**המחלקה להנדסת תוכנה**

**פרויקט גמר – תשע"ח**

שיתוף מפתחות מבוסס הסתברות עבור IoT

Probability Based Keys Sharing for IoT

**מאת**

**שרה ספרין 312548779**

**רעות נגר**

**מנחה אקדמי: דר' גיא לשם אישור: תאריך:**

**רכז הפרויקטים: דר' אסף שפנייר אישור: תאריך:**

# תקציר

במהלך השנים האחרונות, התפתחות בתחום ה"אינטרנט של דברים" (IoT) צוברת תאוצה. מכשירים אלו הופכים ליעד להתקפות. מכיוון שהמשאבים שלהם דלים לא ניתן לממש בהם פתרונות אבטחה קיימים ולכן חסרה בהם אבטחה נאותה.

פרויקט זה מתעסק במציאת פתרון אבטחה עבור רשתות קטנות של IoT. המימוש מתאפשר גם ברשתות גדולות ע"י היררכיה.

אנו מציגות פרוטוקול אבטחה חדש המורכב מחמישה שלבים:

* מציאת מאסטר
* מציאת גודל בריכת המפתחות ויצירתה
* חלוקת מפתחות
* מציאת מפתח משותף
* רשת בטוחה

עבור השלב השני בפרוטוקול הסתמכנו על מאמר מתמטי שדן בגודל מאגר המפתחות שיש ליצור על מנת שתהיה חפיפה של מפתח אחד לפחות בין כל שני צמתים ברשת בהסתברות גבוה. זאת על מנת שהרשת תהיה דינאמית, ושינויים בטופולוגית הרשת לא יפקיעו את אבטחתה.

מימשנו את הפתרון על שלושה מכשירי IoT מסוג 3 Raspberry Pi, כאשר אחד המאסטר והשניים האחרים הם שני צמתים ברשת.

# תוכן העניינים

[תוכן העניינים 3](#_Toc517102256)

[תקציר 4](#_Toc517102257)

[מילון מונחים 4](#_Toc517102258)

[תיאור מסגרת הפרויקט 5](#_Toc517102259)

[מבוא 5](#_Toc517102260)

[אבטחת מידע 5](#_Toc517102261)

[RSA 6](#_Toc517102262)

[AES 6](#_Toc517102263)

[מכשירי IoT 6](#_Toc517102264)

[תיאור הבעיה 7](#_Toc517102265)

[תיאור הפתרון 8](#_Toc517102266)

[פרוטוקול "לחיצת יד" חדש עבור רשת התקני IoT 8](#_Toc517102267)

[תיאור הכלים המשמשים לפתרון 10](#_Toc517102268)

[ארכיטקטורת המימוש 10](#_Toc517102269)

[1. אתחול ומציאת מאסטר 10](#_Toc517102270)

[2. חישוב גודל מאגר המפתחות ויצירתו 11](#_Toc517102271)

[3. חלוקת מפתחות 12](#_Toc517102272)

[4. מציאת מפתח משותף 13](#_Toc517102273)

[5. רשת בטוחה 14](#_Toc517102274)

[נוסחאות מתמטיות 15](#_Toc517102275)

[הודעות 15](#_Toc517102276)

[חוטים (threads) 15](#_Toc517102277)

[מכשירים 15](#_Toc517102278)

[תיאור המערכת שמומשה 15](#_Toc517102279)

[מימוש הפרוטוקול: 15](#_Toc517102280)

[1. אתחול ומציאת מאסטר 16](#_Toc517102281)

[2. חישוב גודל מאגר המפתחות ויצירתו 17](#_Toc517102282)

[3. שליחת מפתחות באופן מוצפן 17](#_Toc517102283)

[נוסחאות מתמטיות 20](#_Toc517102284)

[הודעות 20](#_Toc517102285)

[חוטים (threads) 20](#_Toc517102286)

[מכשירים 20](#_Toc517102287)

[תוכנית בדיקות 20](#_Toc517102288)

[בדיקות פונקציונליות 20](#_Toc517102289)

[בדיקות מערכת 20](#_Toc517102290)

[בדיקות תאימות 21](#_Toc517102291)

[בדיקות תחזוקה (Maintainability) 21](#_Toc517102292)

[מסקנות 21](#_Toc517102293)

[סקירת עבודות דומות בספרות והשוואה 21](#_Toc517102294)

[נספחים 23](#_Toc517102295)

[א. רשימת ספרות \ ביבליוגרפיה 23](#_Toc517102296)

[ב. תרשימים וטבלאות 23](#_Toc517102297)

# מילון מונחים

* מכשיר IoT: מכשיר עם יכולות חישוב קטנות, בעל יכולת שידור אלחוטי
* צומת: מכשיר IoT שנמצא ברשת
* מאסטר/מנהיג: צומת שנבחר לתפקיד יצירת המפתחות והפצתם ברשת
* צופןRSA : מערכת הצפנה אסימטרית שמשתמשת במפתח ציבורי שגלוי לכולם ובמפתח פרטי, סודי.
* צופן AES: מערכת הצפנה סימטרית.

# תיאור מסגרת הפרויקט

הפרויקט שלנו הוא פרוטוקול אבטחה לרשתות קטנות של IoT . הוא ניתן לשימוש גם במערכות גדולות ע"י היררכיית מאסטרים. הפרוטוקול מורכב מ-5 שלבים שבסופם כל שני מכשירים (בהסתברות של 90%-80%) ברשת יכולים לדבר ביניהם ישירות באופן בטוח. המערכת עמידה בפני שינויים, כמו התווספות של מכשיר חדש לרשת ויציאה של מכשיר מהרשת.

בפרויקט השתמשנו ב-IoT מסוג raspberry pi ואת הקוד כתבנו בשפת פייתון. השתמשנו בספריות מתמטיות כמו numpy ו- scipy על מנת לפצח את הנוסחא המתמטית (שהיא בעיית NP קשה) של חישוב גודל הבריכה שעל המאסטר לייצר.

כמו כן השתמשנו בספריית pycrypto על מנת להצפין בRSA וב- AES.

# מבוא

## אבטחת מידע

אבטחת מידע (באנגלית: Information Security) היא ענף העוסק בהגנה של מידע ומערכות מידע מפני כל גישה למידע שאינה ע"י גורמים מאושרים, לכך עליה לספק את שלושת הבאים: סודיות, שלמות וזמינות של המערכות והמידע בהן. אבטחת מידע היא תחום מתקדם מאוד בימינו. האפשרויות הרבות שעומדות לתוקפים של מערכות ממוחשבות להזיק בתחומים כמו פרטיות, פיננסים וביטחון מניעה את האנושות להגן ככל האפשר על מערכות אלו. מושקעים משאבים רבים בעיצוב המערכות, קידוד ותחזוקה שוטפת כדי לאתר פרצות אבטחה מוקדם ככל האפשר. עם התקדמות היכולות של התוקפים נדרשות לעתים פריצות דרך מצד קהילת המפתחים ולכן אבטחת מידע זהו תחום שמושקעים בו כסף ומשאבים רבים במיוחד על מנת לחקור ולגלות דרכי הגנה חדשות ויעילות יותר להתמודדות מול התוקפים[1].

