράσει 10.000 GTs. Τότε, θα πρέπει να τα αγοράσει έναντι 10.000 x 0.001 = 10 Ethers.



Σχήμα 4.4: Αγορά GTs από επενδυτές

Τα Ethers αυτά, αρχικά, αποθηκεύονται μέσα στο συμβόλαιο του Gaming Token, και κάθε φορά που τελειώνουν τα διαθέσιμα προς τους επενδυτές GTs, τα Ethers μεταφέρονται από το συμβόλαιο Gaming Token στον δημιουργό της εφαρμογής. Δηλαδή, έστω ότι ένας δεύτερος επενδυτής αγοράσει τα 30.000 GTs που απομένουν στο συμβόλαιο των επενδύσεων, αυτόματα τα χρήματα που είναι μαζεμένα στο συμβόλαιο αυτό, θα μεραφερθούν στον εξωτερικό λογαριασμό του δημιουργού.

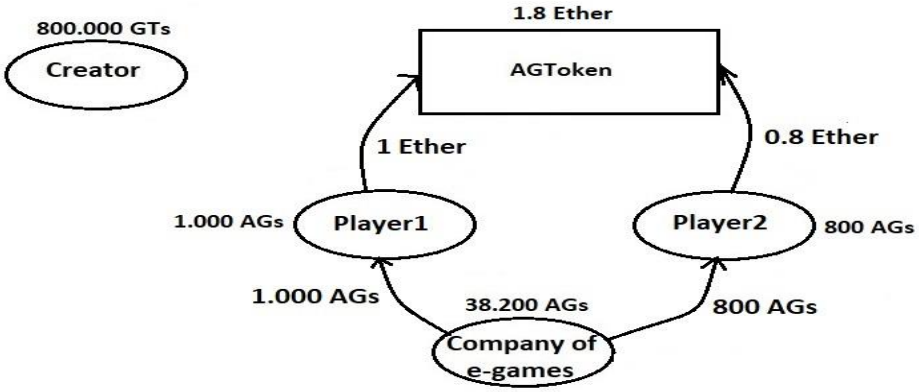


Σχήμα 4.5: Ολοκλήρωση μέρους της ICO και πληρωμή δημιουργού

Βέβαια, τα διαθέσιμα Ethers μπορούν να μεταφερθούν στον δημιουργό, αν αποφασίσει να σταματήσει την ICO. Τότε, τα εναπομείναντα Ethers και GTs πάνε αυτόματα πίσω στον δημιουργό και από τα δύο συμβόλαια πώλησης (GamingTokenSaleInvest και

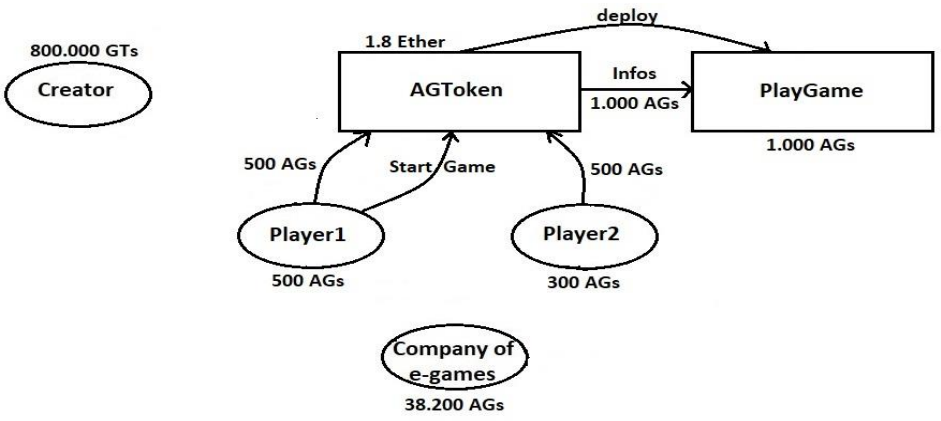
GamingTokenSaleGames). Από τη στιγμή που τελειώνει κάποια ICO, τα tokens που είναι στα χέρια των επενδυτών, συναλλάσσονται μεταξύ όλου του κοινού και, όπως είπαμε, η αξία καθορίζεται από την προσφορά και τη ζήτηση.

Πάμε, τώρα να δούμε τη διαδικασία που ακολουθείται για τα tokens του εκάστοτε παιχνιδιού, στο παράδειγμά μας, δηλαδή, για τα AGs. Τα AGs αρχικά βρίσκονται στην κατοχή του εξωτερικού λογαριασμού που έχει καθορίσει η εταιρεία του παιχνιδιού. Οι παίχτες, μέσω του συμβολαίου του παιχνιδιού, αγοράζουν AGs με αντίτιμο την αξία τους, στην προκειμένη περίπτωση με 0.001 Ether το καθένα. Έστω ότι δύο διαφορετικοί παίχτες, οι Player1 και Player2, αγοράζουν 1.000 AGs και 800 AGs δίνοντας 1 και 0.8 Ether, αντίστοιχα. Αυτόματα, το συμβόλαιο παίρνει τα tokens από τον λογαριασμό της εταιρείας και τα δίνει στους παίχτες.



