digital whisper Technologic Papers

RBCS - A Blockchain Based Reverseshell

מאת שקד אשכנזי

הקדמה

לפני כ-3 חודשים, אבא שלי שאל אותי אם אני חושב שכדאי להשקיע ב-Bitcoin. עניתי לו שאני לא מכיר מספיק את התחום כדי לענות על השאלה. אז התחלתי לחקור על הנושא, ולאט לאט הבנתי ש-Blockchain הוא הרבה מעבר ל-Crypto Currencies ושיש לו פוטנציאל לא ממומש במגוון תחומים. אחד מהם הוא האנונימיות.

במאמר זה, אציג את יתרונות האנונימיות שה-Blockchain מעניק לנו ואסביר על כלי שכתבתי בשם: ReverseBlockchainShell. אך לפני כדי לתת רקע בתחום, נענה על מספר שאלות:

1. איך תהליך ה-Blockchain עובד?
2. מה זה Ethereum ומהן הרשתות השונות?
3. מהו חוזה חכם ואיך כותבים אותו?

אז מזה בעצם Blockchain?

רוב האפליקציות כיום מתבססות על פנייה למקור ספציפי שמכיל את המידע הרצוי על מנת לשלוף או לשנות אותו.

כמו:

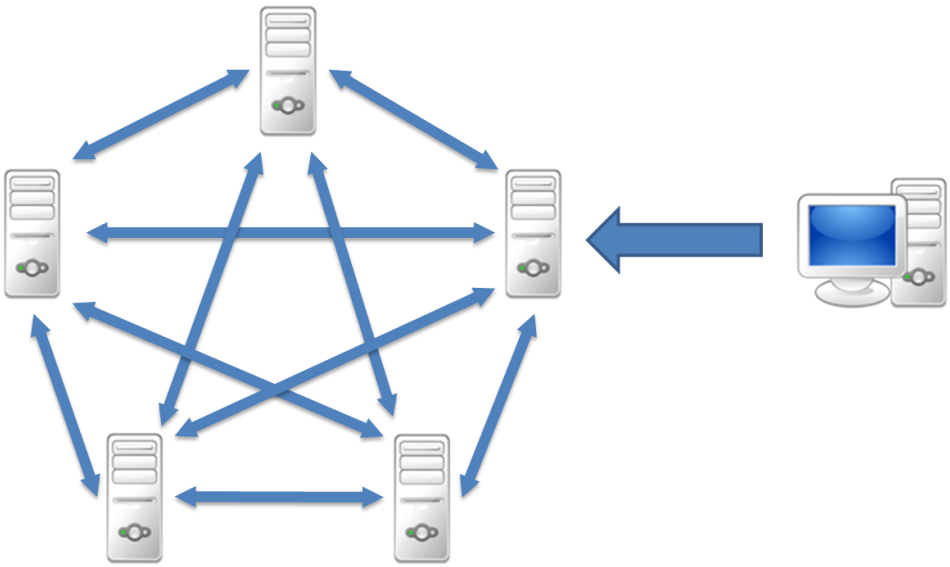
- HTTP
- RDP
- SMB
- SMTP

לקוח - קורא / כותב מידע

שרת - מכיל מידע

[שימו לב שגם רשת P2P כמו BitTorrent עובדת בצורה זו. על מנת להשיג משאב מסוים, אנו ניגש לעמדה מסוימת ברשת]

Blockchain היא רשת המורכבת ממחשבים הנקראים Nodes, המכילים מידע ושקולים אחד לשני. כלומר שניתן לקרוא או לכתוב מידע מהרשת מאיזה Node שנבחר.