כוח יישומי חשוב באבטחת המידע היא קריפטוגרפיה. זהו ענף במתמטיקה ומדעי המחשב העוסק במחקר ופיתוח שיטות אבטחת מידע ותקשורת נתונים, ומייצר שיטות למימוש בפועל של מושגי אבטחת המידע. הגנה זו מבוצעת על ידי הצפנת המידע בעזרת פונקציית הצפנה כלשהי, ושימוש במפתח (Key) שהוא רצף תווים סודי שאינו ידוע לתוקף.  הצופן   (Cipher) הוא כתב הסתר שמתקבל על ידי הפעלת פונקציית ההצפנה על טקסט הקלט. פונקציה הצפנה אידיאלית היא כזו שבהינתן הצופן- הפלט, לא ניתן להסיק ממנו מידע על הקלט.

## צופן RSA

RSA היא מערכת [הצפנת](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) [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) [דטרמיניסטית](https://he.wikipedia.org/wiki/%D7%90%D7%9C%D7%92%D7%95%D7%A8%D7%99%D7%AA%D7%9D_%D7%93%D7%98%D7%A8%D7%9E%D7%99%D7%A0%D7%99%D7%A1%D7%98%D7%99) מעשית הראשונה שהומצאה והיא עדיין בשימוש נרחב במערכות [אבטחת מידע](https://he.wikipedia.org/wiki/%D7%90%D7%91%D7%98%D7%97%D7%AA_%D7%9E%D7%99%D7%93%D7%A2) מודרניות , [תקשורת מחשבים](https://he.wikipedia.org/wiki/%D7%AA%D7%A7%D7%A9%D7%95%D7%A8%D7%AA_%D7%9E%D7%97%D7%A9%D7%91%D7%99%D7%9D) ו[מסחר אלקטרוני](https://he.wikipedia.org/wiki/%D7%9E%D7%A1%D7%97%D7%A8_%D7%90%D7%9C%D7%A7%D7%98%D7%A8%D7%95%D7%A0%D7%99). ב־RSA , כבכל מערכת מפתח ציבורי, מפתח ההצפנה אינו סודי והוא שונה ממפתח הפענוח שנשמר בסוד, על כן היא נקראת אסימטרי. האסימטריה ב־RSA נובעת מהקושי המעשי שב[פירוק לגורמים](https://he.wikipedia.org/wiki/%D7%A4%D7%99%D7%A8%D7%95%D7%A7_%D7%9C%D7%92%D7%95%D7%A8%D7%9E%D7%99%D7%9D_%D7%A9%D7%9C_%D7%9E%D7%A1%D7%A4%D7%A8_%D7%A9%D7%9C%D7%9D) של מספר פריק שהוא כפולה של שני [ראשוניים](https://he.wikipedia.org/wiki/%D7%9E%D7%A1%D7%A4%D7%A8_%D7%A8%D7%90%D7%A9%D7%95%D7%A0%D7%99) גדולים, שהיא [בעיה פתוחה](https://he.wikipedia.org/wiki/%D7%91%D7%A2%D7%99%D7%94_%D7%A4%D7%AA%D7%95%D7%97%D7%94_%D7%91%D7%9E%D7%AA%D7%9E%D7%98%D7%99%D7%A7%D7%94) ב[תורת המספרים](https://he.wikipedia.org/wiki/%D7%AA%D7%95%D7%A8%D7%AA_%D7%94%D7%9E%D7%A1%D7%A4%D7%A8%D7%99%D7%9D).

השימוש: השולח משתמש במפתח ההצפנה הציבורי של הנמען כדי להצפין עבורו מסר כך שרק הנמען מסוגל לפענחו באמצעות המפתח הפרטי המתאים שברשותו.

אלגוריתם RSA נחשב איטי יחסית ועל כן אינו מתאים להצפנה ישירה של מידע בכמות גדולה. לכן, RSA משמש לשיתוף והעברה של מפתח [סימטרי](https://he.wikipedia.org/wiki/%D7%94%D7%A6%D7%A4%D7%A0%D7%94_%D7%A1%D7%99%D7%9E%D7%98%D7%A8%D7%99%D7%AA) סודי אשר בתורו משמש להצפנה מהירה של המידע עם אלגוריתם [סימטרי](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) כמו [AE](https://he.wikipedia.org/wiki/AES)S.

גודל מפתח ההצפנה ב-RSA של 2048 ביט מקביל ברמת האבטחה לגודל מפתח של 128 ביט ב-AES. [2]

## AES

AES (באנגלית: Advanced Encryption Standard) הוא [צופן בלוקים](https://he.wikipedia.org/wiki/%D7%A6%D7%95%D7%A4%D7%9F_%D7%91%D7%9C%D7%95%D7%A7%D7%99%D7%9D) [סימטרי](https://he.wikipedia.org/wiki/%D7%94%D7%A6%D7%A4%D7%A0%D7%94_%D7%A1%D7%99%D7%9E%D7%98%D7%A8%D7%99%D7%AA) שאומץ על ידי [המכון הלאומי לתקנים וטכנולוגיה](https://he.wikipedia.org/wiki/%D7%94%D7%9E%D7%9B%D7%95%D7%9F_%D7%94%D7%9C%D7%90%D7%95%D7%9E%D7%99_%D7%9C%D7%AA%D7%A7%D7%A0%D7%99%D7%9D_%D7%95%D7%98%D7%9B%D7%A0%D7%95%D7%9C%D7%95%D7%92%D7%99%D7%94) (NIST) של [ארצות הברית](https://he.wikipedia.org/wiki/%D7%90%D7%A8%D7%A6%D7%95%D7%AA_%D7%94%D7%91%D7%A8%D7%99%D7%AA) [כתקן](https://he.wikipedia.org/wiki/%D7%AA%D7%A7%D7%9F) [הצפנה](https://he.wikipedia.org/wiki/%D7%94%D7%A6%D7%A4%D7%A0%D7%94) רשמי שהתקבל בעולם כולו, להצפנת נתונים מאסיבית. אלגוריתם AES נמצא בשימוש מעשי נרחב בכל העולם הן בתוכנה והן בחומרה וידוע כאלגוריתם בטוח. [3]

