**בלוק'ין ומידע רפואי  
מאת רבקה אלטשולר, ת"ז 302737713  
במסגרת ממן תכנותי בקורס אבטחת מערכות ממוחשבות**

תחום המטבעות הדיגיטלים תפס מאוד בשנים האחרונות, והטכנולוגיות עליו הוא מתבסס מסקרנות הן בתעשייה והן באקדמיה. אחד הרעיונות שראה אור באמצעות המטבעות הדיגיטלים מכונה בלוקצ'ין. הבלוקצ'ין הוא מבנה נתונים בצורת שרשרת של בלוקים בעל אלמנט זהה להגדרה של readonly בשדה למחלקה בתכנות מונחה עצמים, ערך שנרשם, נרשם פעם אחת ואינו ניתן לשינוי. יישום הרעיון הוא באמצעות כלים קריפטוגרפים בני ימינו, כלומר אין לנו הוכחה שאי אפשר לשנות, אלא שהשינוי הוא קשה, כלומר אינו ניתן לביצוע בכלים שלנו היום.

באופן מעשי, פתרון הבלוקצ'ין הוא שרשרת של בלוקים שתוכן בלוק תלוי בכל הבלוקים הקודמים, כלומר אם משנים בלוק, צריך לשנות את כל הבלוקים הבאים בהתאם. וכן יצירת בלוק תקין דורשת זמן (של עבודה חישובית). כאשר שני האלמנטים מופיעים – אם אי מי ינסה את כוחו וישנה בלוק, הוא יצטרך לשנות ולתקן את כל הבלוקים הבאים, שינוי ותיקון של כל בלוק ידרוש זמן עבודה, כאשר בינתיים נוצרים עוד ועוד בלוקים, באופן מעשי הוא לא יעמוד בקצב. האלמנט המהיר והחשוב הוא שבדיקה שהשרשרת תקינה **אינה** דורשת זמן רב של עבודה חישובית.

אלמנט חשוב נוסף שמונע מאי מי לשנות את שרשרת הבלוקים - היא יצירת צמתים נוספים שמתחזקים אף הם את הבלוקצ'ין, הסכמה בין צמתים לגבי תוכן השרשרת, מייתרת את הצורך בגורם מרכזי. זה נושא מעניין בפני עצמו שמאוד משמעותי בעידן המטבעות הדיגיטלים, אך נשים אותו בצד לעת עתה.

בפרויקט שבניתי חשפתי לblockChain מתודה שנקראת Verify, מתודה זו עוברת על שרשרת הבלוקים ומוודאת שהבלוקים תקינים. פונקציות ערבול קריטוגרפיות מאפשרות לזהות בלוקים באופן ייחודי על ידי תוכן קצר ברמת סבירות גבוהה, דבר זה מאפשר למצוא לבלוק כעין טביעת אצבע, כלומר אם אי מי ישנה תוכן של בלוק, תוצאת אלגוריתם העירבול אף היא תשתנה. בפוקנציה verify אנו בודקים שכל בלוק מחזיק את תוצאת הערבול של הבלוק שקדם לו. בפרויקט, המחלקה Block מכילה שדה PreviousHash, אם משנים את השדה PreviousHash – תוצאת הערבול של הבלוק הנוכחי משתנה ואז הבלוק הבא יצטרך לשנות את PreviousHash אף הוא וכן הלאה. הדבר השני שהפונ' בודקת היא שלבלוקים ישנה הוכחה שבוצעה עליהם עבודה מסדר חישובי מתאים באופן סטטיסטי, דבר זה נקרא PoW (-Proof Of Work) הדרך שבה מימשתי זאת היא על הדרך של ביטקוין. הוספתי לבלוק שדה nounce שנהיה חלק מתוכן הבלוק, ולכן משפיע על תוצאת הערבול, כאשר סוגרים בלוק מחפשים ערך מתאים לnounce כך שתוצאת הערבול תהייה עם אילוץ פורמט מסוים, בצורה זו יש לנסות ערכים שונים עד מציאת ערך מתאים, בדיקת הנסיונות לוקחת זמן חישובי שלא נמצא בכלים של ימינו דרך שאינה קשה לבצע זאת (בדומה למה שלמדנו עם הsalt בסיסמאות)

יצירת מופע של BlockChain דורשת פרמטר proofOfWorkDifficulty שמשמעותו רמת קושי חישובית של הPoW ובצורה טכנית, ערבול שמוכיח שנעשה עבורו PoW הוא כזה שיש לו מספר אפסים מובילים כמו המספר שנמצא ב- proofOfWorkDifficulty ככל רמת הקושי שתקבע תהיה גבוהה, כמות הנסיונות שתדרש תהיה גבוהה באופן סטטיסטי.

