מטלת מנחה (ממ"ן) 15

הקורס: תכנות מערכות דפנסיבי - 20937

חומר הלימוד למטלה : יחידות 1-7

מספר השאלות: 2 + 2 בונוס **משקל המטלה:** 14 נקודות

סמסטר: 2022ג **מועד אחרון להגשה:** 10.11.2022

שאלה 1 (80%)

בתרגיל זה תממשו תוכנת שרת ולקוח המאפשרות ללקוחות להעביר קבצים באופן מוצפן מאצלם אל השרת. השרת ייכתב בשפת Python ואילו הלקוח ייכתב בשפת C++.

חשוב!

קראו היטב את כל המטלה לפני תחילת העבודה. וודאו שאתם מבינים היטב את פרוטוקול התקשורת ואת המבנה של תוכנת השרת והלקוח.

ארכיטקטורה

ארכיטקטורת התוכנה מבוססת על שרת-לקוח. הלקוח יוצר קשר ביוזמתו עם השרת, מחליף איתו מפתחות הצפנה ולאחר מכן מעביר לו את הקובץ המבוקש בתקשורת מוצפנת. הלקוח מוודא שהשרת קיבל את הקובץ באופן תקין ע"י השוואת checksum בשני הצדדים, ובמידה ולא עבר באופן תקין, מנסה להעביר שוב (עד 3 נסיונות). בעמוד 3 מתואר תרשים הזרימה של המערכת.

שרת

תפקיד השרת לנהל את רשימת המשתמשים הרשומים לשירות ולאפשר להם להחליף ביניהם הודעות מסוגים שונים.

א. השרת יכתב בשפת python 3.90 ומעלה.
ב. השרת יתמוך בריבוי משתמשים ע"י תהליכונים (threads) או ע"י selector.
ג. גרסת השרת תהיה **3.**
ד. השרת יפעל עם חבילת הצפנה Crypto.Cipher

פורט

השרת יקרא את מספר הפורט <u>מתוך קובץ טקסט</u> בצורה הבאה :

- שם הקובץ : port.info
- מיקום הקובץ : באותה תיקיה של קבצי הקוד של השרת
- תוכן הקובץ : מספר פורט
  לדוגמא :
  1234

נתונים

השרת ישמור את נתוני הלקוחות והקבצים שנשמרו בזיכרון (RAM). בנוסף, הוא יחזיק בסיס נתונים SQLite שיכלול טבלת רשימת המשתמשים, שמות מפתחות הצפנה שנשלחו להם, וטבלת רשימת הקבצים שהתקבלו מהם, והאם הקובץ עבר אימות מוצלח מול הלקוח בעזרת checksum. כמו כן יחזיק תיקיה מקומית שתכלול את הקבצים שיתקבלו מלקוחות.

שמירת הנתונים תעשה ע"י טבלאות SQL בקובץ בשם server.db.

מידע על הלקוחות ישמר בטבלה בשם clients. מבנה הטבלה:

שם	סוג	הערות
ID	16 בתים (128 ביט)	מזהה ייחודי עבור כל לקוח. אינדקס
Name	מחרוזת (255 תוים)	מחרוזת ASCII המייצגת שם משתמש. **כולל תו מסיים!** (null terminated)
PublicKey	160 בתים	מפתח ציבורי של לקוח
LastSeen	תאריך ושעה	הזמן בו התקבלה בקשה אחרונה מלקוח
מפתח AES	128 ביט	מפתח AES שנשלח ללקוח

מידע על הקבצים שהתקבלו יישמר בטבלה בשם files. מבנה הטבלה:

שם	סוג	הערות
ID	16 בתים (128 ביט)	מזהה ייחודי עבור כל לקוח.
File Name	מחרוזת (255 בתים)	מחרוזת ASCII המייצגת שם קובץ כפי שנשלח מהמשתמש. **כולל תו מסיים!** (null terminated)
Path Name	מחרוזת (255 בתים)	מחרוזת ASCII המייצגת מסלול יחסי ושם קובץ כפי שמאוחסן בתיקיה שרת. **כולל תו מסיים!** (null terminated)
Verified	בוליאני	האם checksum אומת בהצלחה מול הלקוח