## מכשירי IoT

בעולם הטכנולוגי כיום קיימת מגמה מואצת להפוך כל מכשיר המכיל שבב אלקטרוני לבעל יכולת חיבור לרשת. כבר כיום ניתן לראות מכשירים מכל הסוגים שקיימת בהם אפשרות חיבור לאינטרנט. לדוגמה: מצלמות, מדפסות, שלטי מזגן ועוד. החזון לחבר כל מכשיר חשמלי לרשת נובע מן הרצון שמכשירים כאלו ישדרו למכשירים ומערכות סביבם את המידע שבידם, ויקבלו מידע מן הרשת לגבי אירועים ותרחישים שונים. הודעות אלו ישדרגו את יכולותיהם ופעילותם והם יוכלו להתנהג בצורה יותר "חכמה", להסיק מסקנות ולפעול אוטומטית על פיהם ללא התערבות אנושית. לדוגמה, כאשר השבב של מערכת החלונות המותקנת ב"בית חכם" קולט שידור ממערכת המיזוג על הפעלת המזגן, הוא יודע לסגור את החלונות המתאימים באופן אוטומטי כדי לייעל את פעולת המיזוג.

מגמה זאת נקראת: "אינטרנט של דברים" (IoT- Internet Of Things). זוהי רשת של חפצים פיזיים, או "דברים" המשובצים באלקטרוניקה, תוכנה וחיישנים המאפשרים תקשורת מתקדמת בין החפצים ויכולות איסוף והחלפת מידע. רשת זו צפויה להוביל לאוטומציה בתחומים רבים. כיום ישנה התפתחות נרחבת בתחום ה IoT, למשל "הבית החכם", שבו כל המכשירים מחוברים לרשת וניתן להפעיל אותם בשלט רחוק ולתאם בין פעולותיהם. תחום הIoT צפוי לגלגל מחזור של כ20 ביליון דולר בשנת 2020, על פי הודעת חברת 'סיסקו' העולמית, ומושקעים בו משאבים רבים לפיתוח מצד החברות בתחום החומרה והתוכנה.

לקראת השינוי הזה יידרש שיפור גם ברמות האבטחה המקובלות כיום בקרב מכשירים כאלו, שעצם חיבורם לרשת חושף אותם להתקפות חיצוניות והם עלולים להוות טרף קל לתוקף. בפרויקט זה נתמקד בחקירת פתרונות אבטחה, בפרט בשימוש בהצפנת RSA, עבור תחום ה"אינטרנט של הדברים" ויצירת פתרונות אבטחה ייעודיים עבורו המתחשבים בחזקות והחולשות של המוצרים הקיימים בשוק[4].

# תיאור הבעיה

בהבנה של הצורך הבסיסי באבטחה ראויה למידע המשתמש והכרה במגבלות המשאבים של מכשירים אלקטרוניים קטנים, ניתן להבחין במספר בעיות ביישום אבטחה במכשירי IoT:

* מכיוון שמכשירי הIoT הינם מכשירים זולים וקטנים לרוב, ומאופיינים ביכולות עיבוד חלשות, ומשאבים נוספים דלים כמו: זיכרון, שידור וחישה, לכן להטמיע בהם יכולות אבטחה מתקדמות כמו שקיימות במערכות מחשבים גדולות זוהי משימה קשה וכמעט לא נתמכת מבחינת החומרה של ה IoT.
* מכיוון שהשימוש ב IoT הוא כמכשירים שיש להם בדרך כלל תפקיד עיקרי ייעודי והחיבור לרשת רק מוסיף להם יתרון, הם לא נתפסו עד היום בציבור כבעלי ענין עבור תוקפים. אך עם השימוש הגובר בהם מיום ליום נוצלו פרצות האבטחה שבמכשירים מסוג זה לתקיפת מערכות רגישות, כמו מצלמות אבטחה, מדפסות ועוד. לכן נדרש פתרון שיספק אבטחה הולמת כנגד ניסיונות פריצה אפשריים.
* פתרונות האבטחה שכן מיושמים כיום בתחום ה IoT הינם ייעודיים עבור מכשיר מסוג מסוים, ומוטמעים על ידי היצרן. לא נלקחת בחשבון התאמה לשוק המוצרים הכולל של IoT, ולכן פתרונות אלו אינם מתאימים מבחינת תצורה, מגבלות משאבים וייעוד, עבור כל התחום.
* רשתות מכשירים אלו נפרשות על פני אזורים גאוגרפיים נרחבים והם מפוזרים בתפוצה גבוהה. נתון זה מקשה על יכולת ניהול וניטור מצב האבטחה ברשת על ידי גורמים חיצוניים ונדרש יישום פנימי ועצמאי של עקרונות אבטחה.

על מנת לפתור את הבעיות הנ"ל נדרש פיתוח חדשני שיענה על דרישות האבטחה הגבוהות בשוק, יחד עם המגבלות המאפיינות את מכשירי הIoT.

# תיאור הפתרון

קיימות שתי מגמות קוטביות בניסיון להגיע להסכמה על מפתח משותף בין שני צמתים על מנת שישתמשו בו להצפנת ההודעות ביניהם בצופן סימטרי ומהיר:

האחת היא לספק **מפתח יחיד לכל ההתקנים** ברשת ואתו יצפינו את כל ההודעות ביניהם. גישה זו קלה למימוש ואינה דורשת משאבים רבים. אולם, קיימת בה פרצת אבטחה חמורה שברגע שנפרץ מכשיר אחד ברשת- המערכת כולה אינה בטוחה.

מגמה שניה היא ליצור "לחיצת יד" בין **כל זוג התקנים** ברשת וכך יסכמו על מפתח משותף שישמש אותם בתקשורת ביניהם. זוהי גישה שמספקת את האבטחה הטובה ביותר מכיוון שחשיפת התקן ברשת משאירה את כל שאר התקשורת מוגנת. זוהי אבטחה שלמה, אך מחירה מתבטא באיבוד של משאבים רבים ויקרים של התקנים חלשים אלו ופוגם ביעילות המערכת.

הפתרון שאנו מציגות, מנסה לשלב בין שתי הגישות לעיל: לקבל מקסימום אבטחה במינימום תקורה של שימוש במשאבי ההתקנים. אנו מעוניינות לשתף מפתחות בין ההתקנים ברשת באופן הסתברותי, כך שבהסתברות גבוהה מספיק תהייה חפיפה בין קבוצות מפתחות שמחזיקים כל שני התקנים.