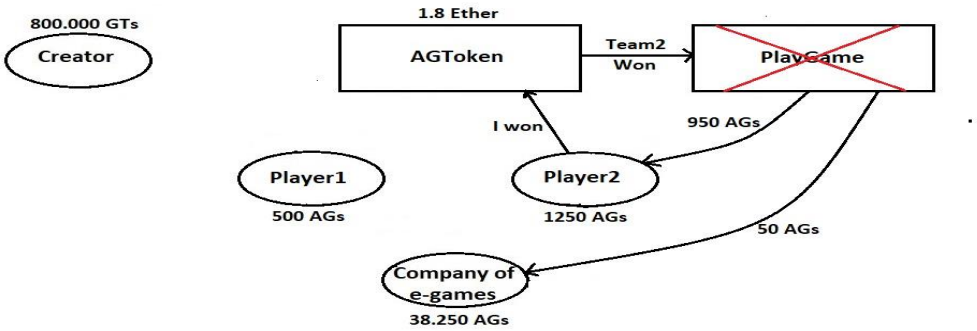
Σχήμα 4.6: Αγορά tokens παιχνιδιού από παίχτες

Στη συνέχεια, έστω ότι ο κάθε παίχτης αποφασίζει να παίξει το παιχνίδι Agame στο mode 1vs1, ρισκάροντας 500 AGs έκαστος. Αφού πραγματοποιηθεί το ματσάρισμα, επιλέγεται ένας από τους δύο παίχτες, έστω ο Player1 (αφού και οι δύο ορίζονται ως αρχηγοί για τις ομάδες τους), ώστε να εκτελέσει μια συναλλαγή για να αρχίσει το παιχνίδι. Συγκεκριμένα, ο παίχτης στέλνει ένα καθορισμένο από την εφαρμογή μήνυμα, το οποίο περιέχει τις διάφορες πληροφορίες για το παιχνίδι, καθώς και δίνει εντολή για τη δημιουργία ενός νέου συμβολαίου που αφορά μόνο το συγκεκριμένο παιχνίδι, μεταξύ των δύο παιχτών. Τα tokens αποθηκεύονται σε αυτό το συμβόλαιο, το οποίο απλά έπειτα περιμένει να λάβει έναν έγκυρο νικητή.



Σχήμα 4.7: Δημιουργία συμβολαίου για τον αγώνα ενός παιχνιδιού

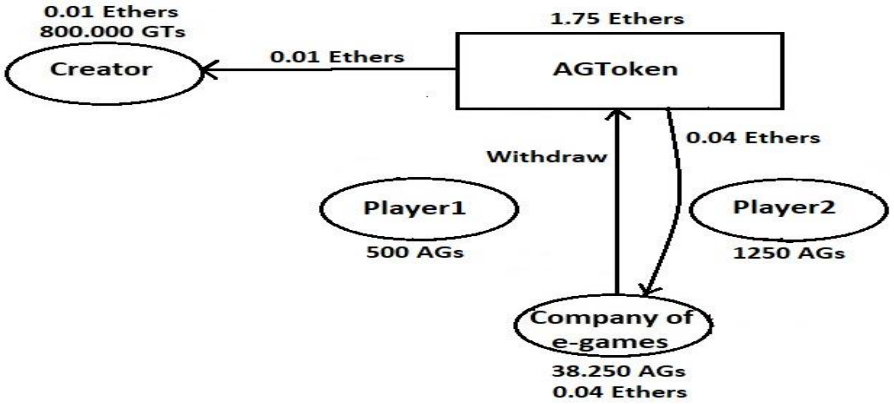
Όταν το παιχνίδι τελειώσει, στέλνεται στο νικητή ένα καθορισμένο μήνυμα-συναλλαγή. Ο νικητής την εκτελεί και παίρνει πίσω το ποσό που του αναλογεί σε AGs, στέλνονται κάποια AGs ως φόρος στον λογαριασμό της εταιρείας και οι μικροφόροι που οι παίχτες πλήρωσαν σε Ethers για να κάνουν τις διάφορες συναλλαγές με τα συμβόλαια, τους επιστρέφονται. Όταν ολοκληρωθεί η όλη διαδικασία, το συμβόλαιο PlayGame, που έχει δημιουργηθεί για την αναπαράσταση του συγκεκριμένου παιχνιδιού, καταστρέφεται. Οι φόροι, όπως και η επιστροφή τους, δεν αναγράφονται στα σχήματα, για λόγους απλότητας. Οπότε, έστω ότι νικάει ο Player2 και ότι ο φόρος που παίρνει η εταιρεία είναι 5%, άρα και από τα 1.000 AGs που παίχτηκαν τα 50.



Σχήμα 4.8: Διαδικασία ολοκλήρωσης ενός αγώνα

Το πώς ακριβώς διασφαλίζεται ότι κάποιος παίχτης δεν κλέβει, θα εξηγηθεί μαζί με την λειτουργία των συμβολαίων.

Τελευταίο βήμα αποτελεί η ανάληψη χρημάτων από την εταιρεία του παιχνιδιού. Η εταιρεία, ανά πάσα στιγμή μπορεί να πάρει Ethers από το συμβόλαιο AGToken, όμως, μόνο όσα μένουν, αν από τα συνολικά αφαιρεθούν αυτά τα οποία είναι ικανά να αγοράσουν όλα τα AGs που κυκλοφορούν στους παίχτες. Δηλαδή, στο παράδειγμά μας, οι παίχτες έχουν 1750 AGs σύνολο, και το συμβόλαιο έχει 1.8 Ethers. Τα 1750 AGs αντιστοιχούν σε 1.75 Ethers. Οπότε, η εταιρεία μπορεί να αναλήψει τα 0.05 Ethers. Από τα 0.05 Ethers που θα μεταφερθούν στον εξωτερικό λογαριασμό της εταιρείας, ένα ποσό, έστω το 20%, δηλαδή 0.01 Ethers μεταφέρονται ως φόρος στον δημιουργό της εφαρμογής.



Σχήμα 4.9: Ανάληψη ποσού από την εταιρεία ενός παιχνιδιού

Αυτή είναι γενικά και όλη η διαδικασία που ακολουθείται από την αρχή με τις ICO μέχρι το τέλος, με την ανάληψη από την εταιρεία.