אופן פעולת השרת

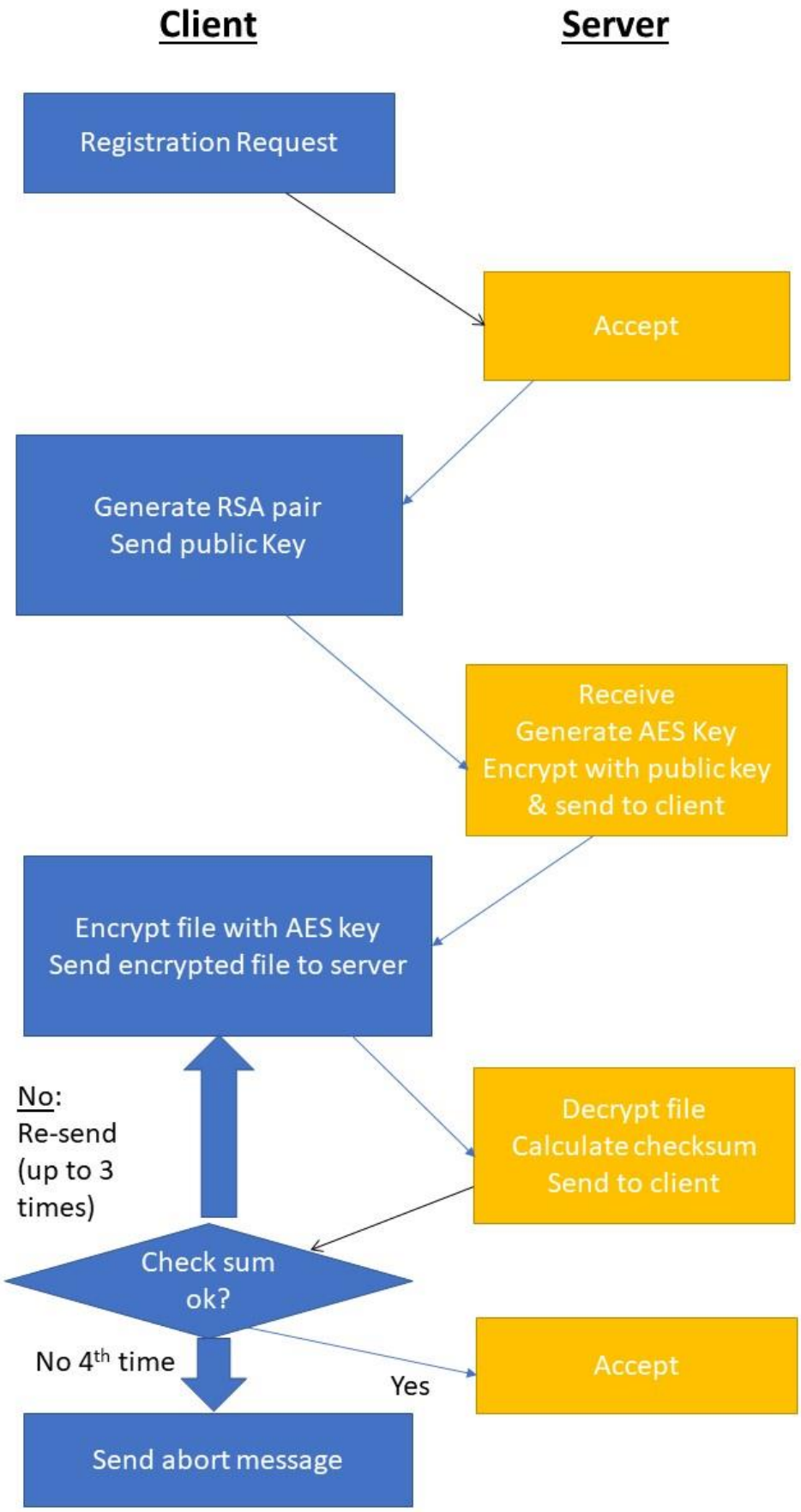
1. קורא את הפורט מתוך הקובץ port.info. (אם הקובץ לא קיים, להוציא אזהרה ולעבוד על פורט ברירת מחדל 1234. לא להגיע לנפילה עם Traceback במידה והקובץ לא זמין.)
2. ממתין לבקשות מלקוחות בלולאה אין סופית.
3. בעת קבלת בקשה מפענח את הבקשה בהתאם לפרוטוקול:

א. בקשה לרישום: במידה ושם המשתמש המבוקש כבר קיים, השרת יחזיר שגיאה. אחרת, השרת ייצר UUID חדש עבור המשתמש, ישמור את הנתונים בזיכרון ובבסיס הנתונים ויחזיר תשובת הצלחה.

ב. מפתח ציבורי מלקוח ייקלט ויעודכן בבסיס הנתונים. בתגובה, ייצור השרת מפתח AES, יצפין אותו בעזרת המפתח הציבורי וישלח בחזרה ללקוח.