על מנת שהתהליך יהיה דינאמי, עצמאי ולא יזדקק לניהול חיצוני הוא ייעשה רק ע"י המכשירים בתוך הרשת.

## פרוטוקול אבטחה חדש עבור רשת התקני IoT

על ידי מערכת חדשה לניהול מפתחות המבוססת על שיתוף מפתחות הסתברותי בין ההתקנים.

להלן הפרוטוקול:

1. יצירת קבוצה של התקני IoT  עם תקשורת ביניהם והגדרת מנהיג לקבוצה.
2. המנהיג מייצר מאגר מפתחות. המנהיג צריך לקבוע מה יהיה ,P גודל המאגר, כאשר P מתבסס על המשתנים להלן. המנהיג קובע גם את רמת ההבטחה שתהיה חפיפה של מפתח אחד לפחות בין כל שני התקני IoT במערכת.

האלגוריתם למציאת גודל המאגר[5]:

**M** - גודל הזיכרון הפיזי (לדוג' 32M).

**n** - מספר התקני ה-IoT ברשת.

**n’** - גודל ה'שכונה' של צומת ברשת האלחוטית. 'שכונה' זוהי קבוצת התקנים שיכולה להיות ביניהם תקשורת ישירה, כלומר שנמצאים בטווח השידור הפיזי אחד של השני.

**Pc** - ההסתברות שיש **לפחות** שני התקני IoT בשכונה שלהם לפחות מפתח אחד משותף באופן ודאי.

**k** - גודל תת הקבוצה של מפתחות שיחזיק כל צומת ברשת. k מוגבל ע"י M, שכן מכשיר לא יכול להכיל קבוצת מפתחות הגדולה מהזיכרון הפנוי שלו.

**p’** - ההסתברות שלכל שני צמתים בשכונה קיים מפתח משותף.

* https://lh5.googleusercontent.com/oPNwTwULAVGT9zdLpXk5pTttHDa_sIZs_U-QyxZnAUCoftw336Uk6GT9pk_lrsfd-j5esOgIO6CXhcXyzGpM8aatGVezNo2ZrpgrN0tqWSpEh2pjTDc72C3IT_vH42S1395BISTbחשב את c, כאשר c קבוע:
* חשב את p: ההסתברות ששני צמתים מחוברים ישירות:
* https://lh5.googleusercontent.com/spCiRq3CDPin_W1-VfwLs7urZYRMAPEJaIDiduLsjHN7VTEAZ7CkXGX0MExFbsL74QZ74K5rt47J9hrtaYNc-pxNTmoy3QEAOk7lZw7cx12KbCZD3V0_gVaFDVwx6_qpx8CjWgo6חשב את d: הערך המצופה של הצומת:
* חשב את p' :
* בהתחשב בערך של p', השתמש במשוואה הבאה כדי לחלץ את הערך של P:

בהינתן P גודל המאגר כמו שחושב באלגוריתם המוצג בשלב 2, המנהיג צריך לייצר מאגר של מפתחות בגודל P ,ולכל מפתח במאגר להוסיף מספר סידורי (index).

1. המנהיג שולח לכל חבר ברשת תת קבוצה שונה של מפתחות בצורה מוצפנת.
2. כל זוג צמתים ברשת מוצאים את המפתחות המשותפים להם.
3. כאשר צומת רוצה לתקשר עם צומת אחר התקשורת תהיה מוצפנת עם המפתח המשותף.

הפתרון שלנו מאפשר להימנע מניהול האבטחה באמצעים חיצוניים לרשת, ממומש כולו ע"י ההתקנים הנמצאים ברשת, ומספק אבטחה טובה מספיק עבור התקשורת בין המכשירים. הפתרון שלנו גמיש לשינויים בטופולוגית הרשת בכך שהתקן יכול להצטרף לרשת בכל זמן נתון והמנהיג ישלח לו תת קבוצה חדשה של מפתחות, וכן התקן שעוזב את הרשת לא חושף את כל מפתחות ההצפנה וניתן להשתמש בכל שאר המפתחות שלא נחשפו.

## תיאור הכלים המשמשים לפתרון

בפרויקט עבדנו עם מכשירי 3 Raspberry Pi, מכשיר בעל קלט ופלט אלחוטיים ויכולת עיבוד קטנה על מנת לבדוק וליישם את האלגוריתם.

הכתיבה נעשתה בשפת Python , הספריות העיקריות שהשתמשנו בהם:

* **socket** ליצירת תקשורת
* **pycryptodome** למימוש הצפנות בRSA ובAES
* **threading** בשביל מקביליות
* **math, numpy, scipy** לפתירת הנוסחאות המתמטיות שנדרשו על מנת למצוא את גודל הבריכה שעל ה"מאסטר" לייצר.

## ארכיטקטורת המימוש

התמקדנו בתכנון קוד פשוט ומודולרי שיבצע את פעולות המאסטר והקליינט באותו קובץ הרצה, תוך שמירה על קיום מופע אחד בלבד של "מאסטר" ברשת. בנוסף הקפדנו שהתוכנה תהיה גנרית ותוכל לרוץ על מגוון של מערכות הפעלה והתקנים.

### ארכיטקטורה של שלבי הפרוטוקול:

1. אתחול ומציאת מאסטר

מיד כשמכשיר נכנס לרשת הוא שולח הודעת broadcastברשת: IS\_THERE\_MASTER ומחכה פרק זמן לתשובה:

* אם קיבל הודעה I\_AM\_MASTER - יעבור למצב MASTER\_FOUND, ויפסיק בחיפוש המאסטר. מעתה יאזין רק להודעות מהמאסטר, על מנת לקבל את המפתחות ושאר התהליך עד לסיום הקמת רשת מאובטחת.
* במקרה ולא קיבל תשובה (פעמיים, שמא קרתה תקלה ברשת) הוא מכריז על עצמו כמאסטר ועובר למצב MASTER\_FOUND. הוא שולח הודעת broadcast I\_AM\_MASTER. אם יקבל הודעת IS\_THERE\_MASTER הוא ישלח הודעה פרטית לאותו מכשיר I\_AM\_MASTER.



1. חישוב גודל מאגר המפתחות ויצירתו

ה"מאסטר" מחשב את כמות המפתחות שהוא נדרש לייצר וכן את מספר המפתחות שעליו לחלק לכל צומת עפ"י נתונים פיזיים כמו גודל זיכרון וכן עפ"י ידע מוערך מוקדם כמו מספר הצמתים ברשת, ב'שכונה' וכ"ו. עפ"י חישובים מתמטיים הוא מגיע לגודל הבריכה אותו עליו לייצר, ומייצר אותם.

