נאנו: רשת מטבעות קריפטוגרפיים מבוזרת ללא עמלות

קולין למהיו
clemahieu@nano.co

תקציר—לאחרונה, ביקוש גבוה ועמידה בעומס מוגבלת הגדילו את זמן העברת התשלום הממוצע ואת העמלות במטבעות קריפטוגרפים פופולריים, דבר שמוביל לחוויה לא מספקת. כאן, נציג את נאנו (NANO), מטבע קריפטוגרפי עם ארכיטקטורה חדשה בשם סריג־בלוקים (Block־lattice) כאשר לכל חשבון יש בלוקצ'יין פרטי משלו, מה שמוביל להעברות כמעט מיידיות ועמידה בעומס בלתי מוגבל. לכל משתמש יש בלוקצ'יין פרטי משלו, מה שמאפשר עדכון א־סינכרוני לשאר הרשת ויוצר העברות מהירות עם תקורה מינימלית. הפעולות שומרות על מעקב של מאזן החשבון במקום סכום הפעולות, מה שמאפשר גיזום אגרסיבי למסד הנתונים ללא פשרות באבטחה. נכון לעכשיו, רשת הנאנו העבירה 4.2 מליון פעולות עם פנקס חשבונות לא גזום בגודל של רק 1.7GB. התשלומים המיידים, חסרי העמלות של נאנו הופכים את המטבע למטבע הקריפטוגרפי המובחר עבור פעולות בין צרכנים.

מושגים – מטבע קריפטוגרפי, בלוקצ'יין, נאנו, פנקס חשבונות מבוזר, דיגיטלי, פעולות

I. הקדמה

מאז היישום של ביטקויין ב2009, החל מעבר גדול ממטבעות סטנדרטיים שנופקו על ידי הממשלה וממערכות פיננסיות לכיוון מערכות תשלום מודרניות מבוססות קריפטוגרפיה, אשר מציעות את האפשרות לאכסן ולהעביר כספים בצורה בטוחה ונטולת צורך באמון [1]. בכדי שיוכלו לפעול בצורה אפקטיבית, מטבע קריפטוגרפי חייב להיות קל להעברה, ללא אפשרות להחזרה ושיהיה בעל עמלה מזערית או ללא עמלה כלל. הזמן הנדרש לבצע תשלום, העמלות הגבוהות וחוסר העמידה בעומס העלו שאלות לגבי השימושיות היומיומית של הביטקוין כמטבע.

במאמר זה, נציג את נאנו, מטבע קריפטוגרפי בעל זמן השהייה נמוך הנבנה על מבנה נתונים חדשני בשם סריג־בלוקים המציע עמידה בעומס בלתי מוגבל ופעולות ללא עמלות. נאנו הוא פרוטוקול פשוט שמטרתו היחידה היא להיות מטבע קריפטוגרפי בעל ביצועים גבוהים. פרוטוקול הנאנו יכול לרוץ על חומרה בעלת עוצמה נמוכה, דבר המאפשר לו להיות מטבע קריפטוגרפי מבוזר פרקטי לשימוש יום יומי.

הסטטיסטיקות של המטבעות הקריפטוגרפים המוצגות במאמר הזה מדוייקות לזמן פרסום המאמר.

II. רקע

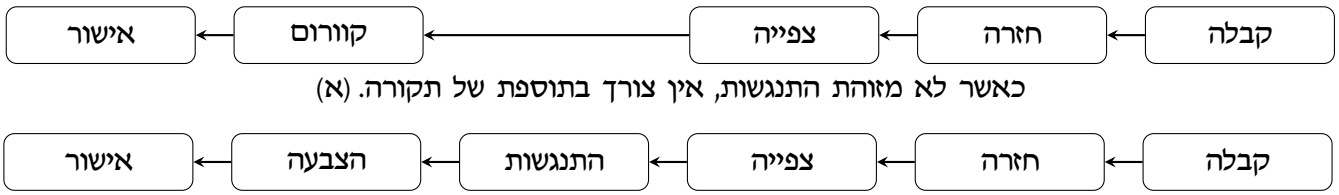
ב2008, אינדיבידואל אנונימי תחת שם העט סאטושי נאקאמוטו פרסם מאמר המציג את המטבע הקריפטוגרפי המבוזר הראשון בעולם, ביטקוין [1]. חידוש מרכזי שהביא הביטקוין הוא הבלוקצ'יין, מבנה נתונים מבוזר פומבי ולא ניתן לשינוי אשר משמש כפנקס חשבונות לתשלומי המטבע. לרוע המזל, ככל שביטקוין גדל, נחשפו מספר בעיות בפרוטוקול אשר יצרו בעייה עבור יישומים רבים:

1) אי עמידה בעומס: כל בלוק בבלוקצ'יין יכול לאחסן כמות מוגבלת של מידע, מה שאומר שהמערכת יכולה לעבד כמות מוגבלת של פעולות בשנייה, מה שהופך את מספר המקומות בבלוק למשאב. נכון לעכשיו, חציון עמלת התשלום הינו $10.38 [2].
2) זמן השהייה גבוה: זמן אישור תשלום ממוצע הינו 164 דקות [3].
3) חוסר יעילות בחשמל: רשת הביטקוין צורכת כ27.28TWh בשנה, כלומר 260KWh עבור פעולה בממוצע [4].

ביטקוין, ושאר מטבעות קריפטוגרפיים, פועלים על ידי השגת הסכמה על פנקסי החשבונות הגלובליים שלהם בשביל לאשר פעולות לגיטימיות תוך כדי התנגדות לגורמים זדוניים. ביטקוין משיג הסכמה על ידי דרך מדידה כלכלית הנקראת הוכחת עבודה (Proof of Work). במערכת הוכחת עבודה משתתפים מתחרים בתחרות בה המטרה היא לחשב מספר חד פעמי (nonce), כך שהגיבוב של כל בלוק נמצא בטווח מטרה. טווח המטרה הוא יחסי באופן הפוך לכוח כמות המחשוב המשותפת של כל רשת הביטקוין בכדי לשמור על זמן ממוצע קבוע למציעת מספר חד פעמי נכון. המשתתף שמצא מספר חד פעמי תקין זוכה באפשרות להוסיף בלוק לבלוקצ'יין; לכן, אלו שמשתמשים ביותר משאבים לחשב את המספר הזה לוקחים תפקיד גדול יותר במצב הבלוקצ'יין. הוכחת עבודה מאפשרת התנגדות נגד התקפת סיביל, שבא יישות מתנהגת כמספר יישויות בכדי להשיג כוח נוסף במערכת מבוזרת וגם בכדי להפחית את מספר מצבי המירוץ שקיימים באופן טבעי במבנה נתונים גלובלי.