ג. הודעה עם קובץ מוצפן: השרת יפענח את הקובץ המוצפן בעזרת מפתח ה-AES המקורי שנשלח לאותו לקוח, ויחשב את ה-CRC. החישוב יתבצע באופן זהה לפקודת cksum בלינוקס:
https://www.howtoforge.com/linux-cksum-command/
לצורך חלק זה בלבד, הסטודנטים רשאים להוריד קוד קיים מהאינטרנט או לממש את חישוב ה-cksum עצמאית, ובלבד שיהיה תואם לתוצאת החישוב בפקודת cksum בלינוקס, וכמובן בין הלקוח והשרת. לאחר החישוב בשרת יישלח ה-CRC ללקוח לאימות.

ד. השרת יקבל הודעת הצלחה מהלקוח (CRC אומת) או שליחה חוזרת של הקובץ עד 3 פעמים.


Client
Server
Registration Request
Accept
Generate RSA pair
Send public Key
Receive
Generate AES Key
Encrypt with public key
& send to client
Encrypt file with AES key
Send encrypted file to server
No:
Re-send
(up to 3
times)
Decrypt file
Calculate checksum
Send to client
Check sum
ok?
No 4th time
Yes
Accept
Send abort message

לקוח

תוכנת הלקוח תדע לתקשר מול שרת, להירשם (במידה ולא רשום מהפעלה קודמת), להחליף איתו מפתחות הצפנה ולאחר מכן להעביר אליו באופן מאובטח קובץ מהלקוח שיאוחסן בשרת. הלקוח אינו מתקשר או מודע ללקוחות אחרים במערכת.

א. תוכנת הלקוח תיכתב בשפת C++, ותיבדק אצלנו בעזרת Visual Studio 2019.
ב. הלקוח יפעל על פי סדר פעולות קבוע, כך שניתן להפעילו במצב Batch mode.
ג. הלקוח יתבסס על הצפנה בעזרת חבילת CryptoPP.
ד. גרסת הלקוח תהיה 3.

קובץ הנחיות ללקוח

- שם הקובץ: transfer.info
- מיקום הקובץ: בתיקיה של קובץ ההרצה (.exe)
- תוכן הקובץ: שורה ראשונה – כתובת IP + נקודתיים + מספר פורט
- שורה שניה – שם הלקוח (מחרוזת עד 100 תוים)
- שורה שלישית – מסלול הקובץ לשליחה לשרת.
- <u>דוגמא</u>:

  127.0.0.1:1234
  Michael Jackson
  New_product_spec.docx

שם ומזהה ייחודי[1]

הלקוח ישמור ויקרא את השם והמזהה הייחודי שלו מתוך קובץ טקסט בצורה הבאה:

- שם הקובץ: me.info
- מיקום הקובץ: בתיקיה של קובץ ההרצה (.exe)
- תוכן הקובץ:

  שורה ראשונה: שם
  שורה שניה: מזהה ייחודי בייצוג ASCII כאשר כל שני תווים מייצגים ערך hex בעל 8 סיביות.
  שורה שלישית: מפתח פרטי שנוצר בריצה הראשונה של התוכנית בפורמט בסיס 64.
  לדוגמא:

Michael Jackson 64f3f63985f04beb81a0e43321880182 MIGdMA0GCSqGSIb3DQEBA…

שגיאה מצד השרת

בכל מקרה של שגיאה הלקוח ידפיס למסך הודעה: "server responded with an error" וינסה לשלוח את ההודעה שוב, עד 3 פעמים, ואם עדיין לא יצליח, ייצא עם הודעת Fatal מפורטת.

[1] בתרגיל זה נעשה שימוש במזהה ייחודי גלובלי (UUID). לקריאה נוספת:
https://en.wikipedia.org/wiki/Universally_unique_identifier

<u>פעולות אפשריות</u>:

בקשת רישום

1. במידה והקובץ me.info לא קיים, הלקוח יקרא שם משתמש מהקובץ transfer.info וישלח בקשת רישום לשרת.
2. הלקוח ישמור בקובץ בשם **me.info** את השם והמזהה הייחודי שיקבל מהשרת.
   שימו לב! במידה והקובץ כבר קיים הלקוח לא יירשם שנית.

מפתח ציבורי

הלקוח ייצר זוג מפתחות RSA, ציבורי ופרטי, וישלח את הציבורי לשרת. וישלח אותו לשרת. בתגובה השרת אמור לשלוח מפתח AES שהוצפן בעזרת המפתח הציבורי.