כל מפתח שהמאסטר מייצר נכנס למערך ששמור אצלו. אינדקס המפתח מוגדר כמיקומו במערך המפתחות. לאחר מכן המאסטר עובר למצב MASTER\_DONE.



1. חלוקת מפתחות

כל "לקוח" מייצר מפתח RSA פרטי ופומבי, ושולח את המפתח הפומבי ל"מאסטר" בהודעת CLIENT\_PUBLIC\_KEY על מנת שיוכל לשלוח לו את המפתחות מוצפנים. המאסטר מגריל k מפתחות מהמאגר ושולח ללקוח כל מפתח בהודעת CLIENT\_RING\_KEYS כשהוא מוצפן ע"י המפתח הפומבי של הלקוח ואליו מצורף אינדקס המפתח. בסיום שליחת כל המפתחות המאסטר שולח הודעת CLIENT\_RING\_END ואז הקליינט עובר למצב CLIENT\_GOT\_KEYS ומפענח את המפתחות שנשלחו בעזרת המפתח הפרטי שלו.



1. מציאת מפתח משותף

כל צומת שקיבל מפתחות שולח הודעת I\_AM\_ON\_THE\_NETWORK , על מנת ששאר הצמתים שנמצאים ברשת יכירו אותו.

כאשר צומת רוצה לדבר עם צומת אחר ברשת, הוא שולח לו הודעת CLIENT\_START\_SESSION עם רשימת אינדקסי המפתחות שלו, הצומת שמקבל את ההודעה בודק חפיפה בין האינדקסים שקיבל לבין האינדקסים שלו, אם מצא מפתח משותף (יקרה בהסתברות של 80%-90%) הוא שולח לצומת הודעת CLIENT\_COMMON\_INDEX שבו המפתח המשותף, במקרה ולא קיים מפתח משותף, יוחזר 1-. שני הצמתים שומרים אצלם במערך השכנים את אינדקס המפתח איתו ידברו ביניהם.



1. רשת בטוחה

כאשר מכשיר רוצה לתקשר עם מכשיר אחר, הוא שולח הודעת MESSAGE\_ENC\_DATA שבו במידע שאותו רוצה להעביר מוצפן ע"י AES, כאשר מפתח ההצפנה הוא המפתח המשותף. המכשיר שמקבל את ההודעה, מפענח בעזרת אותו מפתח.

מכשיר חדש שנכנס לרשת, עובר את שלבי הפרוטוקול ובאפשרותו לתקשר בצורה מאובטחת עם כל מכשיר ברשת. וכן מכשיר שמתנתק מהרשת לא מפריע לתקשורת בין הצמתים האחרים, וכן לא חושף את כל מפתחות ההצפנה.



### נוסחאות מתמטיות

לאחר שנבחר מאסטר ברשת, עליו לחשב את גודל בריכת המפתחות אותה עליו לייצר. חישוב זה אינו טריוויאלי כלל, ולוקח בחשבון גורמים רבים, כמו מס' הצמתים באופן כללי, גודל השכונה - מס' הצמתים ברשת (בטווח הקליטה אחד של השני), גודל הזיכרון הפיזי, גודל כל מפתח, מס' המפתחות שיחולקו לכל צומת, וכמובן ההסתברות הרצויה עבור מפתח משותף בין כל שני צמתים ברשת...

פתירת הנוסחה המתמטית הזו נחשבת לבעיית NP קשה. כיוון שאין אפשרות לבודד את P (גודל הבריכה), אלא יש לבצע קירוב לפתרון ע"י מספר רב של איטרציות, על מספר התחלתי x0, אותו עלינו להגדיר.

על מנת לפצח את הבעיה נעזרנו במתמטיקאי שסייע להעביר את הנוסחאות לmatlab, ולאחר קבלת נוסחה עבור x0 כתבנו הקוד בפייתון.

### הודעות

מבנה:

כל הודעה מכילה את הנתונים הבאים:

* סוג ההודעה: string המוגדר מראש המכיל את סוג ההודעה.
* מזהה המידע: מידע נוסף לגבי תוכן ההודעה. יכול להכיל אינדקס מפתח, iv של AES ועוד.
* מידע (data): תוכן ההודעה. יכול להיות מפתח RSA פומבי, מפתח AES מוצפן ועוד.

סידור לרצף בינארי:

על מנת לשלוח את ההודעה ברשת יש צורך לפרוס את מבנה ההודעה שהוא object להיות רצף תווים בינאריים. נשתמש בספריה pickle כדי לבצע זאת.

שליחה:

ההודעות ברשת נשלחות בפרוטוקול UDP שהוא פרוטוקול תקשורת מהיר אך ללא לחיצת יד ואימות של המידע הנשלח. השתמשנו בUDP מכיוון שרצינו לממש את הפרוטוקול החדש בצורה קלה שמבצעת את התקשורת הנחוצה בלבד ללא עומס מיותר על הרשת ובאופן פשוט למכשירים. בUDP קיימת מגבלה על גודל החבילה הנשלח והוא בערך 556 בתים.

לכן היה צורך לחלק את המידע (הבינארי) בשליחת ההודעה לכמה שליחות. על מנת שהמקבל יידע כמה בתים אמורים להישלח בהודעה מסוימת הגדרנו שכל הודעה מתחילה ב4 בתים- "header" שמכילים את גודל ההודעה, ואח"כ תוכן ההודעה עצמה- "data". השולח מחלק את המידע הזה לבלוקים בגודל שנתמך בUDP ושולח. המקבל מקבל את 4 הבתים הראשונים- header, קורא את גודל ההודעה שאמורה להישלח ומחכה עד לקבלת מספר הבתים שצוינו בheader.

### חוטים (threads)

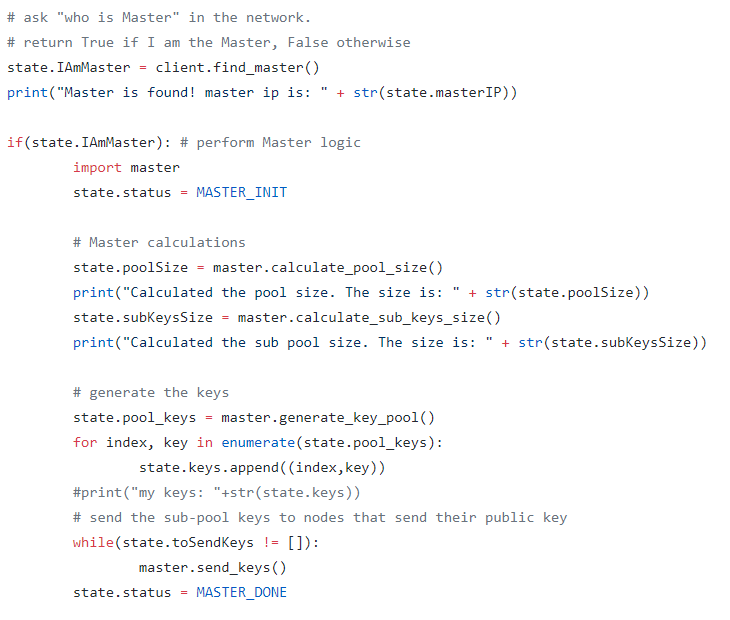
בפרויקט השתמשנו ב-multi-threading:

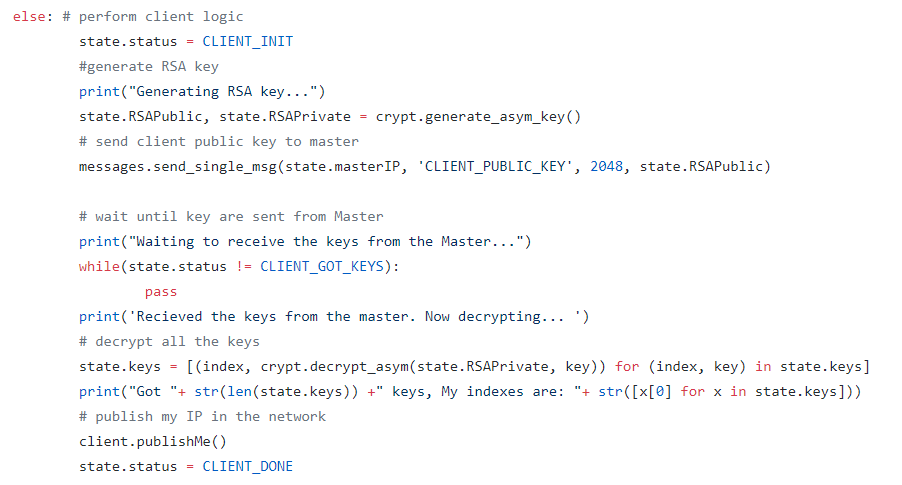
1. עבור האזנה להודעות. כל צומת פותח thread נפרד להאזנה, על מנת שהאזנה להודעות תתאפשר תוך כדי הפעילות השוטפת.
2. כאשר המאסטר שולח לצמתים ברשת את קבוצת תתי המפתחות שלהם, מוצפנים ב-RSA. מכיוון שהצפנתRSA לוקחת זמן, ובייחוד, כאשר עליו לשלוח כמות מפתחות עבור כל צומת, המאסטר יכול לאבד הודעות חשובות כמו הודעת IS\_THERE\_MASTER, ולגרום למופע נוסף של מאסטר. לא ניתן לפתור זאת ע"י שליחה מהthread הראשי מכיוון שתוך כדי שליחה לצומת אחד יכולה להתחיל שליחה לצומת נוסף ואז נזדקק לthread-ים נוספים.

## תיאור המערכת שמומשה

### מימוש הפרוטוקול:

המבנה באופן כללי מוצג כך:





1. אתחול ומציאת מאסטר

המערכת שלנו מורכבת מרשת קטנה של IoT שמחליטים על "מאסטר" באופן דינאמי.



2. חישוב גודל מאגר המפתחות ויצירתו

המאסטר מחשב את גודל בריכת המפתחות שעליו לייצר, ומייצר אותם, מחשב את קבוצת תתי המפתחות שעליו לשלוח לכל מכשיר.



1. שליחת מפתחות באופן מוצפן

הלקוח מייצר מפתח RSA פומבי ופרטי (הלקוח שולח למאסטר את החלק הפומבי).

המאסטר שולח את המפתחות מוצפנים

הלקוח מקבל קבוצה של תתי מפתחות מוצפנים, ומפענח ע"י המפתח הפרטי שיצר.

1. מציאת מפתח משותף

צומת שולח לצומת אחר את אינדקסי המפתחות שלו.

הצומת המקבל מבצע חיתוך עם האינדקסים שלו, ושולח חזרה את אינדקס המפתח המשותף.



שני הצמתים שומרים את האינדקס המשותף.

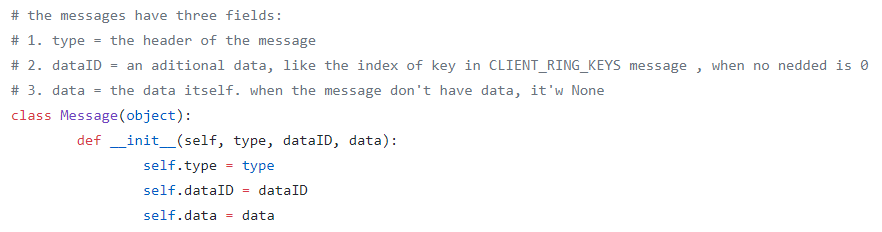
1. רשת בטוחה

כאשר שני צמתים מתקשרים ביניהם הם משתמשים במפתח המשותף כדי להצפין ולפענח הודעות.

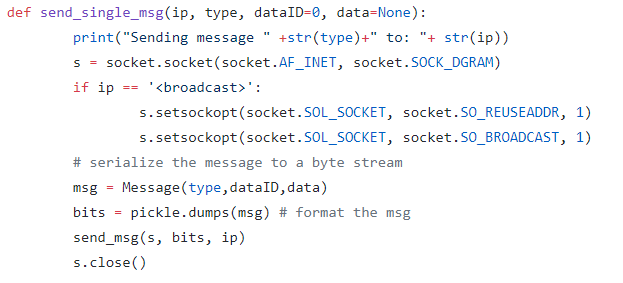
### נוסחאות מתמטיות

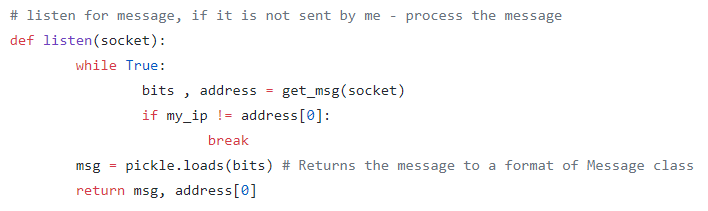
נעזרנו בספריות numpy, scipy על מנת לפתור את הנוסחה.