4.4 Έξυπνα Συμβόλαια Εφαρμογής και Ακριβής Λειτουργία

4.4.1 Έξυπνα Συμβόλαια της εφαρμογής

Τα έξυπνα συμβόλαια που χρησιμοποιούνται για τη λειτουργία της εφαρμογής είναι πέντε. Τα δύο, όμως, εξ' αυτών χρησιμοποιούνται πολλαπλές φορές, δηλαδή δημιουργούνται συνεχώς νέα ίδια συμβόλαια (ο κώδικάς τους είναι ίδιος, όχι οι διευθύνσεις τους) για τις ανάγκες της εφαρμογής.

Το βασικό συμβόλαιο της εφαρμογής είναι το GamingToken (86 lines), δηλαδή το συμβόλαιο, όπου ορίζεται η λειτουργία του gaming token. Το token αυτό ακολουθεί το πρωτόκολλο ERC-20, το οποίο εξηγήθηκε πλήρως σε προηγούμενη ενότητα, αλλά περιλαμβάνει και ορισμένες επιπρόσθετες λειτουργίες-συναρτήσεις. Συγκεκριμένα, περιέχει μια συνάρτηση tokenFallback, η οποία χρησιμεύει στο να αποφεύγονται τα λάθη με τα χαμένα tokens. Η συνάρτηση αυτή ελέγχει αν ο λογαριασμός στον οποίο μεταφέρονται τα tokens σε μια συναλλαγή έχει ή όχι δεδομένα. Αν δεν έχει δεδομένα, σημαίνει ότι είναι εξωτερικός λογαριασμός, οπότε η συναλλαγή προχωράει, ενώ σε αντίθετη περίπτωση σημαίνει ότι αφορά λογαριασμό συμβολαίου, οπότε και ακυρώνει τη συναλλαγή. Περιέχει, επιπλέον, μια συνάρτηση η οποία ξεκινά την ICO με συγκεκριμένη ποσότητα tokens. Τέλος, περιέχει δύο συναρτήσεις, οι οποίες είναι αντίθετες μεταξύ τους. Μέσω της μίας, ο δημιουργός έχει τη δυνατότητα να προσθέσει ποσότητα tokens στη συνολική ποσότητα που ήδη υπάρχει, ενώ μέσω της άλλης, δίνεται η δυνατότητα καταστροφής ορισμένης ποσότητας tokens, από την ήδη υπάρχουσα ποσότητα.

Επόμενο συμβόλαιο είναι το GamingTokenSaleGame (120 lines), το οποίο αφορά τη δημιουργία των tokens για τα διάφορα παιχνίδια. Υπάρχει συνάρτηση μέσω της οποίας μια εταιρεία ηλεκτρονικών παιχνιδιών δημιουργεί ένα νέο token για το παιχνίδι της, με αρχική ποσότητα tokens όσα έχουν οριστεί από τον δημιουργό της εφαρμογής. Δημιουργείται και ένα νέο συμβόλαιο για κάθε νέο παιχνίδι, το οποίο ονομάζεται RandomToken και θα εξηγηθεί πλήρως. Επίσης, περιέχει συνάρτηση μέσω της οποίας μια εταιρεία μπορεί να αγοράσει όσα επιπλέον tokens θέλει, για το ήδη υπάρχον παιχνίδι της. Και στις δύο παραπάνω περιπτώσεις, στέλνεται ίδια ποσότητα gaming tokens για την ICO του gaming token, με αυτή που η εταιρεία δημιούργησε για το παιχνίδι της. Τέλος, υπάρχει συνάρτηση μέσω της οποίας σταματάει η ICO από τον δημιουργό.

Η κανονική ICO του gaming token σε επενδυτές καθορίζεται από το συμβόλαιο GamingTokenSaleInvest (49 lines). Περιέχει μία συνάρτηση, η οποία είναι υπεύθυνη για την πώληση των διαθέσιμων gaming tokens σε επενδυτές, καθώς και μία συνάρτηση μέσω της οποίας σταματάει η πώληση και τα εναπομείναντα tokens επιστρέφουν στην κατοχή του δημιουργού της εφαρμογής.

Κάθε φορά που ένα νέο παιχνίδι γίνεται μέρος της εφαρμογής, δημιουργείται και ένα νέο συμβόλαιο για το token του παιχνιδιού, το RandomToken (158 lines), όπως προειπώθηκε. Μπορούν να δημιουργηθούν φαινομενικά άπειρα τέτοια συμβόλαια, ένα για κάθε νέο παιχνίδι. Το συμβόλαιο αυτό, αν και αποτελεί συμβόλαιο token, δεν ακολουθεί κανένα

πρωτόκολλο. Αυτό συμβαίνει, διότι το token που αφορά το συμβόλαιο δεν βγαίνει στις αγορές, αλλά χρησιμοποιείται μόνο για τους παίχτες του εκάστοτε παιχνιδιού. Περιέχει συναρτήσεις, μέσω των οποίων οι παίχτες μπορούν να αγοράσουν από το συμβόλαιο και να πουλήσουν στο συμβόλαιο tokens, καθώς και συνάρτηση μέσω της οποίας ένας παίχτης μπορεί να προσφέρει tokens σε κάποιον άλλο εξωτερικό λογαριασμό. Επιπλέον, περιέχει συνάρτηση μέσω της οποίας πραγματοποιείται η ανάληψη των διαθέσιμων Ethers από τον λογαριασμό που η εταιρεία έχει θέσει ως κάτοχο του token. Τέλος, υπάρχουν συναρτήσεις, οι οποίες διαχειρίζονται τα παιχνίδια, μία για τη δημιουργία ενός νέου συμβολαίου για κάθε παιχνίδι, του PlayGame, και μία για τον καθορισμό του νικητή του κάθε παιχνιδιού, η οποία στέλνει τα δεδομένα στο PlayGame για να πληρωθούν οι νικητές.