פרוטוקול הסכמה אחר, הוכחת החזקה (Proof of Stake), הוצג לראשונה על ידי פירקוין (Peercoin) ב2012 [5]. במערכת הוכחת החזקה, משתתפים מצביעים עם קול משוקלל השווה לכמות המטבעות שהם מחזיקים. בסידור זה, בעלי ההחזקה הפיננסית הגדולה יותר מקבלים יותר כוח ובאופן טבעי מתומרצים לשמור על כנות המערכת או שיפסידו את השקעתם. הוכחת ההחזקה מבטלת את הצורך בתחרות המבזבזת כוח מחשוב ורק דורשת תוכנה קלת משקל הרצה על חומרה בעלת עוצמה נמוכה.

מאמר הנאנו המקורי ויישום הבטא הראשון פורסמו בדצמבר 2014, מה שהופך אותו לאחד המטבעות הקריפטוגרפים הראשונים המבוססים על גרף מכוון ללא מעגלים (DAG) [6]. מיד לאחר מכן, מטבעות קריפטוגרפים מבוססי DAG התחילו בתהליכי פיתוח. הנודעים מביניהם הם דאגקוין/בייטבול ואיוטה [7] [8]. המטבעות הקריפטוגרפים מבוססי הדאג הללו שברו את תבנית הבלוקצ'יין ושיפרו ביצועי מערכת ואבטחה. בייטבול משיג הסכמה על ידי הסתמכות על "שרשרת מרכזית" המורכבת מעדים אמינים ובעלי מוניטין גבוה בזמן שאיוטה



כאשר לא מזוהה התנגשות, אין צורך בתוספת של תקורה. (א)

במקרה של פעולות מתנגשות, צמתים מצביעים לפעולה המאושרת. (ב)

איור 1: נאנו לא דורש תוספת תקורה לפעולה סטנדרטית. במקרה של פעולות מתנגשות, צמתים חייבים להצביע לפעולה שעליהם ישמרו.

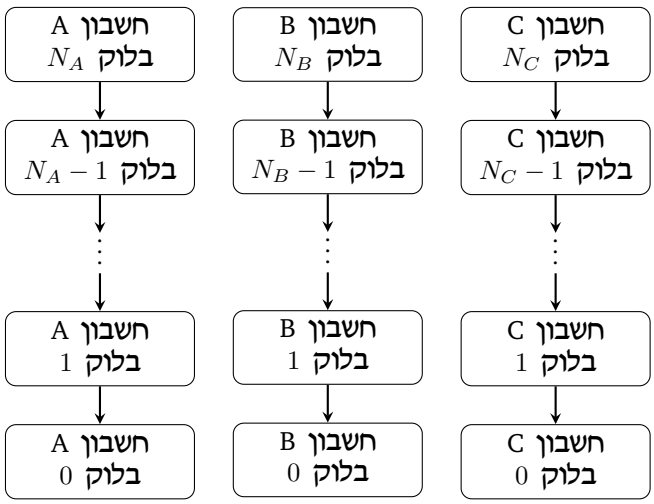
(IOTA) משיג הסכמה על ידי מערכת הוכחת עבודה על פעולות מצטברות. נאנו משיג הסכמה על ידי הצבעה מאוזנת משקל על פעולות מתנגשות. מערכת הסכמה זו מאפשרת פעולות מהירות והחלטיות יותר ובו זמנית שומרת על מערכת מבוזרת וחזקה. נאנו ממשיך פיתוח זה ומיקם את עצמו כאחד מהמטבעות הקריפטוגרפים בעלי הביצועים הגבוהים ביותר.

III. רכיבי נאנו

לפני שנתאר את ארכיטקטורת הנאנו הכללית, נגדיר רכיבים יחידים שמרכיבים את המערכת.

א'. חשבון

חשבון הוא החלק של המפתח הפומבי בחתימה דיגיטלית מבוססות זוג מפתחות. המפתח הפומבי, אשר מכונה גם כהכתובת, משותף עם שאר משתתפי הרשת כאשר המפתח הפרטי נשמר בסוד. פקטה חתומה דיגיטלית של מידע מבטיחה שתוכנה אושר על ידי מחזיק המפתח הפרטי. למשתמש אחד יכול להיות הרבה חשבונות, אבל קיימת כתובת פומבית אחת לכל חשבון.



איור 2: לכל חשבון יש בלוקצ'יין משלו המכיל את הסטוריית המאזן של החשבון. בלוק 0 חייב להיות פעולת פתיחה (חלק ב'־IV)

ב'. בלוק/פעולה

נשתמש במושג "בלוק" ו"פעולה" לסירוגין כאשר בלוק מכיל פעולה אחת. הפעולה מתייחסת לפעולת העברת התשלום עצמה כאשר הבלוק מתייחס לקידוד הדיגיטלי של הפעולה. פעולות חתומות על ידי המפתח הפרטי אשר שייך לחשבון שבו הפעולה בוצעה.

ג'. פנקס חשבונות

פנקס החשבונות הכללי הוא קבוצת כל החשבונות כך שלכל חשבון יש שרשרת פעולות פרטית (איור 2). זהו רכיב מרכזי שנופל תחת קטגוריית החלפת הסכמת זמן־ריצה עם הסכמת זמן־ארכיטקטורה; כולם מסכימים על ידי בדיקת חתימה שרק בעל החשבון יכול לערוך את השרשרת שלו. זה הופך מבנה נתונים משותף למראית עין, פנקס חשבונות מבוזר, לקבוצה של מבני נתונים לא משותפים.

ד'. צומת

הוא חתיכת תוכנה הרצה על מחשב שפועל תחת פרוטוקול נאנו ומשתתף ברשת נאנו. התוכנה מנהלת את פנקס החשבונות ואת כל החשבונות שהצומת שולט בהם, אם קיימים כאלה. צומת יכול לאחסן את כל פנקס החשבונות או את ההסטוריה הגזומה המכילה רק את הבלוקים האחרונים עבור כל בלוקצ'יין של חשבון. כאשר מרימים צומת חדש, מומלץ לאשר את כל ההסטוריה ולגזום בצורה מקומית.

