ביטקוין: מערכת שיתופית לכסף אלקטרוני

סאטושי נאקאמוטו (Satoshi Nakamoto)
satoshin@gmx.com
www.bitcoin.org

תורגם על ידי: מני רוזנפלד
נערך על ידי: גיים נגוין

תקציר. גרסה שיתופית לחלוטין של כסף אלקטרוני תאפשר לשלוח תשלומים מקוונים ישירות מצד אחד לשני מבלי לעבור דרך מוסד כספי. חתימות דיגיטליות מהוות חלק מהפיתרון, אולם עיקר התועלת תאבד אם צד שלישי נאמן עדיין דרוש כדי למנוע ניצול כפול. אנו מציעים פיתרון לבעיית הניצול הכפול המשתמש ברשת שיתופית. הרשת מייצרת חותמות זמן עבור פעולות על ידי גיבובן לשרשרת מתמשכת של הוכחת עבודה, ובכך מנהלת רישום שאינו נתון לשינוי ללא שחזור הוכחת העבודה. השרשרת הארוכה ביותר משמשת לא רק כהוכחה של סדר האירועים שנצפו, אלא גם הוכחה שהיא הגיעה מהריכוז הגדול ביותר של כוח חישובי. כל עוד מרבית כוח החישוב נשלט על ידי צמתים שאינם משתפים פעולה כדי לתקוף את הרשת, הם ייצרו את השרשרת הארוכה ביותר ויעלו בקצב על תוקפים. הרשת עצמה דורשת מבנה מזערי. הודעות מופצות על בסיס מיטב המאמצים, וצמתים יכולים לעזוב את הרשת ולהצטרף אליה כרצונם, ולקבל את שרשרת הוכחת העבודה הארוכה ביותר כהוכחה של מה שקרה בזמן היעדרותם.

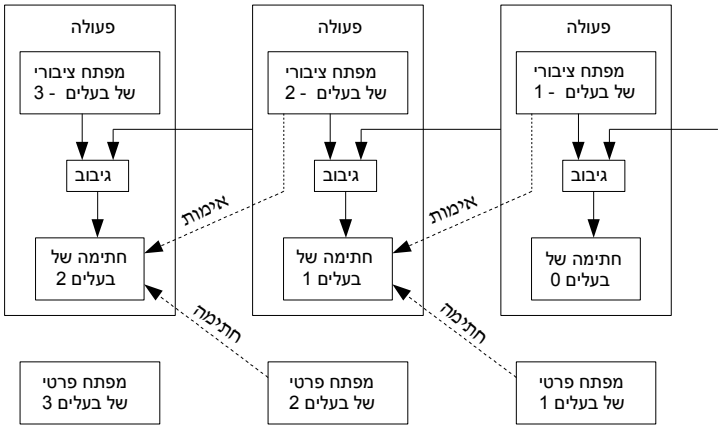
1. מבוא

מסחר באינטרנט כיום מסתמך באופן כמעט בלעדי על מוסדות כספיים המשמשים כצד שלישי נאמן לעיבוד תשלומים אלקטרוניים. המערכת עובדת באופן משביע רצון עבור רוב העסקאות, אך עם זאת, היא עדיין סובלת מהחולשות הטבועות במודל מבוסס האמון. פעולות שהן לחלוטין בלתי-ניתנות לביטול אינן באמת אפשריות, מכיוון שהמוסד הכספי אינו יכול להימנע מגישור מחלוקות. עלות הגישור מעלה את עלויות הפעולה, מה שמגביל את הגודל המעשי המזערי לעסקה ומונע את האפשרות לעסקאות אגביות קטנות, ויש עלות רחבה יותר באובדן היכולת לבצע תשלומים בלתי-ניתנים לביטול עבור שירותים בלתי-ניתנים לביטול. האפשרות לביטול מרחיבה את הצורך באמון. סוחרים חייבים להיזהר מלקוחותיהם, ולהטריח אותם לספק יותר מידע מאשר היה נחוץ אחרת. הם משלימים בלית ברירה עם אחוז מסוים של הונאה, הנתפש כבלתי-נמנע. ניתן להימנע מהעלויות האלו ומחוסר הוודאות בתשלומים על ידי שימוש במטבע גשמי, אך אין בנמצא מנגנון לבצע תשלומים על גבי ערוץ תקשורת ללא צד נאמן.