פרמטר נוסף הנדרש עבור הblockChain הוא durationBetweenBlocksCreation ומשמעותו כל כמה זמן ייסגר בלוק. חשוב להבין שבלוקים חדשים יווצרו בקצב הזה בכל מקרה גם אם אין בהם מידע כלל, זאת כחלק מהארכיטקטורה שתיארתי, שלא תהיה יכולת לשנות ולעמוד בקצב.

מימשתי את הblockchain באופן בלתי תלוי בשמירה פיסית, הblockChain צריך לקבל IBlocksRepository שחושף את היכולת לשמור בלוק, ולקרוא את רשימת הבלוקים, מימשתי אופצית מימוש פשוטה ששומרת את הבלוקים בפורמט json בקבצים, כאשר שם הקובץ הוא המספר הסידורי שלו, הכל תחת תיקית blocks.

הפרויקט שמימשתי הוא מרכיב חשוב ובסיסי להבין את משמעות הbuzzword בלוקצ'ין, המימוש גם הוא מעניין ומשתמש באלמנטים של Rx וCleanCode-. יש עוד הרבה מה להוסיף ולשפר, במיוחד האלמנט של ביזור וקונצנזוס. אלמנט של אבטחת מידע חשוב שאינו מרכיב בבלוקצ'ין הוא מערכות הצפנה והרשאה. אך כאמור הפרויקט הנוכחי הוא מרכיב התחלתי בלבד במסלול לימוד ומחקר בנושא.

בשביל להרים את הבלוקצ'ין, יש ליצר מופע:

var blockChain = new BlockChain(proofOfWorkDifficulty: 4,

durationBetweenBlocksCreation: TimeSpan.FromMinutes(1),

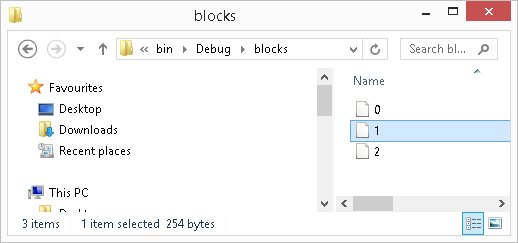
blocksRepository: blocksRepository);

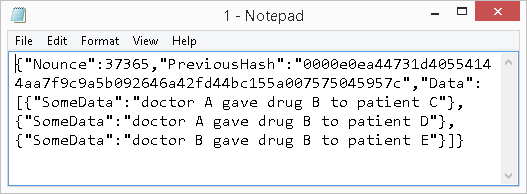
יצרתי מימוש מבוסס קבצים לתחזוקה פיסית של המידע, אתחול ה- blocksRepositoryהוא כדלהלן:

var blocksRepository = new BlocksRepository<Block>();

לצורך הפרויקט השתמשתי ברשומות טקסטואליות, במערכות אמיתיות יש לבצע סרליזיה לאובייקטים המתאימים. הוספת מידע מתבצעת כדלהלן:

blockChain.Add(new Data("doctor A gave drug B to patient C"));

כחלק מהפרויקט אני מראה את הצד של מי שינסה לשנות ללא הצלחה, הrepository שלנו הוא מול מערכת קבצים פשוטה, כל בלוק נשמר בקובץ אחר, ניתן לפתוח ולשנות באמצעות הnotepad:



בעקבות שינוי הבלוק, קרו שני דברים: ראשון, הבלוק הבא מכיל תוצאת ערבול לא נכונה, כי הרי הבלוק השתנה ותוצאת הערבול יחד איתו. שני, תוצאת הערבול החדשה כנראה לא מכילה את הPoW, כלומר במימוש שלנו זה לא יתחיל בכמות אפסים המתאימה. בשביל לבדוק את זה הוצאתי למחלקה חיצונית את היכולות של הערבול:

var hashGenerator = new HashGenerator(proofOfWorkDifficulty: 4);

הrepository שלנו כזכור חושף אפשרות של קריאה (לא של עדכון) ניתן להשתמש בו ו"לתקן" את הבלוקים בעקבות השינוי.

var firstBlock = blocksRepository.First();

var existsNounce = firstBlock.Nounce;

hashGenerator.HashWithProofOfWork(firstBlock);

על פי תוצאות החישוב ניתן לעדכן את הבלוקים.