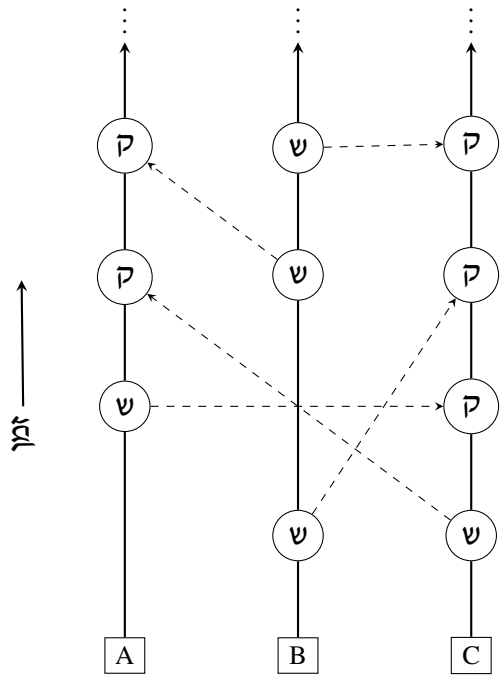
IV. סקירת מערכת

בניגוד לבלוקצ'יינים המשומשים במטבעות קריפטוגרפים אחרים, נאנו משתמשת במבנה של סריג־בלוקים. לכל חשבון יש בלוקצ'יין משלו (שרשרת־חשבון) המקבילה להסטוריית הפעולות/מאזן של החשבון (איור 2). כל שרשרת־חשבון יכולה להתעדכן רק על ידי בעל החשבון; דבר זה מאפשר לכל שרשרת־חשבון להתעדכן באופן מיידי וא־סינכרוני לשאר הסריג־בלוקים, מה שיוצר פעולות מהירות. הפרוטוקול של נאנו הוא קל משקל בצורה משמעותית; כל פעולה נכנסת בגודל המיניאלי הדרוש לפאקטת UDP לצורך העברה באינטרנט. דרישות החומרה לצמתים הן גם מינימאליות, מכיוון שצמתים צריכים רק להקליט ולשדר בלוקים לרוב הפעולות (איור 1).

המערכת מאותחלת עם חשבון ראשוני המכיל את המאזן הראשוני. המאזן הראשוני הוא בעל כמות קבועה ולעולם לא יוכל לגדול. המאזן הראשוני מחולק ונשלח לחשבונות

אחרים על ידי העברת פעולות הרשומות על השרשרת־חשבון של החשבון הראשוני. סכום המאזנים של כל החשבונות לעולם לא יעבור את מאזן החשבון הראשוני, דבר הנותן למערכת גבול עליון על כמות וללא יכולת להגדיל אותה.

חלק הזה יעבור על איך סוגי פעולות שונות נבנות ומועברות דרך הרשת.



איור 3: המחשה של סריג־הבלוקים. כל העברה של כספים דורשת בלוק שליחה (ש) ובלוק קבלה (ק), כאשר כל בלוק נחתם על ידי בעל שרשרת החשבון (A, B, C)

א'. פעולות

העברת כספים מחשבון אחד לאחר דורשת שתי פעולות, פעולת שליחה המפחיתה את הכמות ממאזן השולח ופעולת קבלה המוסיפה את הכמות למאזן המקבל (איור 3).

העברת כמויות כפעולות שונות בחשבונות השולח והמקבל משמשת עבור מספר מטרות:

1) סידור פעולות מתקבלות שהן א־סינכרוניות באופן טבעי.
2) שמירה על פעולות קטנות בכדי שיוכלו להתאים לפאקטת UDP.
3) מאפשרת גיזום פנקס החשבונות על ידי הקטנת עקבת המידע.
4) בידוד של פעולות מאושרות מפעולות שאינן מאושרות.

יותר מחשבון אחד שמעביר לאותו חשבון יעד היא פעולה א־סינכרונית. זמן השהיית הרשת והעובדה שהחשבונות המקבלים לא בהכרח בתקשורת אחד עם השני אומר שאין דרך אוניברסלית מקובלת לדעת איזה פעולה התבצעה קודם. מכיוון שחיבור הינה פעולה אסוציאטיבית, סדר קבלת הפעולות אינו משנה, ולכן אנחנו פשוט צריכים הסכמה כללית. זהו מרכיב עיקרי שמעביר הסכמה בזמן־ריצה להסכמה בזמן־ארכיטקטורה. לחשבון המקבל יש שליטה על ההחלטה איזה העברה הגיעה קודם והיא מבוטאת על ידי סדר הבלוקים המתקבלים החתומים.

אם חשבון רוצה לבצע העברה גדולה שהתקבלה כקבוצת העברות קטנות, נרצה להציג זאת בדרך שמתאימה לפקטת UDP אחת. כאשר חשבון מקבל מסדר העברות שהתקבלו, הוא שומר על סכום מאזן החשבון שלו כך שבכל זמן נתון, יש לו את היכולת להעביר כל כמות בעזרת פעולה בגודל קבוע. דבר זה שונה ממבנה פעולות יוצאות/נכנסות אשר משומש בביטקוין ובמטבעות קריפטוגרפים אחרים.

חלק מהצמתים אינם מעוניינים בהרחבת משאבים לצורך שמירה על הסטורית הפעולות המלאה של חשבון; הם רק מעוניינים במאזן הנוכחי של כל חשבון. כאשר חשבון מבצע פעולה, הוא מקודד את המאזן הנצבר שלו וצמתים אלו רק צריכים לעקוב אחרי הבלוק האחרון, אשר מאפשר להם להתעלם מהיסטוריית המידע ועדיין לשמור על נכונות.

אפילו עם דגש על הסכמי זמן־ריצה, ישנה השהייה כאשר מאשרים פעולות בגלל הצורך לזיהוי וטיפול בגורמים זדוניים ברשת. מכיוון שהסכמים בנאנו מגיעים מהר, בסדר של מילי־שניות עד שניות, נוכל להציג למשתמש שני קטגוריות מוכרות של פעולות מגיעות: מאושרות ולא מאושרות. פעולות מאושרות הן פעולות שבהן חשבון יצר בלוקי קבלה. פעולות לא מאושרות עדיין לא הוטמעו במאזן של מקבל הפעולה. זהו תחליף לאמות המדידה היותר מסובכות והלא מוכרות שבהן מטבעות אחרים משתמשים.