Όπως αναφέρθηκε, ένα νέο συμβόλαιο PlayGame (94 lines) δημιουργείται κάθε φορά που ένας αγώνας ξεκινάει. Το PlayGame περιέχει μία συνάρτηση για την αποκόμιση των κεφαλαίων από τους παίχτες, που λαμβάνουν μέρος στον εκάστοτε αγώνα και μία συνάρτηση, που πληρώνει τους νικητές του αγώνα. Όταν το συμβόλαιο αυτό εκτελέσει όλες του τις λειτουργίες, αυτοκαταστρέφεται. Το συμβόλαιο, όμως, αυτό περιλαμβάνει και ορισμένα τεχνάσματα, μέσω των οποίων επιτυγχάνεται η ασφάλεια της εφαρμογής, τα οποία και θα μελετήσουμε ευθύς αμέσως.

4.4.2 Λεπτομέρειες Λειτουργίας

Για το frontend της εφαρμογής έχει δημιουργηθεί μια διαδικτυακή πλατφόρμα, η οποία επικοινωνεί με τα έξυπνα συμβόλαια και μέσω της οποίας όλοι οι συμμετέχοντες μπορούν να αλληλεπιδράσουν με τα συμβόλαια αυτά. Περιλαμβάνεται και ένας server, ο οποίος επικοινωνεί με τους servers των παιχνιδιών για να στείλει τις απαραίτητες πληροφορίες στους παίκτες. Δεν υπάρχει κανονική βάση δεδομένων, καθώς για τον σκοπό αυτής χρησιμοποιείται το blockchain.

Ας πάρουμε, όμως, κάθε κομμάτι ξεχωριστά, ώστε να γίνει ξεκάθαρη η λειτουργία της διαδικτυακής πλατφόρμας. Αποτελείται από πέντε διαφορετικές html σελίδες, οι οποίες έχουν ένα βασικό css κομμάτι και πολλές λειτουργίες μέσω Javascript (κυρίως jQuery και web3). Όλες οι συναλλαγές πραγματοποιούνται με τη βοήθεια της Metamask.

Η πρώτη σελίδα είναι η σελίδα, που προσφέρει στις εταιρείες ηλεκτρονικών παιχνιδιών τη δυνατότητα να εισάγουν ένα παιχνίδι τους στην εφαρμογή, ώστε να γίνει μέρος της ή να πληρώσουν ώστε να προσθέσουν περισσότερα tokens σε ένα ήδη υπάρχον παιχνίδι. Η σελίδα αυτή αποτελείται από κώδικα html (gametokens.html, 583 lines), κώδικα css (gametokens.css, 44 lines), κώδικα Javascript (game.js, 112 lines), καθώς και πολλά jQuery scripts, εντός του κώδικα html.

Στο πρώτο κομμάτι, η σελίδα ζητάει από την εταιρεία να εισάγει το όνομα του παιχνιδιού, το σύμβολο και τα modes (πώς θα παίζεται το παιχνίδι, π.χ με δύο ομάδες και 5 παίχτες ανά ομάδα). Αφότου συμπληρωθεί η φόρμα, η πλατφόρμα κατευθύνει τον χρήστη της εταιρείας να εκτελέσει μια συνάρτηση του συμβολαίου, δίνοντας ένα μικρό ποσό σε Ether για το gas που θα καταναλωθεί, μέσω της Metamask. Όταν η συναλλαγή ολοκληρωθεί, του επιστρέφει τον λογαριασμό συμβολαίου που αφορά το συγκεκριμένο παιχνίδι.

Στο δεύτερο κομμάτι, όπου η εταιρεία ηλεκτρονικών παιχνιδιών μπορεί να αγοράσει περισσότερα tokens για το ήδη υπάρχον παιχνίδι της, η πλατφόρμα ζητά τη διεύθυνση του συμβολαίου και ο χρήστης μόλις την πληκτρολογήσει και υποβάλλει τη φόρμα, πρέπει να προσφέρει εκτός του ποσού για το gas και ένα ποσό ανάλογο με το πόσα tokens θέλει. Αν ο χρήστης δεν είναι ο κάτοχος του παιχνιδιού η συναλλαγή ακυρώνεται.

Η δεύτερη σελίδα αφορά την ICO του Gaming Token, προς τους διάφορους επενδυτές. Η ποσότητα των GTs, που προσφέρεται στην ICO είναι ίση με αυτή των tokens, που έχουν δημιουργηθεί για όσα παιχνίδια ανήκουν στην πλατφόρμα, έτσι ώστε οι επενδυτές να έχουν λόγο να επενδύσουν. Αποτελείται από κώδικα html (investors.html, 87 lines), κώδικα css (investors.css, 44 lines) και κώδικα Javascript (investors.js, 128 lines).

Η σελίδα εμφανίζει στους επενδυτές την τιμή των Gaming Tokens και πόσα είναι διαθέσιμα για αγορά. Αφότου ο εκάστοτε επενδυτής συμπληρώσει τη φόρμα, η σελίδα του εμφανίζει ένα αναδυόμενο παράθυρο της Metamask, όπου πρέπει να προσφέρει Ethers για το gas που θα χρειαστεί για την εκτέλεση του συμβολαίου, καθώς και για την αγορά των tokens.