כמו:

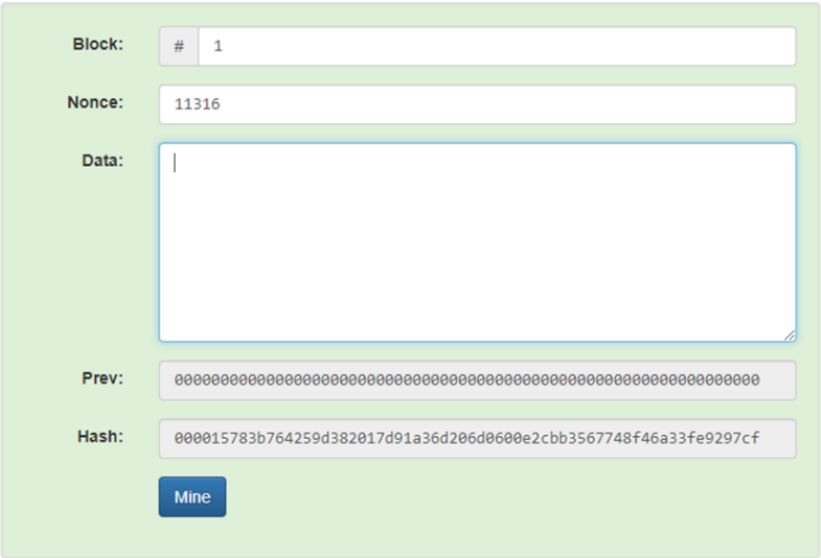
- Bitcoin
- Ethereum
- Flux
- וכו'

לפני שנבין את כל השימושים השונים של ה-Blockchain, ראשית נבין את השימוש הראשוני שלה, שהוא העברת כספים דיגיטלית. קונספט זה נוצר כדי שיהיה אפשר לסחור ולהעביר כספים מבלי לסמוך על אף גורם מתווך (בנקים לדוגמא).

דיסקליימר קטן: במאמר אתייחס למימוש הבסיסי של Blockchain אף כי יש מימושים נוספים, וכל רשת Blockchain ממומשת טיפה אחרת. המידע הבא נלקח ממקורות שונים באינטרנט אך ניתן למצוא את כולו ב-Bitcoin whitepaper, ב-Ethereum whitepaper או באתר של Ethereum.

איך ה-Blockchain עובד?

ה-Blockchain כשמו כן הוא, מבוסס על שרשרת של בלוקים, שמקושרים אחד לשני. כל Block מכיל בתוכו 5 שדות:



השדות הם:

- שדה ה-**#Block**: הוא המספר הסידורי של הבלוק, מספר ייחודי לו בשרשרת.
- שדה ה-**Data**: מכיל את כל העברות הכספים (Transactions) המשוייכות לאותו הבלוק. מי העביר למי וכמה.
- שדה ה-**Hash**: הוא בעצם שיכלול כל הבלוק (מלבד שדה ה-hash) לרצף בינארי, שעובר hash (במקרה של Bitcoin או Ethereum מדובר ב-sha256).
- שדה ה-**Prev**: מכיל את ה-hash של הבלוק הקודם.
- שדה ה-**Nonce**: מכיל מספר חסר משמעות, שקיים רק כדי לשנות את ה-Hash של הבלוק. נרחיב עליו בהמשך.

השימוש ב-hash-ים מבטיח את אמינות המידע. אם אנסה לשנות את שדה ה-Data בבלוק אחד, זה ישפיע על ה-hash של אותו בלוק. מכיוון שכל בלוק מכיל את ה-hash של הבלוק הקודם לו, שינוי של hash אחד ישפיע על ה-hash של הבלוק הבא לו שישפיע על ה-hash של הבלוק הבא וכן הלאה...

בצורה זאת בעקבות הקשר בין בלוק לקודמו, לא ניתן לשנות את המידע שאחד הבלוקים מכיל מבלי לפגוע בשרשרת ו"להרוס" אותה.

איך עובד תהליך הכרייה?

כשמשתמש ברשת רוצה לבצע העברה, הוא פונה ל-Node ברשת. ה-Node מעביר את הבקשה לכל שאר ה-Nodes והם מבצעים תהליך אימות לבקשה (כל אחד בעצמו) עליו יפורט בעמוד הבא.