נחוצה מערכת תשלומים אלקטרוניים המבוססת על הוכחה קריפטוגרפית במקום אמון, המאפשרת לכל שני צדדים החפצים בדבר לבצע פעולה זה עם זה באופן ישיר ללא צורך בצד שלישי נאמן. פעולות שמבחינה חישובית אין זה מעשי להפוך אותן יגנו על מוכרים מפני הונאה, וניתן בקלות לממש מנגנוני השלשה שגרתיים בכדי להגן על קונים. במאמר זה, אנו מציעים פיתרון לבעיית הניצול הכפול המשתמש בשרת חותמות-זמן שיתופי מבוזר בשביל ליצור הוכחה חישובית של הסדר הכרונולוגי של פעולות. המערכת בטוחה כל עוד הצמתים הישרים שולטים יחד ביותר עוצמת חישוב מכל קבוצת צמתים תוקפים המשתפים פעולה.

2. פעולות

אנו מגדירים מטבע אלקטרוני כשרשרת של חתימות דיגיטליות. כל בעלים מעביר את המטבע לבא אחריו על ידי חתימת גיבוב (Hash) של הפעולה (transaction) הקודמת והמפתח הציבורי של הבעלים הבאים, והוספת אלו לסוף המטבע. מקבל תשלום יכול לאמת את החתימות בכדי לוודא את שרשרת הבעלות.

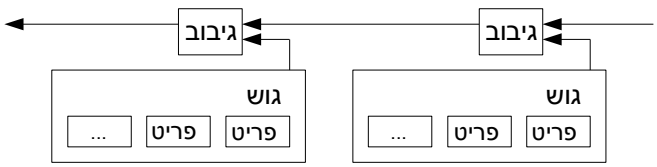


הבעיה היא כמובן שמקבל התשלום אינו יכול לודא שאף אחד מהבעלים לא שילם באמצעות המטבע פעמיים (ניצול כפול, double-spend). פיתרון נפוץ הוא להכניס סמכות מרכזית הזוכה לאמון, מטבעה, שבודקת כל פעולה כנגד ניצול כפול. לאחר כל פעולה, המטבע חייב לחזור אל המטבעה כדי להנפיק מטבע חדש, ורק על מטבעות שהונפקו ישירות מהמטבעה סומכים שלא עברו ניצול כפול. הבעיה בפיתרון זה הוא שהגורל של המערכת הכספית בכללותה תלוי בחברה המפעילה את המטבעה, היות וכל פעולה חייבת לעבור דרכם, בדיוק כמו בבנק.

אנו זקוקים לדרך בה מקבל התשלום יוכל לדעת שהבעלים הקודמים לא חתם על אף פעולה מוקדמת יותר. לצרכים שלנו, הפעולה המוקדמת ביותר היא הקובעת, כך שאיננו דואגים מנסיונות מאוחרים יותר לניצול כפול. הדרך היחידה לאשר את העדר פעולה היא להיות מודע לכל הפעולות. במודל מבוסס המטבעה, המטבעה מודעת לכל הפעולות ומחליטה איזו הגיעה קודם. כדי להשיג זאת ללא צד נאמן, חובה להכריז על פעולות באופן פומבי [1], ונחוצה מערכת להסכמה בין משתתפים לגבי היסטוריה יחידה של הסדר בו הן התקבלו. מקבל התשלום זקוק להוכחה שבזמן בו כל פעולה התבצעה, רוב הצמתים הסכימו כי היא היתה הראשונה שהתקבלה.