ב'. יצירת חשבון

כדי ליצור חשבון, צריך להוציא פעולת פתיחה (איור 4). פעולת פתיחה היא תמיד הפעולה הראשונה בכל שרשרת־חשבון ויכולה להיווצר בקבלה הראשונה של כספים. שדה ה־ACCOUNT שומר את המפתח הפומבי (הכתובת) שנוצר מתוך המפתח הפרטי שמשומש בחתימה. שדה ה־SOURCE מכיל את הגיבוב של הפעולה ששלחה את הכספים. בזמן יצירת חשבון, נציג חייב להיבחר בשביל להצביע בשמך; ניתן לשנות נציג בשלב מאוחר יותר (חלק ו'־IV). החשבון יכול להכריז על עצמו כנציג של עצמו.

```
open {
   account: DC04354B1...AE8FA2661B2,
   source: DC1E2B3F7C...182A0E26B4A,
   representative: xrb_1anr...posrs,
   work: 0000000000000000,
   type: open,
   signature: 83B0...006433265C7B204
}
```

איור 4: אנטומיה של פעולת פתיחה

ג'. מאזן חשבון

מאזן החשבון נשמר בתוך פנקס החשבונות עצמו. במקום לשמור את סכומי הפעולות, אישור (חלק ט'־IV) דורש בדיקה של ההפרש בין המאזן בבלוק השליחה למאזן בבלוק הקודם לו. החשבון המקבל יוכל להגדיל את את המאזן הקודם לתוך המאזן הסופי שניתן בבלוק הקבלה החדש. זה נעשה בכדי לשפר את מהירות העיבוד כאשר מורידים סכומים גדולים של בלוקים. כאשר מבקשים את הסטורית החשבון, סכומים ניתנים מיידית.

ד'. שליחה מחשבון

כדי לשלוח מכתובת, לכתובת חייבת להיות בלוק פתיחה קיים, ולכן מאזן (איור 5). שדה ה־PREVIOUS מכיל את הגיבוב של הבלוק הקודם בשרשרת־החשבון. שדה ה־DESTINATION מכיל את החשבון לשליחת הכספים. בלוק שליחה אינו ניתן לשינוי ברגע שהוא מאושר. ברגע שהבלוק משודר אל הרשת, הכספים מופחתים מיידית מהמאזן של חשבונות השולחים ומחכים כ"ממתינים" עד שהצד המקבל חותם על בלוק שמקבל את הכספים. כספים ממתינים לא נחשבים ככספים ה"מחכים לאישור" מכיוון שמצד השולחים, הכספים האלו נעלמו ולא יכולים לחזור.

```
send {
   previous: 1967EA355...F2F3E5BF801,
   balance: 010a8044a0...1d49289d88c,
   destination: xrb_3w...m37goeuufdp,
   work: 0000000000000000,
   type: send,
   signature: 83B0...006433265C7B204
}
```

איור 5: אנטומיה של פעולת שליחה

ה'. קבלת פעולה

כדי להשלים פעולה, המקבל של הכספים חייב ליצור בלוק קבלה בחשבון־השרשרת שלו (איור 6). שדה ה־SOURCE מצביע על גיבוב בלוק השליחה המתאים. ברגע שבלוק זה נוצר ומשודר לרשת, מאזני החשבונות מעודכנים והכספים עברו באופן רשמי לחשבון המקבל.

```
receive {
   previous: DC04354B1...AE8FA2661B2,
   source: DC1E2B3F7C6...182A0E26B4A,
   work: 0000000000000000,
   type: receive,
   signature: 83B0...006433265C7B204
}
```

איור 6: אנטומיה של פעולת קבלה

ו'. בחירת נציג

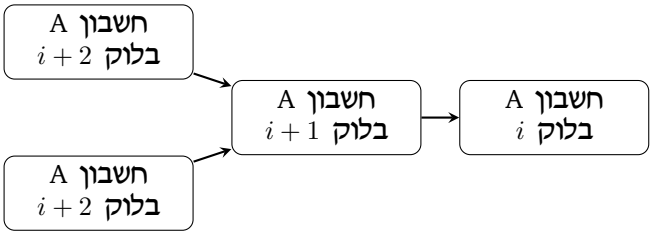
העובדה שמחזיקי חשבונות יכולים לבחור נציג שיצביע בשמם היא כלי ביזור חזק שאין לו השוואה בפרוטוקולי הוכחת עבודה והוכחת החזקה. במערכות הוכחת החזקה קונבנציונאליות, צומת בעל החשבון חייבת לרוץ כדי להשתתף בהצבעה. הרצה של צומת באופן מתמשך אינה פרקטית להמון משתמשים; נתינת הכוח להצביע לנציג בשם החשבון מרגיעה דרישה זו. לבעלי החשבון ניתנת האפשות להעביר הסכמה לכל חשבון בכל רגע נתון. פעולת שינוי משנה את נציג החשבון על ידי הורדת משקל ההצבעה מהנציג הקודם והעברתו לנציג החדש (איור 7). כספים אינם מועברים בפעולה זו ולנציג אין שליטה על הכספים של מחזיק החשבון.

```
change {
   previous: DC04354B1...AE8FA2661B2,
   representative: xrb_1anrz...posrs,
   work: 0000000000000000,
   type: change,
   signature: 83B0...006433265C7B204
}
```

איור 7: אנטומיה של פעולת שינוי

ז'. פיצולים והצבעות

פיצול קורה כאשר j בלוקים חתומים $b_1, b_2, \ldots, b_j$ טוענים שאותו בלוק הוא הקודם (איור 8). בלוקים אלו גורמים למצב קונפליקט על הסטאטוס של החשבון וחייבים להיות מטופלים. רק לבעל החשבון יש את היכולת לחתום על בלוקים לתוך השרשרת־חשבון שלו, לכן פיצול חייב להיות תוצאה של תכנות לקוי או כוונה זדונית (בזבוז כספים כפול) של בעל החשבון.



איור 8: פיצול קורה כאשר שניים (או יותר) בלוקים חתומים מצביעים לאותו בלוק קודם. בלוקים ישנים יותר נמצאים מימין. בלוקים חדשים יותר נמצאים משמאל

בעת גילוי, נציג יצור הצבעה עם קישור לבלוק $\hat{b}_i$ בפנקס החשבונות שלו וישדר אותה לרשת. משקל ההצבעה של הצומת, w_i הוא סכום המאזנים של כל החשבונות שבחרו בו להיות נציג. הצומת תעקוב אחרי הצבעות מגיעות משאר M נציגים ותשמור סכום מצטבר ל4 תקופות הצבעה, סך הכל כדקה אחת, ותאשר את הבלוק המנצח (משוואה 1).

$$v(b_j) = \sum_{i=1}^{M} w_i \mathbb{1}_{\hat{b}_i = b_j} \tag{1}$$

$$b^* = \arg\max_{b_j} v(b_j) \tag{2}$$

לבלוק הפופולארי ביותר b^* יהיה את מרבית הקולות והוא ישמר בפנקס החשבונות של הצומת (משוואה 2). הבלוק(ים) שהפסידו את ההצבעה ייעלמו. אם נציג מחליף בלוק בפנקס החשבונות שלו, הוא יצור הצבעה חדשה עם מספר רצף גבוה יותר וישדר את ההצבעה החדשה אל הרשת. זהו התרחיש היחידי בו נציג מצביע.