במידה והאימות מתבצע באופן תקין, ההעברה מתווספת לשדה ה-Data כטרנזקציה נוספת בבלוק החדש. כאשר Node מאמת מספיק בקשות כדי להרכיב בלוק חדש (כ-1500~ ב-Ethereum בעת כתיבת המאמר), הוא מנסה "לכרות" אותו.

כרייה היא בעצם התהליך החישובי שבו ה-Node משנה את שדה ה-nonce בבלוק הנוכחי שוב ושוב ושוב עד שלבסוף הוא מצליח למצוא nonce שיגרום ל-hash של הבלוק להתחיל עם 30 אפסים כקונבנציה (כמות האפסים אינה קבועה ושונה בין רשת לרשת). כאמור, ערך ה-nonce הוא חסר משמעות מלבד יצירת ה-hash הייחודי.

כשהוא מוצא את אותו ה-nonce, ה-Node מכריז על כך, שולח לשאר ה-Node-ים ברשת את הבלוק והם מוסיפים אותו לתיעוד האישי שלהם. בתמורה למציאת ה-nonce, הוא זוכה בכסף דיגיטלי בעצמו. לכן, על מנת ש-Node יוכל להוסיף בעצמו בלוק חדש ל-Blockchain (ולזכות בפרס על כך), עליו לעבוד קשה כדי למצוא את ה-nonce.

תהליך זה נקרא Proof of Work - PoW כיוון שה-nonce מהווה הוכחה שאותו ה-node עבד. ישנן שיטות נוספות למימוש Blockchain כגון Proof of Stake - PoS אך עליהן לא נדבר במסגרת המאמר.

הערת צד: נמצא כי מעבדים גרפיים שימושיים במיוחד לחישובי מציאת ה-nonce, ולכן בשנים האחרונות נרשם זינוק הן בביקוש והן המחיר שלהם...

נקודה נוספת שהחסרתי בתהליך היא מס, או בשמו ב-Blockchain נקרא Gas. לא נכנס אליה לפרטים במסגרת המאמר אך דעו שעל כל העברה, המשתמש המעביר משלם סכום סמלי על מנת שההעברה תתבצע. הסכום הסמלי מועבר ל-Node שכרה את הבלוק. מס זה הכרחי לתהליך קבלת ההעברות.

איך עובד תהליך האימות?

תהליך האימות מבוסס על הקונספט של הצפנה א-סימטרית. הצפנות מודרניות מתחלקות לשני סוגים עיקריים - הצפנה סימטרית והצפנה א-סימטרית. בהצפנה סימטרית המפתח שמשמש להצפין את המידע משמש אותנו גם לפענוח המידע. לעומת זאת בהצפנה א-סימטרית מפתח ההצפנה שונה ממפתח הפענוח.

בהצפנה זו אחד המפתחות יקרא Public Key והשני Private Key, כיוון שאחד מהמפתחות מונגש לציבור לשימוש והשני נשאר סודי. ניתן להשתמש במפתחות בשתי דרכים:

1. להצפין עם המפתח הציבורי ולפענח עם הפרטי.
2. להצפין עם המפתח הפרטי ולפענח עם הציבורי.

לכל משתמש ב-Blockchain יש מפתח פרטי (SK), מפתח ציבורי (PK) וכתובת לחשבון שלו, כאשר הכתובת נגזרת מהמפתח הציבורי. ז"א שאפשר לשייך בין מפתח ציבורי לכתובת הארנק. נניח שאליס רוצה לבצע העברה של 1 Bitcoin לבוב. כדי לעשות זאת, אליס יוצרת בקשה של העברת 1 Bitcoin מהחשבון שלה לחשבון של Bob:

לאחר מכן מצפינה את הבקשה עם המפתח הפרטי שלה:

ולבסוף שולחת אותה לאחד מה-Nodes ברשת בצירוף המפתח הציבורי:

ה-Node מקבל את הבקשה, ומשתמש במפתח הציבורי שהוא קיבל כדי לפענח אותה ולקרוא את תוכן הבקשה. מיד לאחר מכן, הוא מוודא שהמפתח הציבורי שהוא קיבל, תואם לכתובת המעבירה (From). מכיוון שהמפתח הפרטי שייך לבעל הארנק בלבד, הגורם היחיד שיכול לבצע העברות הוא בעל הארנק.