Η τρίτη σελίδα αφορά τη διαχείριση των tokens των παιχνιδιών από τους κατόχους τους, δηλαδή τους λογαριασμούς, μέσω των οποίων η εταιρεία κάθε παιχνιδιού εκτέλεσε τη δημιουργία του εκάστοτε παιχνιδιού. Αποτελείται από κώδικα html (owner.html, 68 lines), κώδικα css (owner.css, 44 lines) και κώδικα Javascript (owner.js, 1035 lines).

Μέσω αυτής της σελίδας, ο κάτοχος του παιχνιδιού μπορεί να κάνει ανάληψη των tokens. Η σελίδα τον ενημερώνει, πόσα Ethers μπορεί να κάνει ανάληψη και του ζητάει να συμπληρώσει το ποσό που θέλει. Όταν η φόρμα υποβληθεί, προσφέρεται σε αυτόν τον λογαριασμό το μεγαλύτερο ποσοστό αυτών των Ethers, ενώ το υπόλοιπο ποσοστό πηγαίνει στον δημιουργό της εφαρμογής.

Η τέταρτη σελίδα αφορά τη διαχείριση των tokens των παιχνιδιών από τους παίχτες. Αποτελείται από κώδικα html (gamer.html, 94 lines) , κώδικα css (gamer.css, 44 lines) και κώδικα Javascript (gamer.js, 1110 lines).

Μέσω αυτής της σελίδας, οι παίχτες έχουν τη δυνατότητα να επιλέξουν οποιοδήποτε από τα παιχνίδια που είναι μέρος της πλατφόρμας. Αφού επιλέξουν κάποιο παιχνίδι από τη λίστα που τους δίνεται, έχουν τη δυνατότητα να αγοράσουν, να πουλήσουν ή να προσφέρουν σε κάποιον άλλο λογαριασμό tokens του εκάστοτε παιχνιδιού. Στις δύο πρώτες περιπτώσεις, η σελίδα τους αναφέρει πόσα tokens είναι στην κατοχή τους, την τιμή αγοράς και πώλησης των tokens, καθώς και πόσα tokens είναι διαθέσιμα για αγορά, ενώ υπάρχει μία φόρμα, στην οποία συμπληρώνουν την ποσότητα των tokens που θέλουν να αγοράσουν ή να πουλήσουν. Στην τρίτη περίπτωση (προσφορά), η σελίδα τους παρέχει μια φόρμα, στην οποία συμπληρώνουν τη διεύθυνση του εξωτερικού λογαριασμού, στον οποίο θέλουν να προσφέρουν tokens, καθώς και την ποσότητα των tokens, που θέλουν να προσφέρουν. Εδώ, να αναφερθεί, ότι αν κάποιος παίχτης επιλέξει να προσφέρει tokens στον λογαριασμό, που είναι ιδιοκτήτης του παιχνιδιού, τότε προστίθεται στο ποσό των Ethers, που ο λογαριασμός αυτός έχει δυνατότητα να κάνει ανάληψη και μια αξία, ίση με αυτή των tokens, που προσφέρθηκαν. Αυτό συμβαίνει,

διότι, μέρος της ποσότητας των tokens που άνηκαν στους παίχτες δόθηκε στον κάτοχο του παιχνιδιού χωρίς αντάλλαγμα, οπότε το συμβόλαιο του παιχνιδιού, χρειάζεται να έχει τόσα λιγότερα Ethers, για να μπορεί να αγοράσει όλα τα tokens των παιχτών, όση και η αξία των tokens, που προσφέρθηκαν.

Η πέμπτη σελίδα αφορά την εύρεση παιχνιδιού από τους παίκτες. Αποτελείται από κώδικα html (play.html, 72 lines) , κώδικα css (play.css, 44 lines), κώδικα Javascript (play.js, 1011 lines), κώδικα Javascript, που επικοινωνεί με τον server της εφαρμογής (socketclient.js, 1940 lines). Εδώ, να συμπληρωθεί, ότι έχει δημιουργηθεί server για την εφαρμογή (socketserver.js, 125 lines), καθώς και server του εκάστοτε παιχνιδιού (gameserver.js, 22 lines), ώστε να προσομοιωθεί όσο το δυνατόν καλύτερα η real-time εκτέλεσή της.

Μέσω αυτής της σελίδας, οι παίχτες έχουν τη δυνατότητα να επιλέξουν οποιοδήποτε από τα παιχνίδια που είναι μέρος της πλατφόρμας. Αφού επιλέξουν κάποιο παιχνίδι από τη λίστα που τους δίνεται, έχουν τη δυνατότητα να επιλέξουν ποιο mode θέλουν να παίξουν (πόσες ομάδες και πόσοι παίχτες ανά ομάδα), καθώς και ένα εκ τριών ποσοτήτων (100, 500, 1000) tokens για να παίξουν. Η σελίδα επικοινωνεί με τον server της εφαρμογής, στέλνοντάς του όλα τα στοιχεία (παιχνίδι, mode, ποσότητα tokens) και ο server, περιμένει έως ότου μαζευτούν οι παίχτες που χρειάζονται για να συμπληρωθούν ομάδες. Ταυτόχρονα, ο server διαλέγει για κάθε ομάδα έναν αρχηγό. Όταν ο server βρει παιχνίδι, στέλνει σε έναν εκ των αρχηγών μήνυμα, ώστε να εκτελέσει μια συναλλαγή, που θα επικοινωνήσει με το blockchain και θα πάρει την αντίστοιχη ποσότητα tokens από κάθε παίχτη. Όταν η διαδικασία αυτή ολοκληρωθεί, στέλνονται σε κάθε παίχτη οι πληροφορίες του παιχνιδιού. Τέλος, όταν κάποια ομάδα νικήσει, στέλνεται μήνυμα στον αρχηγό της ομάδας, μέσω του οποίου του ζητείται να εκτελέσει μια συναλλαγή, ώστε οι παίχτες της ομάδας του, να πληρωθούν για τη νίκη τους.