3. שרת חותמות-זמן

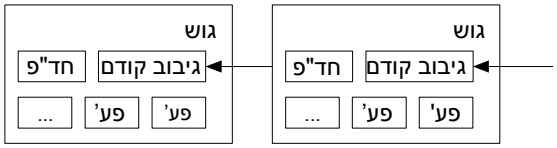
הפיתרון אותו אנו מציעים מתחיל עם שרת חותמות-זמן (timestamp server). שרת חותמות-זמן פועל על ידי לקיחת גיבוב של גוש (block) של פריטים הזקוקים לחותמת זמן והפצתו באופן נרחב, למשל בעיתון או ב-Usenet (ראה [2-5]). חותמת הזמן מוכיחה שבהכרח הנתונים היו קיימים באותו זמן, אחרת לא היו יכולים להיות כלולים בגיבוב. כל חותמת-זמן מכילה את חותמת-הזמן הקודמת בגיבוב שלה, מה שיוצר שרשרת בה כל חותמת-זמן נוספת מתגברת את אלו שבאו לפניה.



4. הוכחת עבודה

בכדי לממש שרת חותמות-זמן על בסיס שיתופי, אנו נצטרך מערכת הוכחת עבודה (Proof-of-work) בדומה ל-Hashcash של אדם בק [6], במקום פרסומים בעיתון או ב-Usenet. הוכחת העבודה כרוכה בחיפוש אחר ערך שכאשר הוא עובר גיבוב, למשל עם SHA-256, הגיבוב מתחיל במספר מסוים של סיביות אפס. הכמות הממוצעת של העבודה הדרושה היא מעריכית במספר סיביות האפס הדרושות, והיא יכולה להתאמת תוך ביצוע גיבוב בודד.

בשביל רשת חותמות-הזמן שלנו, אנו מממשים את הוכחת העבודה על ידי הגדלת מספר חד-פעמי (nonce) בגוש עד שנמצא ערך המעניק לגיבוב של הגוש את המספר הדרוש של סיביות אפס. לאחר שהמאמץ החישובי הושקע בשביל לגרום לגוש לספק את הוכחת העבודה, לא ניתן לשנות אותו מבלי לבצע את העבודה מחדש. מאחר וגושים מאוחרים יותר משורשרים אחריו, העבודה הדרושה כדי לשנות גוש תכלול ביצוע מחדש של כל הגושים שלאחריו.



הוכחת העבודה גם פותרת את בעיית קביעת הייצוג בקבלת החלטות מבוססת רוב. אם הרוב היה מבוסס על "קול לכל כתובת IP", כל מי שמסוגל להקצות כתובות רבות היה יכול לחתור תחתיו. הוכחת עבודה היא למעשה "קול לכל מעבד". החלטת הרוב מיוצגת על ידי השרשרת הארוכה ביותר, בה מושקע מאמץ הוכחת העבודה הרב ביותר. אם רוב עוצמת החישוב נשלטת על ידי צמתים ישרים, השרשרת הישרה תצמח בקצב המהיר ביותר ותשיג כל שרשרת מתחרה. בכדי לשנות גוש מהעבר, תוקף יצטרך לבצע מחדש את הוכחת העבודה של הגוש וכל הגושים לאחריו, ואז להדביק את הפער ולעלות על העבודה של הצמתים הישרים. מאוחר יותר נראה שההסתברות שתוקף איטי יותר ידביק את הפער קטנה באופן מעריכי ככל שגושים עוקבים מתווספים.

בכדי לפצות על הגדילה במהירות החומרה והשתנות ברמת העניין בהפעלת צמתים לאורך זמן, דרגת הקושי של הוכחת העבודה נקבעת על ידי ממוצע נע השם למטרה מספר ממוצע מסוים של גושים לשעה. אם הם נוצרים מהר מדי, דרגת הקושי עולה.

5. רשת

השלבים בהפעלת הרשת הם כדלקמן:

1) פעולות חדשות משודרות לכל הצמתים.
2) כל צומת אוסף פעולות חדשות לתוך גוש.
3) כל צומת עובד על מציאת הוכחת עבודה קשה לגוש שלו.
4) כשצומת מוצא הוכחת עבודה, הוא משדר אותו לכל הצמתים.
5) צמתים מסכימים עם הגוש רק אם כל הפעולות בו הן תקינות ולא נוצלו כבר.
6) צמתים מבטאים את הסכמתם עם גוש בעצם זה שהם עובדים על יצירת הגוש הבא בשרשרת, תוך שימוש בגיבוב של הבלוק שהתקבל בתור הגיבוב הקודם.