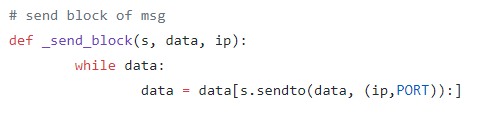
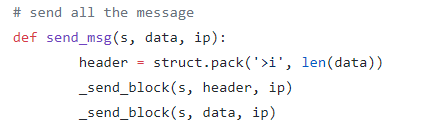
### מימוש שליחת/ קבלת הודעות

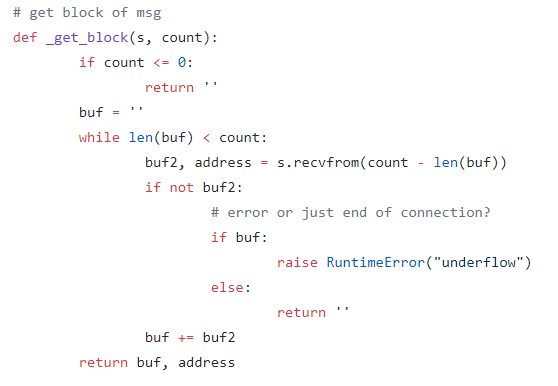
מבנה ההודעה:

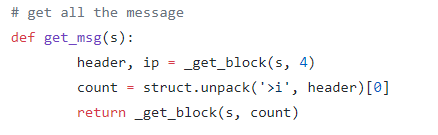
סידור לרצף בינארי בשליחה:



פריסה מחדש לתוך class Message:

שליחת ההודעה:

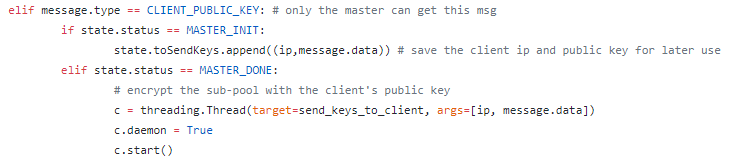
קבלת ההודעה:



### חוטים (threads)

פתיחתthread נפרד עבור האזנה להודעות ברשת:

פתיחתthread עבור שליחת המפתחות מוצפנים ב-RSA:



### מכשירים

במהלך הפרויקט ניסינו לעבוד עם כמה מכשירים מסוגים שונים כמו:

* LinkitSmart
* esp32
* windows pc
* Raspberry Pi 3

חלק ממכשירים אלו היו IoT מהסוג הפשוט ביותר, עם מעט מאד זיכרון, מעבד חלש מאד, ובעיקר הכילו micro-python בלבד. דבר זה לא אפשר לנו להתקין ספריות כבדות כמו numpy, scipy, pyCryptodome.

לאחר מאמצים רבים, עבודה עם סביבות עבודה שונות ומגוון מערכות הפעלה, הגענו למסקנה שאנחנו צריכות לשנות גישה וללכת על מכשירים קצת יותר חזקים שמכילים פייתון בגרסתו המלאה, והחלטנו לעבור להשתמש ב Raspberry Pi 3.

# תכנית בדיקות

בדיקות מקיפות של כל פונקציות המערכת ע"מ לוודא נכונות, מקרי קצה, מקרים חריגים וכו'.

## בדיקות פונקציונליות

* בדיקת הקמת רשת, בה כל הרכיבים מצליחים לתקשר אחד עם השני.
* בדיקה שברשת קיים מנהיג אחד לכולם.
* המנהיג מייצר בריכת מפתחות, בגודל הנדרש לפי האלגוריתם.
* העברת תת קבוצת המפתחות שמועברים באופן מוצפן עוברת בהצלחה.
* כל אחד מהצמתים ברשת מקבל תת קבוצה של מפתחות לפי האלגוריתם.
* כל שני צמתים ברשת חולקים לפחות מפתח אחד משותף, בהסתברות של 90%-80%
* תקשורת מוצפנת בין שני צמתים עוברת בהצלחה.

## בדיקות מערכת

* אינטגרציה בין כל אחד מהשלבים בתהליך הקמת רשת מאובטחת עוברת בהצלחה.
* הרשת יציבה והתקשורת מתבצעת ללא הפרעות.
* המערכת ממשיכה לתפקד לאחר הוספת צומת חדש לאחר הקמת הרשת והפצת המפתחות.
* הרשת ממשיכה לתפקד לאחר כיבוי של אחד הצמתים.

## בדיקות תאימות

* בדיקה שהתהליך עובד על מכשירי IoT בעלי חומרה/תוכנה שונה. כרגע בדקנו על מכשירי raspberry pi ו-pc (בעלות מערכות הפעלה שונות). לכאורה התהליך אמור לעבוד על כל התקן בעל python (ולא micro-python).

## בדיקות תחזוקה (Maintainability)

* האם ניתן לעדכן או לתקן את התכנה אחרי הוצאתה לאור.
* הקוד כתוב בצורה פשוטה, ברורה ומתועדת.

# מסקנות

במהלך הפרויקט אנו שמות דגש על מחשבה ותכנון לפני כל ביצוע בפועל. במיוחד לפני כתיבת הקוד, שם השקענו רבות בהגדרת מצבים ברשת, בצומת "לקוח" ובצומת ה"מאסטר". גילינו שהעלאת החלטות על הכתב וניסוחן מקלה על המימוש, מונעת בלבול ומאפשרת התמקדות בפתרון הבעיה.

לאחר התנסויות בכתיבת התוכנה נוכל להצביע על שתי בעיות עיקריות הקיימות כרגע במימוש שלנו:

* מבנה ההודעות. במהלך כתיבת הפרויקט הרגשנו צורך במבנה קבוע לצורת ההודעות שעוברות ברשת, שיאפשר להכניס ערכים דינאמיים, כמו שליחת מפתחות שונים, או הודעות מסוגים שונים. השתמשנו בספריית שמאפשרת לנו להעביר הודעות במבנה שהוגדר מראש.
* שינינו את ארכיטקטורת מציאת המאסטר. אופן החיפוש אחר המאסטר נעשה בצורה עקומה, שיכל לגרום לריבוי מאסטרים ברשת. כמו כן המצב של מאסטר זמני ומאסטר קבוע יכלה לגרום לשיבושים רבים במערכת, כמו ניתוק מהרשת בזמן ההחלפה, וכך ההחלפה לא עברה או לא עברה בשלמות....
* מבחינת מכשירים. ניסינו לעבוד עם כמה מכשירים, אך זה לא צלח. בהתחלה המגמה הייתה לנסות לעבוד עם מכשירים הכי זולים ופשוטים, אך מכיוון שהמכשירים האלו הכילו micro-python בלבד, לא הייתה אפשרות להתקין ספריות של הצפנה, כי המכשיר לא תמך. לאחר כמה ניסיונות כושלים, והמון עבודה, הוחלט לנסות לעבור למכשירים עם כוחות חישוב גדולים יותר, על מנת שתתאפשר הצפנה.