במקרים מסוימים, בעיות תקשורת קצרות יכולות למנוע את קבלת הבלוק ששודר על ידי שאר השותפים ברשת. משתתפים ברשת שלא ראו את השידור הראשוני יתעלמו מכל בלוק נלווה אחר בחשבון. שידור מחדש של הבלוק

אל הרשת יתקבל על ידי שאר משתתפי הרשת ובלוקים אחרים יוחזרו באופן אוטומאטי. אפילו במצב של פיצול או בלוק חסר, רק החשבונות המשוייכים לפעולה מושפעים; שאר הרשת ממשיכה בעיבוד פעולות לשאר החשבונות.

ח'. הוכחת עבודה

לכל ארבעת סוגי הפעולות יש שדה Work שחייב להיות מאוכלס. שדה ה־Work מאפשר ליוצר הפעולה לחשב מספר חד פעמי (Nonce) כך שהגיבוב של מספר זה ביחד עם שדה ה־Previous בפעולות קבלה/שליחה/שינוי או שדה ה־Account בפעולת פתיחה הינו מתחת לסף מסוים. בניגוד לביטקוין, הוכחת העבודה בנאנו הינה בשימוש כמנגנון נגד ספאם, בדומה לHashcash, וניתן לחשבו בקנה מידה של שניות [9]. ברגע שפעולה נשלחת, הוכחת העבודה של בלוקים עוקבים יכולה להיות מחושבת מראש מכיוון ששדה ה־Previous כבר ידוע; דבר זה יגרום לפעולות להופיע באופן מיידי למשתמש הקצה כל עוד הזמן בין הפעולות יותר גדול מהזמן שנדרש כדי לחשב את הוכחת העבודה.

ט'. אישור פעולה

כדי שבלוק יהיה נחשב כתקף, חייב להיות לו את התכונות הבאות:

1) הבלוק לא יכול להיות קיים בפנקס החשבונות (פעולה כפולה).
2) חייב להיות חתום על ידי בעל החשבון.
3) הבלוק הקודם הוא ראש הבלוק של השרשרת־חשבון. אם הוא קיים אבל הוא לא הראש, ישנו פיצול.
4) לחשבון חייב להיות בלוק פתיחה.
5) הגיבוב המחושב עובר את הסף שנדרש על ידי הוכחת העבודה.

אם מדובר בבלוק קבלה, נבדוק אם גיבוב בלוק המקור הוא בסטאטוס ממתין, כלומר עדיין לא אושר. אם מדובר על בלוק שליחה, המאזן חייב להיות פחות מהמאזן של הבלוק הקודם.

V. כיווני התקפה

נאנו, כמו כל מטבע קריפטוגרפי מבוזר, יכול להיות תחת התקפה של גורמים זדוניים למטרות רווח כלכלי או השבתת המערכת. בחלק זה נציין תרחישי התקפה אפשריים, את השלכותיהן של כל התקפה ואיך הפרוטוקול של נאנו לוקח אמצעי הגנה.

א'. סינכרון של קפיצת בלוק

בחלק ז'־IV , דיברנו על תרחיש בו יכול להיות שבלוק לא שודר כהלכה, ובכך גורם לרשת להתעלם מבלוקים עוקבים. אם צומת מבחינה בבלוק שאין לו קישור לבלוק הקודם לו, ישנן שתי אופציות:

1) להתעלם מהבלוק מכיוון שהוא יכול להיות בלוק זבל זדוני.
2) לבקש בקשה לסינכרון מחדש עם צומת אחר.

במקרה של סינכרון מחדש, חיבור TCP חייב להיווצר עם עוד צומת חדש בכדי להתמודד עם כמות התעבורה הגדלה שפעולת סינכרון מחדש דורשת. מצד שני, אם הבלוק אכן היה בלוק זדוני, אז פעולת הסינכרון מחדש היא מיותרת וגורמת לתעבורה גדולה יותר ללא כל צורך. זהו התקפת רשת שגורמת לדחייה של שירות (DoS).

בכדי להימנע מסינכרונים מחדש מיותרים, צמתים יחכו עד שסף מסוים של קולות נצפה בשביל בלוק שיכול להיות זדוני לפני שהם יתחילו חיבור לצומת חדש בשביל סינכרון. אם הבלוק לא משיג מספיק קולות, אפשר להחשיבו כמידע זבלי.

ב'. הצפת פעולות

יישות זדונית יכולה לשלוח המון פעולות תקפות אך מיותרות בין חשבונות תחת שליטתה בכדי להרוות את הרשת. מכיון שאין עמלת העברה, אותה יישות יכולה להמשיך את ההתקפה הזו ללא סוף. לעומת זאת, הוכחת העבודה הנדרשת בכל פעולה מגבילה את רצף הפעולות שבה היישות הזדונית יכולה לייצר מבלי להשקיע כספית במשאבי מחשוב. אפילו תחת התקפה כזו שמטרתה להציף את פנקס החשבונות, צמתים שהם לא צמתים עם הסטוריה מלאה יכולים לגזום פעולות עבר מהשרשרת. זה מגביל את השימוש באחסון מסוג זה של מתקפה לכמעט כל המשתמשים.

ג'. התקפת סיביל

יישות יכולה להרים מאות צמתים של נאנו על אותה מכונה. לעומת זאת, מכיוון שמערכת ההצבעות היא משוקללת על פי מאזן חשבון, הוספה של צמתים חדשים ברשת לא תוסיף כח הצבעה לתוקף. לכן אין שום תועלת בביצוע התקפת סיביל.