צמתים תמיד מתייחסים לשרשרת הארוכה ביותר כאל השרשרת הנכונה וימשיכו לעבוד על הארכתה. אם שני צמתים משדרים גרסאות שונות של הגוש הבא בו-זמנית, חלק מהצמתים יקבלו את האחד או השני תחילה. במקרה זה, הם יעבדו על הראשון שהם קיבלו, אבל ישמרו את הענף השני למקרה שהוא יהפך לארוך יותר. התיקו יישבר כשהוכחת העבודה הבאה תימצא ואחד מהענפים יהפך לארוך יותר; הצמתים שעבדו על הענף האחר יעברו לענף הארוך יותר.

שידורים של פעולות חדשות לא צריכים בהכרח להגיע לכל הצמתים. כל עוד הם מגיעים לצמתים רבים, הם ייכנסו לתוך גוש בחלוף זמן לא רב. שידורי גוש הם גם סובלניים לגבי הודעות שנשמטו. אם צומת לא מקבל גוש, הוא יבקש אותו כשהוא מקבל את הגוש הבא ומבין שהוא החמיץ אחד.

6. תמריץ

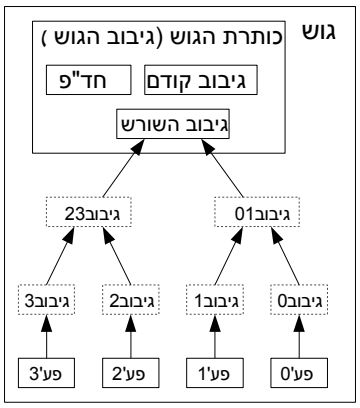
על פי המוסכמה, הפעולה הראשונה בכל גוש היא פעולה מיוחדת שמתחילה מטבע חדש בבעלותו של יוצר הגוש. זה מוסיף תמריץ לצמתים לתמוך ברשת, ומעניק דרך ראשונית להפיץ מטבעות למחזור, מאחר ואין סמכות מרכזית היכולה להנפיק אותם. התוספת הקבועה של מטבעות חדשים מקבילה לכורי זהב המוציאים משאבים כדי להוסיף זהב למחזור. במקרה שלנו, המשאבים הם זמן מעבד ואנרגיה חשמלית.

התמריץ יכול גם להיות ממומן על ידי עמלות פעולה. אם ערך הפלט של פעולה נמוך מערך הקלט, ההפרש הוא עמלת פעולה שמתווספת לערך התמריץ של הגוש המכיל את הפעולה. לאחר שמספר קבוע מראש של מטבעות נכנס למחזור, התמריץ יכול לעבור באופן בלעדי לעמלות פעולה ולהיות נקי לחלוטין מאינפלציה.

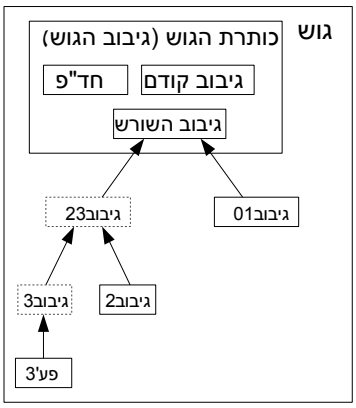
התמריץ יכול לעזור בעידוד צמתים להישאר ישרים. אם תוקף חמדן מסוגל לכנוס יותר כוח חישוב מכל הצמתים הישרים, הוא יצטרך לבחור בין שימוש בו כדי להונות אנשים בגניבת התשלומים שלו בחזרה, או שימוש בו ליצור מטבעות חדשים. מתקבל על הדעת שיהיה זה רווחי יותר בעיניו לשחק לפי הכללים, כללים אלו שחוננים אותו ביותר מטבעות חדשים מאשר כל שאר המשתתפים ביחד, מאשר לרופף את המערכת ואת תקפות ההון של עצמו.

