מסמך חקר

על מנת לממש את הפרויקט חקרתי ארבעה נושאים עיקריים: Block Chain, ממימוש עסקאות בביטקוין, הצפנות א-סימטריות ותקשורת .p2p

**Block Chain**

בלוקצ'יין היא תפיסה טכנולוגית במחשוב המאפשרת פעילות עסקית מאובטחת באינטרנט ואימות של עסקאות בין צדדים שונים ללא צורך בישות ניהול מרכזית. את הגורם המנהל מחליפים "בלוקים" מוצפנים של מידע הנוצרים באמצעות שיתוף מבוסס רשת תקשורת מסוג עמית לעמית (p2p). (הגדרה מתוך ויקיפדיה)

הבלוקצ'יין הומצא על ידי סאטושי ממציא הביקוין.

אם כך, הבלוקים בשרשרת הבלוקים מכילים את המידע שצריך להכיל (לדוגמא העסקאות שנעשו עם המטבע) והם מועברים בין כל המתחברים לרשת (nodes או peers). לכן, אם node מכיל מידע לא תקין רוב הרשת לא תסכים עליו ולכן הוא לא יחשב.

על מנת להוסיף עוד בלוק לשרשרת צריך להראות "שעבדת עליו". כל בלוק מכיל את הhash של הבלוק לפניו ולכן על מנת לשנות את המידע בבלוק מסוים, לדוגמא לעלות את כמות הכסף הנוכחי שלך על ידי ביטול עסקה שעשית בתמורה לכסף אמתי, צריך לשנות את כל המידע של הבלוקים אחריו ומאחר שבשביל ליצור בלוקים לוקח זמן ("מבצעים עבודה"- יורחב בהמשך), מי שמנסה לשנות את השרשרת לא יספיק לפני יצירת הבלוקים החדשים ולכן הוא לא יספיק לפרסם את השרשרת או שיפרסם שרשרת לא תקנית ושאר האינטרנט לא יאשר אותה.

"העבודה" שנעשית על בלוק היא לקחת את כל המידע של הבלוק ועוד שדה אחד של מספר משתנה ולבצע פונקציית hash על המידע, אם התוצאה של הפונקציה כולל מספר מסוים של אפסים (difficulty) בתחילת קוד הhash שמתקבל אז הבלוק תקין. לכן, על מנת לקבל בלוק תקין צריך למצוא את המספר שמספק בלוק תקין, פעולה זאת לוקחת זמן. בפרויקט שלי מספר האפסים הוא קבוע והוא אפס.

על מנת לעודד כרייה, משום שבלי זה הבלוק צ'יין לא יוכל להתקיים, הכורים מתוגמלים במספר מטבעות מסוים. גם הוא קבע בתוכנה שלי והוא 50. דרך זאת, היא הדרך היחידה שנוצר כסף במערכת. בביטקוין האינפלציה הזאת קורת עד שיש מספר מסוים של מטבעות במערכת ואז הכורים יכולים להיות מתוגמלים רק על יד עמלת כרייה שתהיה בעסקאות.

**הצפנה א-סימטרית**

Bitcoin והמטבע שלי הם מטבעות מסוג cryptocurrency. כלומר, המימוש שלהם הוא בעזרת מפתחות ציבוריים ופרטיים.

המימוש בשני מפתחות לכל משתמש נקרא הצפנת מפתח ציבורי.

הצפנת מפתח ציבורי (Public key encryption) היא ענף ב[קריפטוגרפיה](https://he.wikipedia.org/wiki/%D7%A7%D7%A8%D7%99%D7%A4%D7%98%D7%95%D7%92%D7%A8%D7%A4%D7%99%D7%94) הנקרא גם אצפנה אסימטרי, שבו מפתח ההצפנה שונה ממפתח הפענוח. כלומר, כל משתמש מכין לעצמו זוג מפתחות: מפתח ציבורי שהוא [מפתח הצפנה](https://he.wikipedia.org/wiki/%D7%9E%D7%A4%D7%AA%D7%97_%D7%94%D7%A6%D7%A4%D7%A0%D7%94) הנגיש לכל ומפתח פרטי (Private key) מתאים, הנשמר בסוד ומשמש לפענוח. ההתאמה היא [חד-חד-ערכית](https://he.wikipedia.org/wiki/%D7%97%D7%93_%D7%97%D7%93_%D7%A2%D7%A8%D7%9B%D7%99%D7%AA) (לכל מפתח ציבורי קיים אך ורק מפתח פרטי יחיד המתאים לו, ולהפך). כדי להצפין מסר בשיטה זו על המצפין להשיג לידיו עותק אותנטי של המפתח הציבורי של המקבל, שבעזרתו הוא מצפין ושולח לו את המסר. רק המקבל מסוגל לשחזר את הטקסט המוצפן בעזרת המפתח הפרטי המתאים שברשותו. ביטחון שיטת המפתח הציבורי נשען על הקושי שב[חישוב](https://he.wikipedia.org/wiki/%D7%97%D7%99%D7%A9%D7%95%D7%91) המפתח הפרטי מתוך המפתח הציבורי. מסיבה זו מכונה שיטה זו "א-סימטרית", בניגוד לשיטת הצפנה [סימטרית](https://he.wikipedia.org/wiki/%D7%A6%D7%95%D7%A4%D7%9F_%D7%A1%D7%99%D7%9E%D7%98%D7%A8%D7%99), שבה מפתח הפענוח זהה למפתח ההצפנה (על כל פנים ניתן לחישוב בקלות מתוך מפתח ההצפנה).