לאחר וידוא האמינות של הבקשה, ה-Node מוודא שבעל הארנק מחזיק בכמות הכסף שהוא רוצה להעביר על ידי מעבר על היסטוריית ההעברות שלו.

סיכום התהליך הכולל

כאשר משתמש מבצע העברה, הוא חותם את הבקשה עם המפתח הפרטי שמשויך לכתובת שלו, ושולח את הבקשה לאחד ה-Nodes ברשת. ה-Node שקיבל את הבקשה מעביר אותה לשאר ה-Nodes ומאשר את הבקשה. לאחר מכן הוא (וכל שאר ה-Nodes גם כן) מוסיף אותה לבלוק שהוא מנסה ליצור ומקבל בקשות נוספות. כשהוא מקבל מספיק בקשות, הוא מנסה לכרות את הבלוק על ידי שינוי שדה ה-nonce למציאת ה-hash הנדרש.

ה-Node בר המזל שמוצא את ה-nonce, משתף את כולם ב-nonce שהוא מצא, וזוכה בפרס כספי וגם ב-Gas (מס) שכל המעבירים שילמו מראש. לבסוף כל ה-Nodes מוסיפים את הבלוק החדש לשרשרת ומתחילים את התהליך מחדש.

Test-Nets-ו Ethereum

Ethereum היא אחת מהמימושים לרשת Blockchain והמטבע בה נקרא Ether. קיימות מספר רשתות מבוססות על הטכנולוגיה של Ethereum. לעומת Bitcoin, לרשתות Ethereum יש יכולות נוספת מעבר להעברת כספים, והיא חוזים חכמים. חוזים חכמים מאפשרים למשתמשים ברשת להריץ קוד ולשמור מידע על גבי ה-Nodes ברשת.

הרשת המרכזית נקראת ה-MainNet, וקיימות רשתות נוספות מבוססות Ethereum שנקראות TestNets. ב-TestNets הכסף שסוחרים בו נקרא Test-Ether. ה-Test-Ether הוא חסר ערך וניתן לקבלו על ידי בקשה לאתרים מסוימים.

חוזים חכמים

חוזים חכמים הם קטעי קוד הנמצאים על ה-Blockchain כחלק משדה ה-Data שבבלוק, ומאפשרים למשתמשי קצה להריץ פונקציות מתוך החוזה על גבי ה-Nodes. לחוזים חכמים ישנם מספר רחב של שימושים, החל מ-NFTs (שהם קצת יותר מ-JPEG לעומת הדעה הרווחת...) וכלה באפליקציות או אפילו חברות שלמות שיכולות להתנהל על גבי ה-Blockchain. ב-Ethereum החוזים החכמים נכתבים בשפה Solidity. דוגמא לחוזה חכם:

```
// SPDX-License-Identifier: UNLICENCED
pragma solidity ^0.6.0;

contract StoreNum {
    uint256 private num = 0;

    // SetNum() gets a number and sets private variable "num" to it.
    function SetNum(uint256 number) public {
        num = number;
    }

    // GetNum() returns the number stored in "num" variable.
    function GetNum() public view returns (uint256) {
        return num;
    }
}
```

בחוזה ה"חכם" הנל ניתן לשמור מספר במשתנה על ידי קריאה לפונקציה SetNum ולקרוא את המספר על ידי קריאה לפונקציה GetNum. כאשר נקרא לפונצקיה SetNum, המספר שנעביר לה ישאר על גבי ה-Blockchain עד השימוש הבא ב-SetNum שהוא יוחלף.