ד'. התקפת פני־ספנד

מתקפת פני־ספנד היא מתקפה בה התוקף מבזבז כמויות אינפיניטיסימליות למספר רב של חשבונות בכדי לבזבז את משאבי האחסון של צמתים. קצב פרסום הבלוקים מוגבל על ידי הוכחת העבודה, דבר שמגביל את יצירת החשבונות והפעולות לדרגה מסויימת. צמתים שהם לא צמתים בעלי היסטוריה מלאה יכולים לגזום חשבונות מתחת למדידה סטטיסטית שבה החשבון הוא ככל הנראה לא חשבון תקף. לסיום, נאנו מכוונת לשימוש מינימאלי של אחסון קבוע כך שהגודל הנדרש לשמירה על עוד חשבון הוא יחסי לגודל של בלוק פתוח + מפתוח = $96B + 32B = 128B$. זהו שווה ערך לקח של1GB יכול לאכסן 8 מליון חשבונות פני־ספנד. אם צמתים רוצים לגזום בצורה אגרסיבית יותר, הם יכולים לחשב הפצה שמבוססת על פי כמות כניסות לחשבון ולהעביר חשבונות שאליהם הכניסה נמוכה לאחסון איטי יותר.

ה'. התקפת הוכחת עבודה מחושבת מראש

מכיוון שבעל החשבון הינו היישות היחידה שמוסיפה בלוקים לשרשרת החשבון שלו, בלוקים עוקבים יכולים להיות מחושבים, ביחד עם הוכחת העבודה שלהם, לפני שהם משודרים לרשת. כאן התוקף מייצר מספר עצום של בלוקים עוקבים, שבו לכל אחד ערך מינימאלי, במשך תקופה מסויימת של זמן. בשלב מסוים, התוקף מבצע השבתת שירות (DoS) על ידי הצפת הרשת עם המון פעולות תקפות. זוהי גרסא מתקדמת של הצפת הפעולות בחלק ב'־V . כזו התקפה תעבוד רק באופן זמני קצר, אבל

יכולה להיות משומשת ביחד עם עוד התקפות כגון התקפת 50%< (חלק ו'-V) בכדי להגדיל את הצלחת ההתקפה. הגבלת רצף הפעולות ועוד טכניקות כרגע נחקרות בכדי להתמודד עם התקפה זו.

ו'. התקפת 50%<

מערכת ההסכמה של נאנו היא מערכת הצבעות מאוזנת משוקללת. אם תוקף יכול להשיג 50%< של כוח ההצבעה, הוא יכול לגרום לרשת להתעלם מהסכמה ובכך לשבור את המערכת. תוקף יכול להוריד את הכמות של המאזן שהוא צריך להפסיד על ידי חסימה של צמתים טובים מלהצביע דרך שיבוש מערכת (DoS). נאנו נוקטת באמצעים הבאים בשביל להתמודד עם התקפה זו:

1) ההגנה המרכזית נגד סוג זה של התקפה היא העובדה שמשקל ההצבעה שווה להשקעה במערכת. בעל חשבון מתומרץ באופן טבעי לשמור על אמינות המערכת כדי להגן על השקעתו. נסיון להפוך את פנקס החשבונות יגרום להרס המערכת כולה, מה שיהרוס את השקעתו.
2) מחיר ההתקפה הוא יחסי לשווי השוק של נאנו. במערכות הוכחת עבודה, טכנולוגיה יכולה להיות מומצאת כך שתינתן שליטה לא יחסית בהשוואה להשקעה הכספית ואם ההתקפה מצליחה, הטכנולוגיה יכולה לחזור להיות בשימוש אחרי שההתקפה נגמרת. עם נאנו, מחיר התקפת המערכת גדל עם המערכת עצמה ואם התקפה תהיה מוצלחת, ההשקעה בהתקפה לא יכולה לחזור לבעלים.
3) בכדי לשמור על קוורום מקסימלי של מצביעים, הצעד הבא של ההגנה הוא הצבעה על ידי נציגים. בעלי חשבון שאינם יכולים להשתתף בהצבעות באופן אמין מסיבות של של חוסר חיבור לרשת יכולים לבחור נציג שישתתף בהצבעה עם משקל המאזן שלהם. בצעד זה אנחנו ממקסמים את מספר הנציגים ואת גיוונם ומחזקים את הרשת.
4) פיצולים בנאנו אף פעם לא קורים בטעות, אז צמתים יכולים להחליט איך לתקשר עם צמתים מפוצלים. הזמן היחיד בו חשבונות של משתמש לא תוקף חשופים לבלוקים מפוצלים זה אם הם מקבלים מאזן מחשבון תוקף. חשבונות שרוצים להיות בטוחים מבלוקים מפוצלים יכולים לחכות קצת או המון לפני שהם מקבלים מחשבון שמייצר בלוקים מפוצלים או שהם יכולים לבחור לא לקבל לעולם. מקבלים גם יכולים לבחור ליצור חשבונות נפרדים כאשר הם מקבלים כספים ממקורות מפוקפקים בכדי לבודד חשבונות אחרים.
5) קו הגנה אחרון שעדיין לא נכנס לשימוש הוא מלוט בלוקים (BLOCK CEMENTING). נאנו הולך מעל ומעבר בכדי ליישב פיצול בלוקים בצורה מהירה דרך הצבעות. צמתים יוכלו להיות מקונפגים לבלוקי בטון, דבר הימנע מהם לחזור חזרה אחרי תקופה מסויימת של זמן. הרשת היא מספיק בטוחה דרך התמקדות בזמן העברה מהיר בכדי למנוע פיצולים מעורפלים.

גרסא יותר מתוחכמת למתקפה של 50%< מתוארת באיור 9. "מנותקים" הוא אחוז הנציגים שנבחרו אך לא מחוברים לרשת בכדי להצביע. "החזקה" היא כמות ההשקעה שאיתה התוקף מצביע. "פעילים" היא כמות הנציגים שמחוברים ומצביעים בהתאם לפרוטוקול. תוקף יוכל להוריד את כמות ההחזקה שהם צריכים לוותר עליה על ידי הוצאת מצביעים אחרים מהרשת דרך התקפת DoS. אם התקפה זו יכולה להחזיק, הנציגים שמותקפים יהפכו להיות בלתי מסונכרנים וזה מומחש על ידי "אסנכרון". לסיום, תוקף יכול להשיג קפיצה קצרה של כוח הצבעה על ידי החלפת התקפת הDoS שלו בקבוצה חדשה של נציגים בזמן שהקבוצה הישנה מסנכרנת מחדש את פנקס החשבונות שלה. זהו מומחש על ידי "התקפה".

מנותקים	אסכנרון	התקפה	פעילים	החזקה

איור 9: סידור פוטנציאלי של הצבעה שיכול להוריד 51% מדרישות התקפה.

אם תוקף יכול לגרום למצב בו "החזקה" >"פעילים" על ידי שילוב של כל התרחישים הללו, הוא יוכל להפוך קולות בהצלחה בפנקס החשבונות בתמורה להחזקה שלו. נוכל להעריך כמה סוג זה של התקפה יעלה על ידי בחינה של שווי השוק של מערכות אחרות. אם נניח ש-33% מהנציגים מנותקים מהרשת או מותקפים על ידי DoS, תוקף יצטרך לרכוש 33% משווי השוק בכדי לתקוף את המערכת דרך הצבעה.