המונח הצפנת מפתח ציבורי מתייחס לשיטות קריפטוגרפיות שהדגש בהן הוא שימוש במפתח הצפנה לא סודי. השימושים העיקריים במפתח ציבורי הם:

* הצפנת מסרים - מסר המוצפן באמצעות המפתח הציבורי של המקבל, לא ניתן לפענוח על ידי איש מלבד המקבל המחזיק בידיו את המפתח הפרטי המתאים. במקרה זה הצפנת מפתח ציבורי מספקת סודיות או חשאיות.
* [חתימה דיגיטלית](https://he.wikipedia.org/wiki/%D7%97%D7%AA%D7%99%D7%9E%D7%94_%D7%93%D7%99%D7%92%D7%99%D7%98%D7%9C%D7%99%D7%AA) - מסר שנחתם באמצעות המפתח הפרטי של השולח, ניתן לאימות על ידי כל מי שיש לו גישה למפתח הציבורי של השולח. באמצעותו ניתן להוכיח מי הוא מקור המסמך. במקרה זה ההצפנה מספקת זיהוי.
* [אימות מסרים](https://he.wikipedia.org/wiki/%D7%A7%D7%95%D7%93_%D7%90%D7%99%D7%9E%D7%95%D7%AA_%D7%9E%D7%A1%D7%A8%D7%99%D7%9D) - מסר המאומת על ידי השולח שמייצר תג אימות מתאים ושולח אותו יחד עם המסר איתו המקבל יכול לוודא שלא בוצעו במסמך שינויים זדוניים בידי גורם זר ללא ידיעתו. במקרה זה הצפנת מפתח ציבורי מספקת אימות.
* שיתוף מפתחות - ב[פרוטוקול שיתוף מפתח](https://he.wikipedia.org/wiki/%D7%A4%D7%A8%D7%95%D7%98%D7%95%D7%A7%D7%95%D7%9C_%D7%A9%D7%99%D7%AA%D7%95%D7%A3_%D7%9E%D7%A4%D7%AA%D7%97) השולח והמקבל מחליפים ביניהם מידע כלשהו בערוץ פתוח הנגיש לכול ובתום חילופי המסרים המשתתפים חולקים ביניהם מידע סודי, כך שאיש לא יודע מהו, ממנו המשתתפים מכינים מפתח הצפנה.

בפרויקט שלי השימוש במפתח ההצפנה הוא לאימות. כל עסקה מכילה מידע ואת המפתח הציבורי של מי שמקבל את הכסף, מי שרוצה להשתמש בכסף חייב לספק חתימה שתתאים למפתח הציבורי ובכך יש הוכחה לכך שהכסף נמצא אצל המשתמש שלו. (יורחב בהמשך).

**עסקאות**

את מימוש העסקאות שלי מימשתי במבנה שממומש בדומה לביטקוין.

כל עסקה מורכבת ממבנה קבוע:

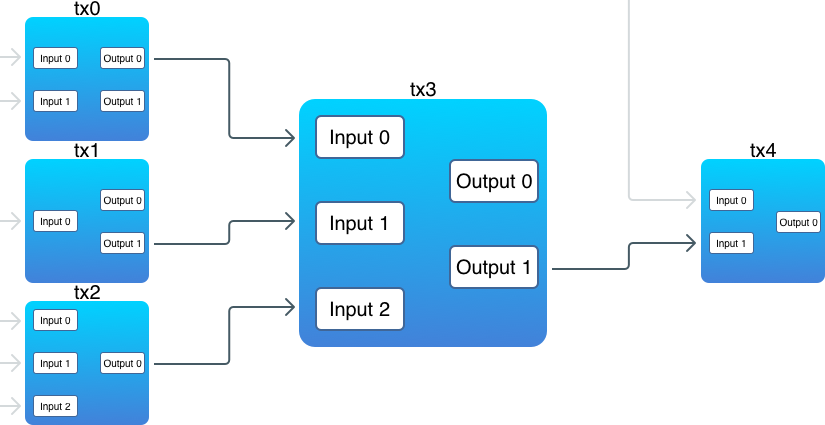
* הhash של האינפוטים והאאוטפוטים שמשמש כid של העסקה.
* רשימת אינפוטים (inputs) המממשים להוכחת מקור הכסף בעסקה.
* רשימת אאוטפוטים (outputs) שהם בעצם הענקת הכסף למי שאמור לקבל.