7. שחרור נפח כונן

מהרגע בו הפעולה האחרונה במטבע קבורה תחת מספיק גושים, הפעולות המנוצלות לפניה יכולות להיזרק כדי לחסוך בנפח כונן. כדי לסייע בכך מבלי לשבור את גיבוב הגוש, הפעולות מגובבות בעץ מרקל (Merkle Tree, ראה [5][2][7]), באופן שרק השורש כלול בגיבוב הגוש. גושים ישנים יכולים לעבור צמצום על ידי גדימת ענפים מהעץ. אין צורך לשמור את הגיבובים הפנימיים.



פעולות מגובבות בעץ מרקל

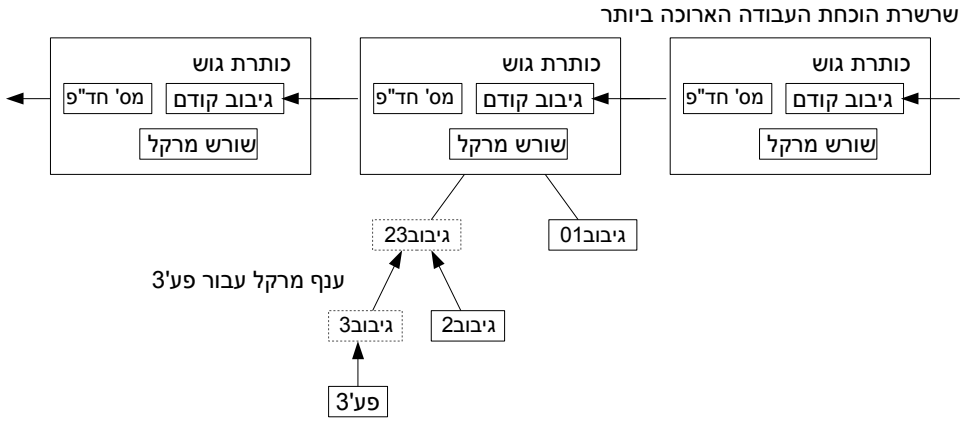


לאחר גיזום פע' 2-0 מהגוש

כותרת גוש ללא פעולות תתפוס בערך 80 בתים. אם אנו מניחים שגושים נוצרים כל 10 דקות, מדובר ב- 80 בתים * 6 * 24 * 365 = 4.2MB לשנה. בהינתן שמערכות מחשב טיפוסיות נמכרות עם 2GB זיכרון נכון לשנת 2008, וחוק מור צופה צמיחה עכשווית של 1.2GB לשנה, לא אמורה להיות בעיית אחסון גם אם חובה לשמור את כותרות הגושים בזיכרון.

8. אימות מפושט של פעולות

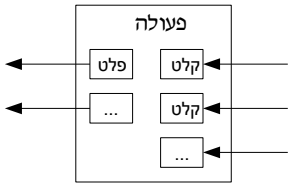
ניתן לאמת פעולות מבלי להפעיל צומת רשת מלא. משתמש צריך רק לשמור עותק של כותרות הגושים של שרשרת הוכחות העבודה הארוכה ביותר, אותו הוא יכול לקבל על ידי שאילתות לצמתים ברשת עד שהוא משוכנע שיש לו את השרשרת הארוכה ביותר, ולהשיג את ענף המרקל המקשר את הפעולה לגוש בו יש לה חותמת זמן. הוא לא יכול לבדוק את הפעולה בעצמו, אבל על ידי קישורה למקום בשרשרת, הוא יכול לראות שצומת ברשת הסכים איתה, וגושים שמתווספים אחריה מהווים אישוש נוסף שהרשת מסכימה איתה.



בתור שכזה, האימות הוא אמין כל עוד צמתים ישרים שולטים ברשת, אבל הוא פגיע יותר אם תוקף גובר על הרשת. בעוד צמתים ברשת יכולים לאמת פעולות בעצמם, השיטה המפושטת יכולה ללכת שולל אחר פעולות בדיוניות של תוקף כל עוד הוא יכול להמשיך לגבור על הרשת. אסטרטגיה אחת להגן מפני זה היא לקבל התרעות מצמתים ברשת כשהם חושפים גוש לא קביל, ולהניע את התוכנה של המשתמש להוריד את הגוש המלא והפעולות החשודות כדי לאשר את חוסר העקביות שלהן. עסקים שמקבלים תשלומים באופן תדיר כנראה ירצו עדיין להפעיל צמתים משל עצמם לצורך מידה רבה יותר של אבטחה בלתי-תלויה ואימות מהיר יותר.