Όλη η εφαρμογή χρησιμοποιεί για την ασφάλεια όλων των συναλλαγών της το Ethereum blockchain, εκτός από ένα σημείο. Το σημείο αυτό είναι όταν ένας αγώνας πραγματοποιείται. Υπήρχαν δύο επιλογές για να επιτευχθεί η ασφάλεια, ή να εκτελούνται όλες οι συναλλαγές σε έναν αγώνα από τον εξωτερικό λογαριασμό που έχει τεθεί ως ιδιοκτήτης του token από την εταιρεία ηλεκτρονικών παιχνιδιών, ή να εκτελούνται απευθείας από τους παίχτες. Στην πρώτη περίπτωση, θα υπήρχε πλήρης ασφάλεια, καθώς ο έλεγχος της ορθότητας θα πραγματοποιούνταν απευθείας από την εταιρεία, όμως θα έπρεπε η εταιρεία να έχει ένα άτομο συνεχώς μπροστά από έναν υπολογιστή, μέσω του οποίου θα έλεγχε κάθε συναλλαγή ξεχωριστά. Στη δεύτερη περίπτωση, θα πραγματοποιούνταν όλα γρηγορότερα και χωρίς τη χρήση ενός τέτοιου ατόμου, όμως θα έπρεπε να βρεθούν τεχνάσματα μέσω των οποίων να εξαλείφεται η όποια νοθεία από τους παίχτες.

Στα πλαίσια της συγκεκριμένης εφαρμογής, χρησιμοποιήθηκε η δεύτερη επιλογή. Οι παίχτες από μόνοι τους εκτελούν όλες τις συναλλαγές ενός παιχνιδιού. Το τέχνασμα που καθορίστηκε για να μην υπάρξουν νοθείες έχει ως εξής:

I. Επιλέγεται από κάθε ομάδα ένας παίχτης ως αρχηγός της.

II. Για όλους τους αρχηγούς, ο server της εφαρμογής μετατρέπει τους λογαριασμούς τους σε δυαδική μορφή, γεννά έναν τυχαίο δυαδικό αριθμό, με ίσα ψηφία με τους

λογαριασμούς και εκτελεί την πράξη xor μεταξύ τους. Έπειτα, παίρνει το αποτέλεσμα και το κρυπτογραφεί με χρήση του αλγορίθμου του Keccak-256.

III. Στέλνονται όλες οι πληροφορίες λογαριασμών και όλοι οι τυχαίοι αριθμοί ως εκτελέσιμο μήνυμα σε έναν εκ των αρχηγών, ο οποίος τίθεται αρμόδιος να επικοινωνήσει με τα έξυπνα συμβόλαια. Ο αρχηγός αυτός δεν μπορεί να δει το μήνυμα, αλλά μόνο να το εκτελέσει. Οι τυχαίοι αριθμοί στέλνονται από τον server της εφαρμογής, στον server του εκάστοτε παιχνιδιού, ενώ οι λογαριασμοί των παιχτών δεν κρατώνται αυτούσιοι από τον server της εφαρμογής, αλλά ούτε κρυπτογραφημένοι. Ο server της εφαρμογής περιέχει μόνο τις πληροφορίες επικοινωνίας (sockets) με κάθε αρχηγό ομάδας. Οπότε, μόλις ο επιλεγμένος αρχηγός εκτελέσει το μήνυμα και το στάδιο αυτό τελειώσει, έχουν παρθεί όλα τα ποσά από τους παίχτες, το συμβόλαιο PlayGame έχει κάνει την ίδια διαδικασία κρυπτογράφησης, άρα περιέχει μόνο κάποιους κρυπτογραφημένους λογαριασμούς, έναν για τον αρχηγό κάθε ομάδας, ο server της εφαρμογής δεν περιέχει καμία πληροφορία για τους λογαριασμούς των παιχτών, ενώ ο server του παιχνιδιού περιέχει μόνο τους τυχαίους αριθμούς, αντιστοιχισμένους, όμως με τις ομάδες, τις οποίες αφορούν.

IV. Όταν ο αγώνας τελειώσει και βρεθεί ο νικητής, ο server του παιχνιδιού, στέλνει στον server της εφαρμογής τον αριθμό της ομάδας και τον τυχαίο αριθμό που αντιστοιχεί στην ομάδα αυτή. Ο server της εφαρμογής στέλνει στον αρχηγό της νικήτριας ομάδας ένα μήνυμα που περιέχει τα δύο αυτά δεδομένα, ώστε να τα στείλει ως πληροφορίες κατά την εκτέλεση του συμβολαίου. Ο αρχηγός αυτός εκτελεί το συμβόλαιο και, αν η κρυπτογράφηση του λογαριασμού που εκτελεί το συμβόλαιο (δηλαδή σε κανονικές συνθήκες του λογαριασμού του αρχηγού των νικητών) ταυτίζεται με την αποθηκευμένη κρυπτογράφηση του PlayGame, τα ποσά μοιράζονται στους παίχτες της νικήτριας ομάδας. Πριν το συμβόλαιο καταστραφεί, επιστρέφει τα μικροποσά που χρησιμοποίησαν οι παίχτες ως φόρους για τις διάφορες εκτελέσεις πίσω σε αυτούς.