Call vs. Transact

ישנן שתי דרכים לקרוא לפונקציה של חוזה חכם, אחת היא Call והשניה Transact. פונקציות שמשנות מידע ב-Blockchain יקראו באמצעות Transact בעוד שפונקציות שאינן משנות מידע ב-Blockchain יקראו באמצעות Call. כאשר אנו קוראים לפונקציה עם Transact, על מנת ששינוי הערך יתועד, נצטרך שהבקשה שלנו תרשם ל-Blockchain. על כך, אנו נשלם Gas כמו בכל העברה ב-Blockchain. לעומת זאת, בקשות Call הינן לקריאה בלבד. זאת אומרת שלא נצטרך לשמור מידע ב-Blockchain ולכן לא נצטרך לשלם Gas.

ב-Solidity כאשר נרצה לכתוב פונקציה לקריאה בלבד, נוסיף לה את התגית "view" וכך יהיה ניתן לקרוא לה רק באמצעות Call (כמו שניתן לראות בפונקציה GetNum).

אנונימיות

יצא לכם לשמוע על פושעים שדורשים תשלום במטבע שמבוסס על רשת Blockchain כגון Bitcoin או Ethereum מתוך הנחה שכך לא יוכלו לעקוב אחרי הכסף? אם כן, אז דעו לכם שההנחה לא מדויקת.

Blockchain בצורתו הבסיסית הוא פסודו-אנונימי, משמע הוא אנונימי למחצה. אמנם על הרשת עצמה אנחנו מזדהים רק באמצעות כתובת ארנק, מפתחות פרטיים וציבוריים אך הרשת עצמה מוגדרת כציבורית ולכן כלל הפעולות שקרו ברשת גלויות לעיני כל מי שמעוניין. למשל, כדי לקרוא את תוכן הבלוקים של Ethereum ניתן להשתמש באתר EtherScan.

לכן ע"י קריאת ה-Blockchain ניתן לעקוב אחרי ארנק מסוים, לצפות בכל הפעולות שהוא ביצע ובסופו של דבר (עם מספיק מידע) גם לאתר אותו.

לדוגמא:

נגיד ונרצה להשיג את שמו של מחזיק בכתובת X.

ידוע לנו שכתובת X העבירה 1 Bitcoin לכתובת Y, וידוע לנו מיהו בעל הארנק Y. אז ניצור קשר עם בעל ארנק Y ונבקש ממנו מזהים של בעל ארנק X.

בדוגמא, התבססנו על כך שיש לנו שיוך בין כתובת Y לבין בעל הארנק Y. הנחה זו היא אפשרית כיוון שעל מנת לקנות מטבעות דיגיטליים כמו Bitcoin או Ethereum ממקורות רשמיים מסויימים, על הקונה להזדהות עם תעודה מזהה כלשהי. כך הגוף שמכר את המטבעות הדיגיטליים יכול לעקוב אחרי הפעולות של אותו חשבון ולקשרן לזהותו של בעל הארנק, ובנוסף לזהות בעלי ארנקים שסחרו עם אותו אדם (כמו X בדוגמא).

זיהוי ע"פ כתובת IP

האנונימיות שאנו מקבלים בזכות השימוש בתשתית ה-Blockchain היא בין היתר בגלל שה-Blockchain לא רץ במקום מרוכז (שרת/DataCenter/Cloud וכו'). אם ה-Blockchain היה רץ במקום מרוכז היו יכולים לאתר את משתמשי ה-Blockchain ע"י חיפוש בלוגים של השרת הרלוונטי ומציאת כתובת ה-IP שהם פנו דרכה.

בעקבות כך שה-Blockchain רץ בצורה מבוזרת אין לדעת דרך איזה Node משתמש יצר את החיבור ל-Blockchain ולכן לא ניתן יהיה למצוא קישור לכתובתו.

אז לאחר שכיסינו את נושא זה, אפשר להתחיל לדבר על הכלי.

הכלי ReverseBlockchainShell (השימוש בכלי הינו למטרות למידה בלבד!)