ז'. הרעלת אתחול

ככל שתוקף מחזיק מפתח פרטי ישן עם מאזן יותר זמן, כך גדל הסיכוי שלמאזן שהיה קיים באותו הזמן לא יהיה נציגים משתתפים בגלל שהמאזן או הנציגים עברו לחשבון חדש יותר. זה אומר שאם צומת מאותחל לייצוג ישן של הרשת איפה שלתוקף יש קוורום של כוח הצבעה בהשוואה לנציגים באותה נקודה בזמן, התוקף יוכל להתעלם מהחלטת הבחירה של צומת זו. אם משתמש חדש זה רוצה לתקשר עם כל אחד חוץ מהצומת התוקף כל הפעולות שלו יידחו בגלל שיש להם ראשי בלוקים שונים. התוצאה הסופית היא שצמתים יכולים לבזבז את זמנם של צמתים אחרים ברשת על ידי שליחה של מידע רע. בכדי למנוע את זה, צמתים יכולים להיות משוייכים למסד נתונים התחלתי של חשבונות ידועים עם ראשי בלוקים טובים; זהו תחליף להורדת כל מסד הנתונים עד לבלוק הראשוני. ככל שההורדה קרובה יותר לזמן הנוכחי, כך גבוה יותר הסיכוי של הגנה מוצלחת מפני ההתקפה. בסוף, התקפה זו היא כנראה לא יותר גרועה משליחת מידע זבלי לצמתים ברגעי האתחול, מכיוון שהם לא יוכלו לייצר פעולות עם כל אחד שיש לו מסד נתונים עכשוי.

VI. יישום

נכון להיום, היישום המדובר מיושם בC++ ונמצא בGITHUB מאז 2014 [10].

א'. מאפייני פיתוח

יישום הנאנו עומד בתקן הארכיטקטורה המתוארת במאמר זה. מפרטים נוספים מתוארים כאן:

1) אלגוריתם חתימה:

נאנו משתמש באלגוריתם עקומה אליפטית ED25519 מותאם עם גיבוב BLAKE2B עבור כל החתימות הדיגיטליות [11]. ED25519 נבחר מכיוון שהוא מאפשר חתימה מהירה, וידוא מהיר ואבטחה גבוהה.

2) אלגוריתם גיבוב:
מכיוון שאלגוריתם הגיבוב משומש רק בכדי למנוע ספאם ברשת, בחירת האלגוריתם פחות חשובה בהשוואה למטבעות קריפטוגרפיים מבוססי כרייה. היישום שלנו משתמש ב־BLAKE2B כאלגוריתם עיכול נגד תכני בלוקים. [12].
3) פונקציית גזירת מפתח:
בארנק המדובר, מפתחות מוצפנים על ידי סיסמא והסיסמא עוברת דרך פונקציית גזירת מפתח בכדי להגן נגד מכונות המותאמות לפריצה (ASIC). נכון להיום Argon2 [13] הוא המנצח בתחרות הפומבית היחידה המכוונות ליצירת פונקציית גזירת מפתח עמידה.
4) זמן השהייה של בלוקים:
מכיוון שלכל חשבון יש בלוקצ'יין משלו, עדכונים יכולים להתבצע באופן א־סינכרוני למצב הרשת. לכן אין זמני השהייה ופעולות מפורסמות באופן מיידי.
5) פורטוקול הודעות UDP:
המערכת שלנו מתוכננת לפעול באופן בלתי מוגבל עם שימוש מינימאלי של כוח מחשוב ככל שאפשר. כל ההודעות במערכת עוצבו להיות חסרות מצב ולהתאים בפאקטת UDP אחת. דבר זה מאפשר למשתתפים לא כבדים עם חיבור חלש לאינטרנט להשתתף ברשת מבלי ליצור חיבורי TCP קצרי טווח. TCP משומש אך ורק עבור משתתפים חדשים כאשר הם רוצים לייצר את הבלוקצ'יינים באופן מהיר ובכמות גדולה בבת אחת.
צמתים יכולים להיות בטוחים שהפעולות שלהם התקבלו ברשת על ידי צפייה בפעולות ששודרו מצמתים אחרים מכיוון שהם יוכלו לראות מספר העתקים החוזרים לעצמם.

ב'. IPv6 ומולטיקאסט

בנייה על הפרוטוקול חסר החיבוריות UDP מאפשר ליישומים עתידיים להשתמש בIPv6 מולטיקאסט כתחליף להצפת פעולות רגילה ולשדר הצבעות. דבר זה יפחית את רוחב הפס של הצריכה ברשת ויאפשר יותר גמישות לצמתים בעתיד.

ג'. ביצועים

בזמן הכתיבה הנוכחי, 4.2 מליון פעולות עובדו על ידי רשת הנאנו, מה שיצר בלוקצ'יין בגודל 1.7GB. זמני פעולות נמדדות בשניות. התייחסות נוכחית ליישום המתבצע על SSDs פשוט יכול לעבד 10,000 פעולות בשניה ומוגבל בגבול עליון בעיקר מקלט ופלט.

VII. שימוש במשאבים

זהו מעבר על משאבים המשומשים על ידי צומת נאנו. בנוסף, אנחנו עוברים על רעיונות להפחתת השימוש במשאבים במקרים ספציפיים. צמתים מופחתי משאבים בדרך כלל יקראו קלים, גזומים או צמתי וידוא תשלום מופשט (SPV).

א'. רשת

כמות פעילות הרשת תלויה בכמה הרשת תורמת לבריאות הרשת.
1) נציג:
צומת נציג דורש משאבי רשת מקסימלים מכיוון שהוא צופה בתעבורת הצבעות מנציגים אחרים ומפרסם את ההצבעה שלו.
2) חסר אמון:
צומת חסר אמון דומה לצומת נציג אבל הוא רק צופה, הוא לא מכיל בתוכו מפתח פרטי של חשבון נציג ולא מפרסם הצבעות משלו.
3) בעל אומן:
צומת בעל אמון צופה בתעבורת הצבעות מצומת נציג אחד שבו הוא בוטח שיבצע הסכמה בצורה נכונה. דבר זה מפחית את כמות התעבורת הצבעות מנציגים המגיעים לצומת זה.
4) קל:
צומת קל הוא גם צומת בעל אמון שרק צופה בתעבורה עבור חשבונות שבהם הוא מעוניין לאפשר תעבורת רשת מינימלית.
5) התחלתי:
צומת התחלתי מגיש חלקים מפנקס החשבונות או את כולו לצמתים שמתחברים לרשת. זה נעשה על ידי חיבור TCP ולא UDP מכיוון שזה מאפשר כמות גדולה יותר של מידע הנדרש בכדי לבצע בקרת זרימה מתקדמת.