**האינפוט** מורכב מהid של העסקה שהעניק לו את הכסף לתשלום (יכולים להיות כמה עסקאות כאלו לכן יש כמה אינפוטים), מהאינדקס של ההאוטפוט בעסקה המצוינת, ומההוכחה שאתה יכול בעצם להשתמש באאוטפוט על מנת לבצע עסקאות עם הכסף.

ההוכחה מורכבת משני דברים: מהחתימה על הid של העסקה שממנה לקוח ההאוטפוט, על הכתובת של מקבל הכסף מהעסק שאתה יוצר ומהסכום, בכך אף אחד לא יכול להשתמש בחתימה על מנת להעביר כסף אליו, ומהמפתח הציבורי שלך.

**האאוטפוט** מורכב מכמות הכסף שאתה רוצה לשלם ומהכתובת שאתה רוצה לשלם אליה. במימוש שלי יכולים להיות שתי אינפוטים בעסקה אחד שמשמש כתשלום כסף כתוצאה מהעסקה ואחד שמשמש כעודף. מאחר שהיא אפשר להשתמש באינפוט על מנת להוכיח רק חצי אאוטפוט, על מנת לא לשלם את כל הסכום שמצוין באינטפוט, לדוגמא אם קיבלת 50 מטבעות ואתה רוצה להעביר למישהו אחר רק 20, צרי להוסיף עודף לעסקה. העודף הוא בעצם עוד אאוטפוט שמעביר את הכסף שצריך כעודף למי שיצר את העסקה.

העודף הוא כמו בחיים האמתיים, לדוגמא, אם אני רוצה לשלם על משהוא שעולה 10 שקל ויש לי רק 20 אני משלם את העשרים ומקבל בחזרה את העודף.



תהליך האימות של ההוכחה באינפוט מתבצע כך:

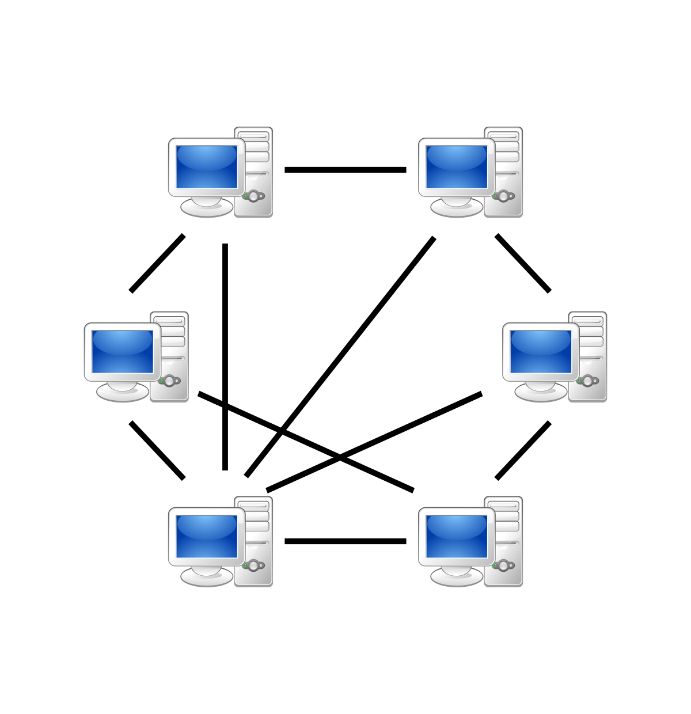
המאמת לוקח את המפתח הציבורי מההוכחה ועל ידיו מחשב את הכתובת של מי שמשתמש באאוטפוט (הכתובת של כל משתמש נבנית מהמפתח הציבורי שלו) והוא בודק אם הכתובת מתאימה לזאת שמצוינת באאוטפוט. יתר על כן, מתבצע בדיקה של החתימה (האימות החשוב), בעזרת המפתח הציבורי מאמתים אם החותם השתמש במפתח הפרטי שהמפתח הציבורי שייך אליו, ואם פרטי העסקה מתאימים לחתימה.