ReverseBlockchainShell או בקיצור RBCS, הינו Shell נורמטיבי אשר מעביר את המידע בין ה-Client ל-Server דרך ה-Blockchain. אם אינכם יודעים מזה ReverseShell תוכלו לקרוא על הכלי במאמר "לא אנומאלי ולא במקרה" או ממקור אחר באינטרנט.

המימוש שלי לכלי נכתב ב-Solidity על גבי Ethereum Blockchain וצד לקוח ושרת ב-Python. אך הכלי יכול לרוץ על רשתות מסוגים שונים באותה מידה.

כאשר התוקף ירצה להריץ פקודה על הנתקף, הוא יפנה לחוזה חכם אשר ישמור את הפקודה ל-Blockchain והנתקף יקרא מתוכו. לאחר מכן הנתקף יריץ את הפקודה ויפנה לחוזה החכם אשר יכתוב את הפלט שלה ל-Blockchain. לבסוף התוקף יקרא את פלט הפקודה וישלח פקודה חדשה להרצה.

החוזה החכם שרץ על ה-Blockchain נראה בבסיסו ככה:

```
contract Shell {
    string command = "";
    string output = "";


    // SetCommand() gets a string and put it in "command" variable, for the client to run.
    // SetCommand() can only be called by owner.
    function SetCommand(string memory command_) public {
        command = command_;
    }

    // GetCommand() return the command requested (by the attacker) to the client.
    function GetCommand() public view returns (string memory) {
        return command;
    }

    // SetOutput() gets a string and puts it "output" variable. Also sets isOutputReady to true.
    function SetOutput(string memory output_) public {
        output = output_;
    }

    // GetOutput() return the command output.
    // GetOutput can only be called by owner.
    function GetOutput() public view returns (string memory) {
        return output;
    }
{
```

- פונקציות Set ו-Get להזנה וקריאה של פקודות.
- ופונקציות Set ו-Get להזנה וקריאה של פלט הפקודות.

החוזה החכם המלא מכיל פונקציות ופרמטרים נוספים לייעול התהליך ולשמירה על סנכרון. ניתן למצוא את כל הקוד מקור ב-github.

אנונימיות ב-Testnets

כמו שאנחנו כבר יודעים, על מנת לשמור מידע ב-Blockchain, נצטרך להשתמש ב-transact כדי לבצע טרנזקציה ועל מנת לעשות זאת, נצטרך לשלם מס הנקרא Gas.

RBCS מבצע 2 טרנזקציות על כל פקודה שנריץ וזה יוצר הרבה עלויות Gas. במיוחד אם נרצה שהכלי יעבוד יחסית מהר (ככל שנשלם יותר Gas הטרנזקציות יעברו יותר מהר). אבל אני ישראלי. ואני לא הולך לשלם $0.24 (המס הממוצע בעת כתיבת המאמר) על כל פקודה שארצה להריץ. אבל אין דאגה, בשביל זה יש לנו Testnets. בגלל שערך המטבעות ב-Testnet הוא אפסי, אם אשתמש ב-Testnet של Ethereum אוכל להריץ בדיוק את אותו קוד ללא עלות. אך קיימות 2 בעיות לפנינו בעת השימוש ב-Testnet:

זהות רשתית

Testnets במהותן מתבססות על מטבעות ללא ערך ולכן אין מניע לגורם חיצוני להשתתף בתהליך כ-Node. ה-Nodes לא מקבלים כסף אמיתי מכריית הבלוקים אלא רק מטבעות חסרי ערך. מסיבה זו ה-Testnets לרוב יהיו מרוכזים במקום אחד (אירגון שאחראי על ריצת ה-Testnet ומרוויח ממנה בדרך אחרת) ולכן יהיה ניתן לגלות את המשתמש ע"י תחקור של לוגים בשרת הרלוונטי. (איני יודע אם רשתות Testnets מחוייבות לשמור תיעוד של הפניות לרשת, אך תיאורטית זה אפשרי, ולכן אין הבטחה לשמירה על אנונימיות)

למזלנו, רשת הבדיקות Ropsten, כמו ה-MainNet, היא מבוזרת ומשתמשת ב-PoW. משמע, ש-Nodes ברחבי העולם מריצים את הרשת לעומת TestNets אחרים ובכך מאפשרים לנו להסתיר את זהותנו.