ב'. גודל הדיסק

תלוי בדרישות המשתמש, הגדרות שונות לצמתים דורשות דרישות אחסון שונות.
1) היסטורי:
צומת המעוניין לשמור את כל ההיסטוריה של הפעולות ידרש לספק את כמות האחסון המקסימלית.
2) עכשווי:
עקב הארכיטקטורה שמטרתה לשמור חשבונות מצטברים עם בלוקים, צמתים צריכים לשמור אך ורק את הבלוקים האחרונים או בלוקי הראש עבור כל חשבון בכדי להשתתף בהסכמה. אם צומת לא מעוניין לשמור את כל ההיסטוריה הוא יכול לשמור רק את בלוקי הראש.
3) קל:
צומת קל לא שומר מידע מפנקס החשבונות בצורה מקומית. הוא אך ורק משתתף ברשת על ידי צפייה בפעילות על חשבונות שבהם הוא מעוניין או בכדי ליצור פעולות חדשות עם מפתחות פרטיים שבהם הוא מחזיק.

ג'. מעבד

1) יצירת פעולות:
צומת המעוניין לייצר פעולות חדשות מחוייב להפיק מספר חד פעמי בצורת הוכחת עבודה בכדי לעבור את מנגנון ההגנה של נאנו. חישוב של מבחר רכיבי חומרה מסופק בנספח א'.
2) נציג:
נציג מוכרח לאמת חתימות עבור בלוקים, הצבעות וגם להפיק חתימות משל עצמו בכדי להשתתף בהסכמה. כמות משאבי המעבד עבור צומת נציג היא קטנה משמעותית מיצירת פעולה ואמורה לעבוד עם כל מעבד יחיד במחשב בן זממנו.
3) צופה:
צומת צופה לא מייצר הצבעות משל עצמו. מכיוון שתקורת יצירת חתימות היא מינימלית, דרישות המעבד הן כמעט זהות לדרישות המעבד בהרצת צומת נציג.

VIII. מסקנה

במאמר זה הצגנו את המסגרת עבור מטבע קריפטוגרפי חסר עמלות, ללא צורך באמון ובעל זמן השהייה נמוך המשתמש במבנה נתונים חדשני בשם סריג־בלוקים ומנגנון הצבעה הנעזר בהוכחת החזקה בעזרת נציגים. הרשת דורשת משאבים מינימלים, לא דורשת חומרה חזקה לצורך כרייה ויכולה לייצר תפוקה גבוהה של פעולות. כל זה בר השגה על ידי בלוקצ'יינים פרטיים עבור כל חשבון, דבר המסיר בעיות גישה וחוסר יעילות עבור מבנה נתונים גלובלי. זיהינו כיווני התקפה אפשריים על המערכת והצגנו טיעונים נגדיים על איך נאנו חסין לצורות התקפה אלו.

נספח א'
אמות מדידה של החומרה הנדרשת להוכחת עבודה

כפי שהוזכר קודם, הוכחת העבודה בנאנו נועדה להפחתת הספאם ברשת. יישום הצומת שלנו מספק האצה המנצלת את היתרונות שיש ל־OpenCL על כרטיס מסך מתאים. טבלה I מספקת השוואה אמיתית בין אמות מדידה של מבחר רכיבי חומרה. נכון לעכשיו סף הוכחת העבודה הוא קבוע אבל סף משתנה עלול להיות מיושם כאשר כוח המחשוב הממוצע יגדל בעתיד.

טבלה I: אמות מדידה של החומרה הנדרשת להוכחת עבודה

מכשיר	פעולות בשנייה
Nvidia Tesla V100 (AWS)	6.4
Nvidia Tesla P100 (Google,Cloud)	4.9
Nvidia Tesla K80 (Google,Cloud)	1.64
AMD RX 470 OC	1.59
Nvidia GTX 1060 3GB	1.25
Intel Core i7 4790K AVX2	0.33
Intel Core i7 4790K,WebAssembly (Firefox)	0.14
Google Cloud 4 vCores	0.14־0.16
ARM64 server 4 cores (Scaleway)	0.05־0.07

תודות

נרצה להודות לבריאן פוג על סידור ועיצוב מאמר זה ולאמיר הגפני ורון הגפני על תרגומו לעברית.

ביבליוגרפיה*

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008. [Online]. Available: http://bitcoin.org/bitcoin.pdf
[2] "Bitcoin median transaction fee historical chart." [Online]. Available: https://bitinfocharts.com/comparison/bitcoin-median_transaction_fee.html
[3] "Bitcoin average confirmation time." [Online]. Available: https://blockchain.info/charts/avg-confirmation-time
[4] "Bitcoin energy consumption index." [Online]. Available: https://digiconomist.net/bitcoin-energy-consumption
[5] S. King and S. Nadal, "Ppcoin: Peer-to-peer crypto-currency with proof-of-stake," 2012. [Online]. Available: https://peercoin.net/assets/paper/peercoin-paper.pdf
[6] C. LeMahieu, "Raiblocks distributed ledger network," 2014.
[7] Y. Ribero and D. Raissar, "Dagcoin whitepaper," 2015.
[8] S. Popov, "The tangle," 2016.
[9] A. Back, "Hashcash - a denial of service counter-measure," 2002. [Online]. Available: http://www.hashcash.org/papers/hashcash.pdf
[10] C. LeMahieu, "Raiblocks," 2014. [Online]. Available: https://github.com/clemahieu/raiblocks
[11] D. J. Bernstein, N. Duif, T. Lange, P. Shwabe, and B.-Y. Yang, "High-speed high-security signatures," 2011. [Online]. Available: http://ed25519.cr.yp.to/ed25519-20110926.pdf
[12] J.-P. Aumasson, S. Neves, Z. Wilcox-O'Hearn, and C. Winnerlein, "Blake2: Simpler, smaller, fast as md5," 2012. [Online]. Available: https://blake2.net/blake2.pdf
[13] A. Biryukov, D. Dinu, and D. Khovratovich, "Argon2: The memory-hard function for password hashing and other applications," 2015. [Online]. Available: https://password-hashing.net/argon2-specs.pdf