9. צירוף ופיצול ערך

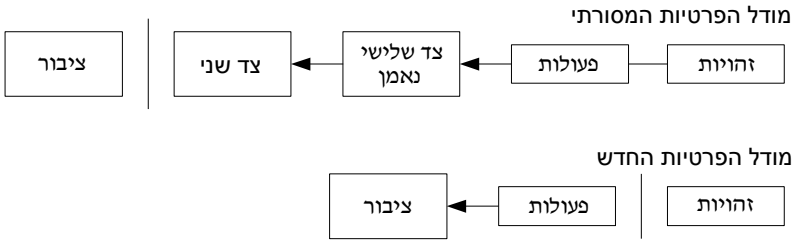
אף על פי שהיה אפשר לטפל במטבעות באופן פרטני, יהיה זה מסורבל לקיים פעולה נפרדת לכל אגורה בהעברה. כדי לאפשר לערך לעבור צירוף ופיצול, פעולות מכילות קלטים ופלטים מרובים. באופן רגיל יהיו או קלט אחד מפעולה קודמת גדולה יותר או קלטים מרובים המצרפים כמויות קטנות יותר, ולכל היותר שני פלטים: אחד בשביל התשלום, ואחד בשביל להחזיר את העודף, אם קיים, בחזרה אל השולח.



ראוי לציין שהסתעפות, בה כל פעולה תלויה במספר פעולות, ופעולות אלה תלויות ברבות נוספות, אינה בעיה כאן. לעולם אין צורך להפיק עותק שלם העומד בפני עצמו של ההיסטוריה של פעולה.

10. פרטיות

המודל הבנקאי המסורתי משיג רמה מסוימת של פרטיות בכך שהוא מגביל את הגישה למידע לצדדים המעורבים ולצד השלישי הנאמן. ההכרח בהכרזת כל הפעולות באופן פומבי מוציא מכלל חשבון את השיטה הזאת, אבל עדיין ניתן לשמור על פרטיות על ידי שבירת זרימת המידע במקום אחר: שמירת האלמוניות של מפתחות ציבוריים. הציבור יכול לראות שמישהו שולח כמות מסוימת למישהו אחר, אך ללא מידע המקשר את הפעולה לגורם כלשהו. דומה הדבר לרמת המידע המשחררות בורסות מניות, בהן הזמן והגודל של פעולות פרטניות, ה"פסנוע", מוצג באופן גלוי, אך מבלי לומר מי היו הצדדים.



כאמצעי הגנה נוסף, יש להשתמש בזוג מפתחות חדש לכל פעולה כדי למנוע מהם להיות מקושרים לבעלים משותף. מידה מסוימת של קישור היא עדיין בלתי-נמנעת עם פעולות מרובות קלטים, שבהכרח חושפות שהקלטים שלהן היו שייכים לאותו בעלים. הסיכון הוא שאם הבעלים של מפתח נחשף, קישור יכול לחשוף פעולות נוספות ששייכות לאותו בעלים.

11. חישובים

אנו בוחנים תרחיש בו תוקף מנסה ליצור שרשרת חלופית מהר יותר מהשרשרת הישרה. גם אם הדבר מתבצע, המערכת אינה נפתחת לשינויים שרירותיים, כמו יצירת ערך יש מאין או לקיחת כסף שמעולם לא היה שייך לתוקף. צמתים לא יסכימו לקבל פעולה בלתי-קבילה כתשלום, וצמתים ישרים לעולם לא יסכימו עם גוש המכיל אותן. תוקף יכול רק לנסות לשנות אחת מהפעולות שלו כדי לקחת בחזרה כסף שהוא הוציא לאחרונה.