**תקשורת p2p**

בשונה מהתקשורת הנפוצה ברשת- בין שרת ללקוח. הרעיון במטבעות מאובזרים (כמו שלי) זה שאין שרת מרכזי שמפקח על העברות ובודק תשלומים וכו' אלא כל הרשת (המחשבים שמריצים את הארנק) אחראים על כך. לכן, הכסף לא נמצא בשליטה יחידה של שרת, בנק או חברה מסוימת והוא נמצא בשליטת כולם. בכך אין עמלה גבוהה על העסקאות במטבע (אצלי אין בכלל).

המימוש של תקשורת הוא כך שכל מחשב שמריץ את התוכנה (נקרא גם peer או node) הוא יכול לשמש גם כשרת המספק שירותים וגם כלקוח המבקש אותם. במימוש שלי השרת אחראי לקבלת ההודעות מהמשתמשים האחרים והלקוח על השליחה של ההודעות.

כל node שמתנהג לפי החוקים שמוגדרים בתוכנה הוא רכיב מהימן, ואחד שלא הוא לא. הרשת מתבססת על כך שרוב המחשבים הם מהימנים ואם המצב לא יהיה כך היא תקרוס או לא תעבוד טוב.

תרשים התקשורת בp2p

בפרוטוקול תקשורת של ביטקוין ישנם סוגים שונים של הודעות. אני ממש את ההודעות הבאות. כל הודעה מכילה את הכתובת של השולח:

* Version- כאשר node מתחבר למערכת מחדש הוא שולח את ההודעה הזאת והוא בעצם מעדכן את הרשת במצב הבלוקצ'יין שלו. ההודעה מכילה את האורך של הבלוקצ'יין שלו. כתגובה הpeers האחרים מחזירים לו הודעת version אם מצב הבלוקצ'יין שלו לא מעודכן ממצב האחרים הוא צריך להוריד את הבלוקצ'יין המעודכן מהרשת.
* Addr- הודעה זאת מכילה רשימת כתובות שהnode הנוכחי מכיר.
* Getblocks- בקשה להראות אילו בלוקים יש לך מהבלוק האחרון שיש לי. בתשובה תהיה בהודעת inv .
* Inv- הודעה זאת מראה אילו עסקאות או בלוקים יש לשולח. ההודעה מכילה את הסוג של המידע שבתוכה- בלוק או עסקה ואת הhashes של הבלוקים או העסקאות.
* Getdata- הודעה זאת היא בקשה לעסקה מסוימת או בלוק מסוים. ההודעה מכילה את סוג המידע המבוקש ואת הhash שלו.
* Transaction- הודעה זאת היא תגובה לgetdata של עסקה והיא מכילה מחרוזת שממנה יוכל המקבל להרכיב את העצם של העסקה.
* Block- הודעה זאת היא תגובה לgetdata של בלוק והיא מכילה מחרוזת שממנה יוכל המקבל להרכיב את העצם של הבלוק.
* Getadresses – בקשה לכתובות הידועות כפעילות.

כאשר המשתמש מריץ את תוכנת הארנק, בהתחלה הארנק יחפש peers מחוברים על ידי הסתכלות בבסיס הנתונים שלו על כתובות שהוא התחבר בעבר. הכתובות האלו יהיו מהתחלה כוללים בתוכם כתובות של "מארחים" ידועים שהם בעצם משתמשים שידוע שהם רצים הרבה זמן או כל הזמן ואפשר להתחבר אליהם על מנת לקבל כתובות של משתמשים חדשים שלא ידועים לnode.

ביבליוגרפיה

* Tutorial: Building Blockchain in Go

https://jeiwan.cc/posts/building-blockchain-in-go-part-1/

* סרטונים שמסבירים על הבלוקצ'יין והמימוש שלו:

<https://www.tutorialspoint.com/blockchain_online_training>

* מידע על בלוקצ'יין:

<https://www.edureka.co/blog/blockchain-tutorial/>

* מידע על אימות חתימות ועל ההצפנה בבלוק:

<https://blockgeeks.com/guides/best-bitcoin-script-guide/>

* מידע על הפרוטוקול בביטקוין:

<https://en.bitcoin.it/wiki/Network#Bootstrapping>

* מימוש p2p:

<http://cs.berry.edu/~nhamid/p2p/>

* ויקיפדיה על block chain:

<https://he.wikipedia.org/wiki/%D7%91%D7%9C%D7%95%D7%A7%D7%A6%27%D7%99%D7%99%D7%9F>

* ויקיפדיה על מפתח ציבורי:

<https://he.wikipedia.org/wiki/%D7%9E%D7%A4%D7%AA%D7%97_%D7%A6%D7%99%D7%91%D7%95%D7%A8%D7%99>