כעת בכל מקרה כאשר נפעיל את המתודה Verify נקבל הודעת שגיאה. שרשרת הבלוקים אינה יכולה להיות תקינה, גם אם נשנה באופן יעיל יותר. קצב יצירת הבלוקים שמתאים בסדר גודל לזמן שצריך להשקיע בתיקון הבלוקים - אינו מאפשר שינוי.

**הקשר לעולם הרפואי**

בעולם הפיננסי הבלוקצ'ין מהווה חומת אבטחה מצוינת מפני תקיפות רשת והונאות כספים, כאשר בעלי עניין שונים מסכימים לגבי שרשראות בלוקים הגדלות באופן קבוע מבלי יכולת לשנות. בעקבות החידוש הזה חברות סטארטאפ רבות אימצו את הbuzzword כדי להרשים את המשקיעים/לקוחות. ויודעי דבר הכריזו על הבלוקצ'ין כפתרון קסם לתעשיות נוספות. אך יש להבין היטב את היתרונות וההשלכות. ולהבין היכן ולמה הדבר מביא תועלת. יש להבין שבבסיס מערכת כזו יש ליצר בלוקים ללא הפסק גם אם אין נתונים לשמור, הכוח של הביזור הוא גדול, אך הוא מבזבז משאבים ומתאים לרישומים של מידע בעל אנטרופיה נמוכה.

על אותה הדרך נטען שהבלוקצ'ין יכול להוות פתרון עבור רשומות רפואיות: מערכות שונות מתחזקות נתונים רפואיים שונים, ישנו קושי בכך שהן אינן מתקשרות ביניהן. נטען, שניתן לבנות פלטפורמה מבוססת בלוקצ'ין שבה פציינט יתחזק את תיקו ויחלוק אותו עם נותני השירות. אך יש להבין שמידע רפואי מכיל נפח רב יותר ממידע פיננסי, צילומים למשל הם בעלי נפח רב. אך הפרט העיקרי הוא שיש להבין מה נותן הבלוקצ'ין כאן מעבר לשרת מרכזי יעודי. האם הסיבה שעד היום לא נבנתה מערכת כזו הייתה כי חסרה הגנה מפני **שינוי הנתונים**?

ההיבט של Readonly שייך כאשר מתעדים אירועים שקרו לטוב ולרע, הדרישה שהפרטים יהיו אכן ללא שינוי הכרחי לתיעוד ומעקב של policies בחברות. מי שעומד מאחורי דרישות מהסוג הזה צריך להיות הרגולטור שצריך להכיר בחורי האבטחה הפותחים אפשרות של שינוי מידע לאחור, ולדרוש רישום בפלטפורמה חסינת שינויים. וכך הודות לבלוקצ'ין, ניתן להטמיע במערכות רפואיות יכולת חדשה, נקרא לה Proof, שרישום מסוים קרה. ניתן להמשיך להשתמש במערכות הקיימות ובאמצעי האחסון, אך ניתן לבצע ערבול על הנתונים, ורק את תוצאת הערבול (בעל נפח DATA קטן) להכניס לבלוקצ'ין, כאשר יידרש לוודא שלא בוצע שימוש בנתונים, יבוצע שוב הערבול ויושווה לערך שהוטמע בבלוקצ'ין. ככל שהבלוקצ'ין שימושי יותר, קצב יצירת בלוקים מהיר מוצדק יותר ואז מאובטח יותר. לכן רעיון של בלוקצ'ין רב תכליתי יכול מאוד להתאים. חברת Tierion יצרה פתרון שנקרא [Chainpoint](https://chainpoint.org) ומציעה באמצעותו שירותים מתאימים.

יישומים של BlockChain עבור מערכות רפואיות הוא שדה מחקר חדש, [BHTY](https://blockchainhealthcaretoday.com) (Blockchain in Healthcare Today), הינה פלטפורמה למחקר עם ביקורת עמיתים בתחום.

דוגמת ריצה