# סקירת עבודות דומות בספרות והשוואה

סקרנו מס' פתרונות אפשריים שהוצעו עבור אבטחה ב- IoT, אולם לכל אחד מן הפתרונות קיימים חסרונות:

* שימוש בענן

ישנו פתרון אבטחה להתקניIoT המסתמך על שימוש בענן על מנת להשיג את האבטחה הרצויה. כל התקן ברשת יתחבר לשירות בענן שיתמוך בניהול האבטחה של כל ההתקנים, ומולו יתבצעו פעולות האימות וההצפנה[6].

פתרון זה דורש ניהול של שירות בענן, וזו תקורה שדורשת משאבים נוספים, בנוסף, יהיה צורך בהגדרת ספקים אמינים לכזה שירות, ופתרון במקרה של התחזות. במחקר שלנו אנו מנסים להביא את השליטה באבטחה להיות בבלעדיות אצל רשת ההתקנים המקומית, ללא קישור לשירות חיצוני.

* שימוש בTPM:

פתרון זה מסתמך על מכשיר הTPM לביצוע פעולות קריפטוגרפיות ונתינת שירותי הצפנה למכשירים ברשת[7]. זה נראה אמנם רעיון ישים, אך החיסרון בו שדרוש מכשיר TPM כזה עבור כל רשת IoT, וזה גובה עלות נוספת. בפרויקט שלנו אנו מממשים פתרון אבטחה שמתבסס אך ורק על מכשירים הקיימים ברשת, ללא עלות נוספת.

* ארכיטקטורת IoT מאובטחת לערים חכמות המטפלת בפגיעויות במערכות IoT:

הארכיטקטורה כוללת רשתות שחורות ומערכת ניהול מרכזית (KMS) המספקות סודיות, שלמות, פרטיות והפצה מרכזית יעילה. המטרה הייתה לספק שירותי אבטחה הממתנים את הפגיעות של רשתות IoT בשכבות הקישור והרשת, במיוחד עבור נתונים קריטיים.

החסרונות של גישה זו כוללים היעדר פתרון פרטיות להגדרת מיקום המכשיר ואיתור ניתוב חדש עבור צמתים, מה שמוביל לאובדן נתונים.

* ארכיטקטורת SDN לפיתוח יישומי IoT:

ארכיטקטורת SDN אומצה על מנת לספק בסיס לפיתוח מערכת מאובטחת שמאפשרת למנהלי מערכות להציג את העולם באופן גלוי של איומים אפשריים להתקפות ברשת ה- IoT ולספק להם את הזכות לשלוט ברשת מפני האיומים.

עם זאת, אבטחה, מדרגיות ואמינות הן חלק מהחסרונות של רשתות SDN. ההפרדה בין מטוסי הבקרה והנתונים של ה- SDN גורמת לביצועים ירודים בעיבוד חבילות, אשר מובילה לבעיות משמעותיות, כגון עיכוב או אובדן של חבילות והתקפות DoS (DDoS).

* ארכיטקטורת אבטחה חדשה המבוססת על SDN עבור ה- IoT, הידועה גם בשם תחום ה- SDN באמצעות בקרי הגבול:

המחברים תיארו כיצד ניתן להשתמש ב- SDN כדי לחבר בין התקני IoT הטרוגניים, כיצד ניתן לשפר את האבטחה של כל דומיין, וכיצד ניתן לחלק את כללי האבטחה מבלי לפגוע בביטחון של כל תחום.

עם זאת, המחברים לא היו מסוגלים להתמודד עם האתגר של הבטחת תעבורה רצויה ולא רצויה והגנה על הארגון, אשר הם החסרונות העיקריים של שימוש בבקרי הגבול.

* פרוטוקול לניהול מפתחות בצורה קלה:

הפרוטוקול תלוי בהתאמות של רכיבי אבטחה שונים ב- IoT כדי להגדיר ערוצי תקשורת מאובטחים ומוגנים עבור IoT. במהלך העברת הנתונים לאורך הערוץ, הפרוטוקול מבטיח סודיות נתונים ואימות צומת מוגבל.

עם זאת, פרוטוקול האבטחה מוגבל, ואינו מפרט את ההתאמה הנדרשת בין תקורה לתקשורת לבין מספר צדדים שלישיים]8].

מערכות ניהול הפרויקט:

|  |  |  |
| --- | --- | --- |
| # | מערכת | מיקום |
| 1 | מאגר קוד | <https://github.com/reutnagar/distributed-RSA-for-IoT> |
| 2 | יומן | https://trello.com/b/DkjV5sEx/a |
| 3 | סרטון שלב אלפא | https://drive.google.com/file/d/1yG61mZb-n9U0TWnGObKPunGcjKk5-YSS/view |
|  | סרטון סופי |  |

# נספחים

## רשימת ספרות \ ביבליוגרפיה

[1] “אבטחת מידע – ויקיפדיה.” [Online]. Available: https://he.wikipedia.org/wiki/אבטחת\_מידע. [Accessed: 19-Nov-2017].

[2] “RSA – ויקיפדיה.” [Online]. Available: https://he.wikipedia.org/wiki/RSA. [Accessed: 19-Nov-2017].

[3] Wikipedia. (n.d.). *AES Wikipedia*. Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Advanced\_Encryption\_Standard

[4] “האינטרנט של הדברים – ויקיפדיה.” [Online]. Available: https://he.wikipedia.org/wiki/האינטרנט\_של\_הדברים. [Accessed: 19-Nov-2017].

[5] L. Eschenauer and V. D. Gligor, “A key-management scheme for distributed sensor networks,” *Proc. 9th ACM Conf. Comput. Commun. Secur.*, pp. 41–47, 2002.

[6] M. Tao, J. Zuo, Z. Liu, A. Castiglione, and F. Palmieri, “Multi-layer cloud architectural model and ontology-based security service framework for IoT-based smart homes,” *Futur. Gener. Comput. Syst.*, vol. 78, pp. 1040–1051, 2018.

[7] H. Hamadeh, S. Chaudhuri, and A. Tyagi, “Area, energy, and time assessment for a distributed TPM for distributed trust in IoT clusters,” *Integr. VLSI J.*, vol. 58, no. December 2016, pp. 267–273, 2017.

[8] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, “Internet of Things security: A survey,” *J. Netw. Comput. Appl.*, vol. 88, no. March, pp. 10–28, 2017.

## תרשימים וטבלאות

מראה רשת בעלת RSA מבוזר ב"בית חכם". בדוגמה זו המכשיר המנהיג הוא המדפסת. בידיו נמצאת רשימת המפתחות, אותה הוא יחלק לשאר המכשירים בבית באמצעות תקשורת אלחוטית.