ניתן לאפיין את המירוץ בין השרשרת הישרה ושרשרת של תוקף בתור הילוך אקראי בינומי. מאורע ההצלחה הוא שהשרשרת הישרה מתארכת בגוש אחד, מה שמגדיל את ההובלה בשיעור +1, ומאורע הכישלון הוא שהשרשרת של התוקף מתארכת בגוש חד, מה שמפחית את הפער בשיעור -1.

ההסתברות שתוקף ידביק את הפער מגרעון נתון מקביל לבעיית מפולת המהמר. נניח כי מהמר עם אשראי בלתי-מוגבל מתחיל עם גירעון ומשחק בניסיון להגיע לאיזון מספר נסיונות היכול להיות אינסופי. אנו יכולים לחשב את ההסתברות שהוא יגיע אי-פעם לאיזון, כלומר שהתוקף ידביק אי-פעם את הפער עם השרשרת הישרה, כדלקמן [8]:

p = ההסתברות שצומת ישר ימצא את הגוש הבא
q = ההסתברות שהתוקף ימצא את הגוש הבא
q_z = ההסתברות שהתוקף ידביק אי-פעם את הפער מפיגור של z גושים

$$q_z = \begin{cases} 1 & if\ p \le q \\ (q/p)^z & if\ p > q \end{cases}$$

בהינתן ההנחה שלנו כי $p>q$, ההסתברות קטנה באופן מעריכי ככל שגדל מספר הגושים איתם התוקף צריך להדביר את הפער. היות והסיכויים פועלים נגדו, אם אין לו את המזל בזינוק קדימה בתחילה, הסיכויים שלו נהיים אפסיים ככל שהפיגור שלו גדל.

עתה נבחן כמה המקבל של פעולה חדשה צריך להמתין כדי להיות בטוח במידה מספיקה שהשולח אינו יכול לשנות את הפעולה. אנו מניחים שהשולח הוא תוקף הרוצה לגרום למקבל להאמין לזמן-מה שהוא שילם לו, ואז להחליף את הפעולה כדי לשלם בחזרה לעצמו. המקבל יקבל התרעה כשזה יקרה, אך התוקף מקווה שזה יהיה מאוחר מדי.

המקבל יוצר זוג מפתחות חדש ונותן את המפתח הציבורי לשלוח זמן קצר לפני החתימה. זה מונע מהתוקף להכין שרשרת גושים מבעוד מועד על ידי עבודה רצופה עליה עד שיש לו מספיק מזל כדי להוביל במידה מספקת, ואז לבצע את הפעולה באותו הרגע. ברגע בו הפעולה נשלחת, השולח הלא-ישר מתחיל לעבוד בחשאי על שרשרת מקבילה המכילה גרסה חלופית של הפעולה שלו.

המקבל מחכה עד שהפעולה התווספה לגוש ו-z גושים שורשרו לאחריו. הוא לא יודע את מידת ההתקדמות המדויקת של התוקף, אבל בהנחה שהגושים הישרים לקחו פרק זמן ממוצע לכל גוש, ההתקדמות האפשרית של התוקף היא משתנה פואסון עם תוחלת:

$$\lambda = z\frac{q}{p}$$

כדי למצוא את ההסתברות שהתוקף יוכל עדיין להדביק את הפער, אנו כופלים את צפיפות התפלגות פואסון לכל מידת התקדמות אפשרית בהסתברות שהוא ידביק את הפער מנקודה זאת:

$$\sum_{k=0}^{\infty}\frac{\lambda^k e^{-\lambda}}{k!}\cdot\begin{cases}(q/p)^{(z-k)} & if k \le z\\ 1 & if k > z\end{cases}$$

שינוי סדר האיברים כדי להימנע מסכימת הזנב האינסופי של ההתפלגות...

$$1-\sum_{k=0}^{z}\frac{\lambda^k e^{-\lambda}}{k!}\left(1-(q/p)^{(z-k)}\right)$$

המרה לקוד בשפת C...

```
#include <math.h>
double AttackerSuccessProbability(double q, int z)
{
    double p = 1.0 - q;
    double lambda = z * (q / p);
    double sum = 1.0;
    int i, k;
    for (k = 0; k <= z; k++)
    {
        double poisson = exp(-lambda);
        for (i = 1; i <= k; i++)
            poisson *= lambda / i;
        sum -= poisson * (1 - pow(q / p, z - k));
    }
    return sum;
}
```