Στο υπόλοιπο frontend της εφαρμογής, όλες οι συναλλαγές εκτελούνται από τους εξωτερικούς λογαριασμούς των ατόμων που θέλουν να τις πυροδοτήσουν. Σε πολλούς κώδικες του frontend, που οι γραμμές του κώδικα φαίνονται υπερβολικές, σε σχέση με τις διαδικασίες που επιτελούν, είναι επειδή η αναφορά στα smart contracts, γίνεται μέσω ενός αρχείου, που περιέχει όλες τις πληροφορίες του συμβολαίου (ABI) και της διεύθυνσης του συμβολαίου (address). Το ABI αποτελείται από πολλές γραμμές κώδικα, οπότε και για αυτόν τον λόγο οι κώδικες είναι πολύ μεγάλοι. Στο σημείο αυτό έχουν εξηγηθεί πλήρως όλες οι πτυχές της αποκεντρωμένης αυτής εφαρμογής.

Ο κώδικας της εφαρμογής διατίθεται στο παρακάτω link:
https://github.com/XaleAth/GamingToken

5 Συμπεράσματα

5.1 Συνεισφορά

Η συνεισφορά της εργασίας και τα συμπεράσματα που προέκυψαν είναι τα παρακάτω:

- Υλοποιήθηκε μια αποκεντρωμένη εφαρμογή (Dapp), η οποία δίνει τη δυνατότητα σε εταιρείες ηλεκτρονικών παιχνιδιών να χρησιμοποιήσουν την ασφάλεια και την ταχύτητα που προσφέρει το blockchain στους χρήστες του, ώστε να δώσουν την ευκαιρία στους παίχτες των παιχνιδιών τους να ανταγωνίζονται για πραγματικά χρήματα.

- Υλοποιήθηκε όλη η αυτοματοποιημένη διαδικάσια για τη σωστή χρήση της εφαρμογής, μέσω έξυπνων συμβολαίων, καθώς και όλη η διαδικτυακή πλατφόρμα, μέσω της οποίας τίθεται σε ισχύ η επικοινωνία με τα έξυπνα αυτά συμβόλαια.

- Παρατηρήθηκε η δυσκολία ομαλής λειτουργίας μιας τέτοιας αποκεντρωμένης εφαρμογής, καθώς αν υπάρξει έστω ένα λάθος σε κάποιο έξυπνο συμβόλαιο, όλη η εφαρμογή πρέπει να δημιουργηθεί εκ νέου και όλα τα ποσά που έχουν δοθεί από τους χρήστες της εφαρμογής μπορεί να βρεθούν σε κίνδυνο.

5.2 Μελλοντική έρευνα

Η βασική κατεύθυνση έρευνας, η οποία διαφαίνεται από το τελευταίο συμπέρασμα αφορά την παροχή αισθήματος ασφάλειας σε όλους τους χρήστες της εφαρμογής. Οι χρήστες είναι απαραίτητο να νιώθουν σίγουροι ότι μπορούν να χρησιμοποιήσουν τα χρήματά τους για τη χρήση της εφαρμογής χωρίς τον φόβο αναπάντεχης απώλειας. Άρα πρέπει τα συμβόλαια αυτά να υποβληθούν σε όλους τους ελέγχους για επιθέσεις και να τροποποιηθούν αναλόγως, πριν αποφασιστεί η έξοδος της εφαρμογής αυτής στην πραγματική αγορά.

Επιπλέον, η εφαρμογή αυτή έχει ένα μειονέκτημα. Η αξία του Ether δεν είναι σταθεροποιημένη, αλλά αλλάζει συνεχώς. Η αξία των tokens της εφαρμογής δεν αντιστοιχεί σε πραγματικά απτά χρήματα (π.χ σε USD), αλλά σε Ethers, δηλαδή στα ψηφιακά αυτά νομίσματα του Ethereum. Οπότε, η αξία των tokens σε πραγματικά χρήματα, μπορεί να έχει αναπάντεχες αλλαγές. Ένας επενδυτής μπορεί να παρατηρήσει μείωση στα κεφάλαια του, ακόμα και σε περιόδους που η αξία ενός GT είναι περισσότερα Ethers, λόγω πτώσης της αξίας του Ether ή και το αντίθετο. Για έναν παίχτη αυτό είναι ακόμα χειρότερο, καθώς θέλει να γνωρίζει το ποσό που έχει στην κατοχή του. Για το λόγο αυτό, θα μπορούσε να βρεθεί ένας τρόπος, ώστε η αξία των tokens της εφαρμογής να αντιστοιχίζεται σε κάποιο πραγματικό νόμισμα. Αυτό έχει ήδη πραγματοποιηθεί για ένα token που ανήκει στην πλατφόρμα του Ethereum, οπότε είναι εφικτό.

Επιπρόσθετα, θα μπορούσαν να πραγματοποιηθούν ορισμένες αλλαγές και προσθήκες στην εφαρμογή, ώστε να υπάρχει μεγαλύτερη ασφάλεια στη χρήση της. Σίγουρα, θα πρέπει οι παίχτες, οι οποίοι ανταγωνίζονται μεταξύ τους, να μην επιλέγονται μόνο βάσει του ποσού που επιλέγουν να ρισκάρουν. Σε όλους τους ανταγωνιστικούς αγώνες, υπάρχει η τιμή της επιδεξιότητας για κάθε παίχτη. Αν παίχτες διαφορετικών επιδεξιοτήτων

ανταγωνίζονται μεταξύ τους, ο αγώνας είναι άνισος. Οπότε, θα μπορούσε να προστεθεί η πληροφορία αυτή στην εύρεση αγώνα. Επιπλέον, για να είναι σίγουρο ότι ένας παίχτης με μεγαλύτερη επιδεξιότητα δε χρησιμοποιεί τον λογαριασμό κάποιου άλλου, ώστε να παίζει με παίχτες μικρότερης επιδεξιότητας, για να βγάζει εύκολα χρήματα, θα μπορούσε να γίνεται χρήση της κάμερας του υπολογιστή του κατά τη διάρκεια του παιχνιδιού, ανά τακτά χρονικά διαστήματα. Οι εικόνες αυτές να συγκρίνονται με αντίστοιχες εικόνες κάθε παίχτη, μέσω ενός προγράμματος και, αν παρατηρηθεί μικρότερη ομοιότητα από κάποια προκαθορισμένη από το πρόγραμμα, ο παίχτης αυτός να χάνει όλα του τα κεφάλαια και ο λογαριασμός του να κλειδώνεται.