קבלת ה-TestEther

כדי לקבל TestEther המשתמש צריך לפנות לשירות הנקרא Faucet. אותו שירות מעניק TestEther בחינם לכתובת ארנק שנזין לו. ברוב המקרים, ה-Faucet ידרוש לעבור תהליך הזדהות כלשהו, אך יש גם כאלה שלא. אל תטעו, השירות אשר מעניק את ה-FakeEther דורש שהמשתמש יפנה אליו, ולכן משתמש ישאיר "עקבות" של הכתובת שממנה הוא ניגש עם קישור לאנק שלו. באמצעות מעבר על הלוגים שקיימים בשרת ה-Faucet, יהיה ניתן לקשר את כתובת ה-IP של המשתמש לכתובת הארנק שלו.

בינתיים לא מצאתי פתרון ארוך טווח להרצת הכלי על רשת בדיקות. אך תיאורטית אם הייתי יוצר ארנק ב-TestNet בעצמי, טוען אליו TestEther ומפרסם את פרטיו (כולל המפתח הפרטי שמאפשר לבצע העברות מהחשבון), היה ניתן להשתמש בכלי בחופשיות (עם החשבון שמקושר אלי).

כמובן שלא אעשה זאת גם כי הכלי הוא למטרות למידה בלבד, ומבחינתי ה-POC מספיק טוב למטרה זו (וגם כי לא הייתי רוצה שהשב"כ ידפוק לי על הדלת מחר בבוקר).

Mixers-ו Tumblers

ברשתות MainNet (בין אם Ethereum-MainNet או בין אם Blockchains אחרים) קיימים שירותים הנקראים Mixers או Tamblers. שירותים אלו מקבלים בקשות ממשתמשים רבים ומעבירים את כל הבקשות דרכם, כמו תחנת ביניים להעברות. ככל שיותר משתמשים משתמשים בשירות, כך הוא יותר אמין ויותר קשה לקשר את ההעברות.

השימוש בשירותים אלו יכול להיתפס כלא חוקי. משתמש אשר מסתיר את זהותו לחלוטין כנראה שרוצה להסתיר משהו (בין אם הלבנת כספים, פשעי סייבר או פשעים אחרים), או שהוא פשוט תומך במאבק לאנונימיות. אדם שהשתמש בשירותים אלו יכול להיות משוייך לפעולות בלתי חוקיות, ולכן אני ממליץ בחוזקה לא להשתמש בשירותים הללו.

בנוסף, בעקבות הדרך שבה עובד הכלי (קבלה של מטבעות לארנק השירות ומסירה של מטבעות למשתמש היעד) השימוש בכלי כזה מתועד ב-Blockchain. כיוון שכל המידע ב-Blockchain חשוף לציבור וכתובת הארנק של שירות ה-Mixing חשופה גם כן, ניתן לזהות משתמשי Mixing. כתוצאה מכך, אפליקציות מסויימות של העברה ומסחר במטבעות דיגיטליים עשויות לקטלג את הארנק כ"מסוכן" ולא לאפשר שימוש באפליקצייה לאותו משתמש.

סיכום

במאמר זה הוסבר על הבלוקצ'יין, תהליכי הכרייה שלו, והאנונימיות המסוימת שהוא מאפשר. בנוסף, הוצג הכלי RBCS במטרה להדגים שימוש במאפיין האנונימיות שהטכנולוגיה מעניקה לנו. מאפיין זה הוא אחד מני רבים שניתן לנצל בטכנולוגיה זו, וזהו רק קצה הקרחון של עולם ה-Blockchain.

במידה מסוימת מאמר זה הוא הזמנה שלי עבורכם לחקור וללמוד את הנושא, להטביע את חותמכם, ובכך להיות חלוצים בעולם תוכן חדש ומהפכני שנבנה בזממנו.