מהרצת מספר תוצאות, אנו יכולים לראות שההסתברות קטנה באופן מעריכי עם z.

```
q=0.1
z=0    P=1.0000000
z=1    P=0.2045873
z=2    P=0.0509779
z=3    P=0.0131722
z=4    P=0.0034552
z=5    P=0.0009137
z=6    P=0.0002428
z=7    P=0.0000647
z=8    P=0.0000173
z=9    P=0.0000046
z=10   P=0.0000012

q=0.3
z=0    P=1.0000000
z=5    P=0.1773523
z=10   P=0.0416605
z=15   P=0.0101008
z=20   P=0.0024804
z=25   P=0.0006132
z=30   P=0.0001522
z=35   P=0.0000379
z=40   P=0.0000095
z=45   P=0.0000024
z=50   P=0.0000006
```

מציאת התנאים הדרושים להסתברות קטנה מ-0.1%...

```
P < 0.001
q=0.10   z=5
q=0.15   z=8
q=0.20   z=11
q=0.25   z=15
q=0.30   z=24
q=0.35   z=41
q=0.40   z=89
q=0.45   z=340
```

12. סיכום

הצענו מערכת לעסקאות אלקטרוניות שאינה מסתמכת על אמון. התחלנו עם המסגרת הרגילה של מטבעות העשויים מחתימות דיגיטליות, המספקת שליטה חזקה בבעלות, אך אינה שלמה ללא דרך למנוע ניצול כפול. כדי לפתור זאת, בצענו רשת שיתופית המשתמשת בהוכחת עבודה כדי לנהל רישום של היסטוריה פומבית של פעולות, שבמהרה הופכת לבלתי-מעשית מבחינה חישובית להשתנות על ידי תוקף אם צמתים ישרים שולטים במרבית כוח החישוב. הרשת עמידה מעצם הפשטות חסרת המבנה שלה. צמתים עובדים בו-זמנית עם מעט תיאום. הם אינם צריכים להזדהות, היות והודעות אינן מנותבות למקום מסוים וצריכות להימסר רק על בסיס מיטב המאמצים. צמתים יכולים לעזוב את הרשת ולהצטרף אליה מחדש כרצונם, תוך קבלת שרשרת הוכחת העבודה כהוכחה למה שקרה בהיעדרם. הם מצביעים עם עוצמת חישוב, ומביעים את הסכמתם עם גושים תקינים בעבודה על המשכתם, ודוחים גושים בלתי-קבילים בסירוב לעבוד עליהם. ניתן לאכוף כל כלל או תמריץ שיידרש עם מנגנון ההסכמה הכללית הזה.

הפניות

[1] W. Dai, "b-money," http://www.weidai.com/bmoney.txt, 1998.

[2] H. Massias, X.S. Avila, and J.-J. Quisquater, "Design of a secure timestamping service with minimal trust requirements," In *20th Symposium on Information Theory in the Benelux*, May 1999.

[3] S. Haber, W.S. Stornetta, "How to time-stamp a digital document," In *Journal of Cryptology*, vol 3, no 2, pages 99-111, 1991.

[4] D. Bayer, S. Haber, W.S. Stornetta, "Improving the efficiency and reliability of digital time-stamping," In *Sequences II: Methods in Communication, Security and Computer Science*, pages 329-334, 1993.

[5] S. Haber, W.S. Stornetta, "Secure names for bit-strings," In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, pages 28-35, April 1997.

[6] A. Back, "Hashcash - a denial of service counter-measure," http://www.hashcash.org/papers/hashcash.pdf, 2002.

[7] R.C. Merkle, "Protocols for public key cryptosystems," In *Proc. 1980 Symposium on Security and Privacy*, IEEE Computer Society, pages 122-133, April 1980.

[8] W. Feller, "An introduction to probability theory and its applications," 1957.

Hebrew translation by: Meni Rosenfeld
Translation formatted by: Jim Nguyen