Τέλος, αξίζει να σημειωθεί ότι η πλατφόρμα του Ethereum είναι σε διαδικασία αλλαγής. Πλησιάζει ο ερχομός του Ethereum 2, το οποίο θα προσδίδει πολλές περισσότερες δυνατότητες στα έξυπνα συμβόλαια, ενώ θα αλλάξει και τελείως ο αλγόριθμος απόδειξης, ο οποίος θα μετατραπεί από proof-of-work σε proof-of-stake. Οπότε, όλες οι αποκεντρωμένες εφαρμογές, που είναι χτισμένες στην πλατφόρμα του Ethereum, θα έχουν νέες δυνατότητες, άρα και η συγκεκριμένη εφαρμογή, θα μπορεί να βελτιωθεί με τρόπους άγνωστους τη συγκεκριμένη χρονική στιγμή.

Βιβλιογραφία

1. **Ameer Rosic.** What is Blockchain Technology? A Step-by-Step Guide For Beginners. *Blockgeeks.* https://blockgeeks.com/guides/what-is-blockchain-technology/. 25th June 2019.
2. —. History of Blockchain. *BINANCE-ACADEMY.* www.binance.vision/blockchain/history-of-blockchain. 25th June 2019.
3. **Hal Finney.** Reusable Proofs of Work. *Nakamoto Institute.* https://nakamotoinstitute.org/finney/rpow/index.html. 25th June 2019.
4. **Ray King.** Token vs Coin: What's the Difference?. *Bit Degree TUTORIALS.* www.bitdegree.org/tutorials/token-vs-coin/. 26th June 2019.
5. **Rajarshi Mitra.** Utility Tokens vs Security Tokens: Learn The Difference – Ultimate Guide**.** *Blockgeeks.* https://blockgeeks.com/guides/utility-tokens-vs-security-tokens/. 26th June 2019.
6. **Jack Filiba.** Decentralized Applications. *Coinsquare.* https://news.coinsquare.com/learn-coinsquare/dapps-a-look-into-the-world-of-decentralized-applications/. 27th June 2019.
7. **Lucas Mostazo.** What is BLOCKCHAIN? The best explanation of blockchain technology**.** *Youtube.* www.youtube.com/watch?v=3xGLc-zz9cA. 28th June 2019.
8. **—.** Hashing**.** *Lisk.* https://lisk.io/academy/blockchain-basics/how-does-blockchain-work/what-is-hashing. 28th June 2019.
9. **Christof Paar**. Lecture 21 (update): SHA-3 Hash Function by Christof Paar. *Youtube.* www.youtube.com/watch?v=JWskjzgiIa4. 29th June 2019.
10. **—.** MD4. *Wikipedia.* https://en.wikipedia.org/wiki/MD4. 29th June 2019
11. **—.** How does a blockchain work. *Savjee Simply Explained.* www.savjee.be/videos/simply-explained/how-does-a-blockchain-work/. 30th June 2019.
12. **Ray Patterson.** The Proof-of-Work in Cryptocurrencies: Brief History. Part 1. *BYTECOIN.* https://bytecoin.org/blog/the-proof-of-work-in-cryptocurrencies-brief-history-part-1. 30th June 2019.
13. **DANIEL.** What's a Merkle Tree? Komodo's Guide To Understanding Merkle Trees. *KOMODO.* https://komodoplatform.com/whats-merkle-tree/. 1st July 2019.
14. **Jimi S. .** Blockchain: How mining works and transactions are processed in seven steps. *GOODAUDIENCE.* https://blog.goodaudience.com/how-a-miner-adds-transactions-to-the-blockchain-in-seven-steps-856053271476. 1st July 2019.
15. **Dr. Gavin Wood .** ETHEREUM: A SECURE DE. *Ethereum.* https://ethereum.github.io/yellowpaper/paper.pdf. 2nd July 2019.
16. **—.** Ethereum. *Wikipedia.* https://en.wikipedia.org/wiki/Ethereum. 3rd July 2019.
17. **—.** Ethereum Explained: Merkle Trees, World State, Transactions, and More. *PEGASYS.* https://pegasys.tech/ethereum-explained-merkle-trees-world-state-transactions-and-more/. 4th July 2019.
18. **—.** ERC-20. *Wikipedia.* https://en.wikipedia.org/wiki/ERC-20. 4th July 2019.

19. **Chris Chinchilla.** A Next-Generation Smart Contract and Decentralized Application Platform. *Github.* https://github.com/ethereum/wiki/wiki/White-Paper. 5th July 2019.

20. **—.** Trie. *Wikipedia.* https://en.wikipedia.org/wiki/Trie. 6th July 2019.

21. **JAKE FRANKENFIELD.** Initial Coin Offering (ICO). *Investopedia.* www.investopedia.com/terms/i/initial-coin-offering-ico.asp. 7th July 2019.

22. **—.** Personal blockchain for Ethereum Development. *Github.* https://github.com/trufflesuite/ganache. 8th July 2019.