קבלת מפתח AES והצפנת הקובץ

לאחר שהלקוח מקבל את מפתח ה-AES, הוא פותח את המפתח בעזרת המפתח הפרטי של ה-RSA וקולט את מפתח ה-AES. בתגובה הוא מצפין בעזרתו את הקובץ שהוא נדרש להעביר, ושולח את הקובץ המוצפן לשרת. במקביל, הוא אמור לחשב את ה-CRC של הקובץ כדי שיוכל להשוות אותו ל-CRC שמתקבל מהשרת.

אימות השליחה בעזרת CRC

השרת אמור לקלוט את הקובץ המוצפן מהלקוח, לפתוח את ההצפנה בעזרת מפתח ה-AES, ולחשב גם הוא את ה-CRC ולשלוח אותו ללקוח לאימות.

פרוטוקול התקשורת

כללי

- הפרוטוקול הוא בינארי וממומש מעל TCP.
- כל השדות המספריים חייבים להיות עם ערכים גדולים מאפס (**unsigned**) ומיוצגים כ- **little endian**
- פרוטוקול זה תומך **בבקשות** לשרת **ותשובות** ללקוח. בקשות או תשובות יכולות להכיל **"הודעה"**.
- הודעה עוברת בין לקוחות

זכרו! הפרוטוקול <u>מחייב</u> ולא ניתן לעשות בו שינויים. כפועל יוצא, כל שרת ולקוח הממשים את הפרוטוקול יכולים לעבוד אחד מול השני.

רישום למערכת

1. כל לקוח שמתחבר בפעם הראשונה נרשם בשירות עם שם (מחרוזת באורך מקסימלי של 255 בתים) ומעביר את המפתח הציבורי שלו
2. השרת יחזיר ללקוח מזהה ייחודי שנוצר עבורו או שגיאה אם השם כבר קיים בבסיס הנתונים.

פרטי הפרוטוקול

בקשות

מבנה בקשה מהלקוח לשרת. השרת יפענח את התוכן (payload) לפי קוד הבקשה.

בקשה לשרת

Request	שדה	גודל	משמעות
כותרת (Header)	Client ID	16 בתים (128 ביט)	מזהה ייחודי עבור כל לקוח
	Version	בית	מספר גירסת לקוח
	Code	2 בתים	קוד בקשה
	Payload size	4 בתים	גודל תוכן הבקשה
תוכן (payload)	payload	משתנה	תוכן הבקשה. משתנה בהתאם לבקשה

תוכן (payload)

התוכן משתנה בהתאם לבקשה. לכל בקשה מבנה שונה.

קוד בקשה 1100 – רישום

שדה	גודל	משמעות
Name	255 בתים	מחרוזת ASCII המייצגת שם משתמש. **כולל תו מסיים!** (null terminated)

* **שימו לב:** השרת יתעלם מהשדה Client ID

קוד בקשה 1101 – שליחת מפתח ציבורי

שדה	גודל	משמעות
Name	255 בתים	מחרוזת ASCII המייצגת שם משתמש. **כולל תו מסיים!** (null terminated)
Public Key	160 בתים	מפתח ציבורי של לקוח

קוד בקשה 1103 – שליחת קובץ

שדה	גודל	משמעות
Client ID	16 בתים	מזהה ייחודי של הלקוח השולח
Content Size	4 בתים	גודל הקובץ (לאחר הצפנה)
File Name	255 בתים	שם הקובץ הנשלח
Message Content	משתנה	תוכן הקובץ. **מוצפן** ע"י מפתח סימטרי.

קוד בקשה 1104 – CRC תקין

שדה	גודל	משמעות
Client ID	16 בתים	מזהה ייחודי של הלקוח השולח
File Name	255 בתים	שם הקובץ הנשלח

קוד בקשה 1105 – CRC לא תקין, שולח שוב (לאחר מכן תגיע שוב בקשה 1103)

שדה	גודל	משמעות
Client ID	16 בתים	מזהה ייחודי של הלקוח השולח
File Name	255 בתים	שם הקובץ הנשלח

קוד בקשה 1106 – CRC לא תקין בפעם הרביעית, סיימתי

שדה	גודל	משמעות
Client ID	16 בתים	מזהה ייחודי של הלקוח השולח
File Name	255 בתים	שם הקובץ הנשלח

תשובות:

תשובה מהשרת

Response	שדה	גודל	משמעות
כותרת (Header)	Version	בית	מספר גירסת שרת
	Code	2 בתים	קוד התשובה
	Payload size	4 בתים	גודל תוכן התשובה
תוכן (payload)	payload	משתנה	תוכן התשובה. משתנה בהתאם לתשובה

קוד תשובה 2100 – רישום הצליח

שדה	גודל	משמעות
Client ID	16 בתים	מזהה ייחודי של לקוח

קוד תשובה 2101 – רישום נכשל

קוד תשובה 2102 – התקבל מפתח ציבורי ושולח מפתח AES מוצפן

שדה	גודל	משמעות
Client ID	16 בתים	מזהה ייחודי של לקוח
מפתח סימטרי מוצפן	משתנה	מפתח AES מוצפן ללקוח

קוד תשובה 2103 – קובץ התקבל תקין עם CRC:

שדה	גודל	משמעות
Client ID	16 בתים	מזהה ייחודי של הלקוח השולח
Content Size	4 בתים	גודל הקובץ (לאחר הצפנה)
File Name	255 בתים	שם הקובץ הנשלח
Cksum	4 בתים	CRC

קוד תשובה 2104 – מאשר קבלת הודעה, תודה

הצפנה

פרוטוקול התקשורת משתמש בהצפנה סימטרית על מנת לקודד את ההודעה בין הלקוחות ובהצפנה אסימטרית על מנת להחליף מפתח בין הלקוחות.

בתרגיל זה השתמשו בספריה Crypto++[2] **(ראו נספח א')**

הצפנה סימטרית

עבור הצפנה סימטרית השתמשו ב- AES-CBC.

אורך המפתח **128 ביט**. ניתן להניח שה- IV מאופס תמיד (הזיכרון מלא באפסים).

שימוש כזה ב- IV לא בטוח אם משתמשים באותו מפתח בכל פעם, אך לצורך הממן הוא מספק.

הצפנה אסימטרית

עבור הצפנה אסימטרית השתמשו ב- RSA.

אורך המפתחות **1024 ביט**.

שימו לב: הספריה Crypto++ מחזיקה מפתחות ציבוריים בפורמט X509[3]. פורמט זה מכיל Header לפני המפתח עצמו וערכים נוספים. לכן, גודלו הסופי (בצורה בינארית) הוא **160 בתים** (עבור מפתחות בגודל שונה גודלו הסופי של המפתח ישתנה בהתאם).

דגשים לפיתוח

1. מומלץ לעבוד עם מערכת לניהול קוד (כדוגמת גיט[4])
2. עבדו באופן מודולרי ובדקו את עצמכם כל הזמן
   א. זהו את המחלקות והפונקציות החשובות
   ב. **בצד השרת:**
   כיתבו קוד לטיפול בבקשה אחת. הוסיפו תמיכה בריבוי לקוחות בשלב מאוחר יותר
   ג. **בצד הלקוח:**
   ממשו את הרכיבים הגדולים באופן בלתי תלוי בחלקים אחרים של המערכת (תקשורת, הצפנה, פרוטוקול וכו').
3. ממשו קוד לבדיקה כבר בשלבים מוקדמים של הפרוייקט
   א. **בצד השרת:**
   השתמשו בהדפסות למסך או בכתיבה ללוג כדי לעקוב אחרי התקשורת. תוכלו גם לטעון את המודול לתוך ה- interpreter ולעבוד באופן דינמי.
   ב. **בצד הלקוח:**
   כיתבו פונקציות קטנות שבודקות חלקים נפרדים של המערכת. השתמשו בפונקציות הללו תוך כדי כתיבת הקוד עצמו.
4. כתיבת הקוד
   א. ממשו את התוכנה לפי עקרונות תכנות מונחה עצמים
   ב. שימו לב לייצוג ערכים בזיכרון כ- little-endian או big-endian
   ג. הקפידו על תיעוד של הקוד (comments)

---

[2] https://www.cryptopp.com/
[3] https://en.wikipedia.org/wiki/X.509
[4] https://www.atlassian.com/git/tutorials/what-is-version-control

ד. תנו שמות משמעותיים למשתנים, פונקציות ומחלקות. המנעו ממספרי קסם!
ה. הודעה יכולה להיות גדולה מאוד (בגודל דינמי). חשבו על הדרך הנכונה ביותר לקבל ולשלוח כמות מידע גדולה.
ו. **אבטחת מידע** – חשבו לאורך כל הדרך על כתיבת קוד בטוח לפי העקרונות שלמדתם: האם בדקתם את הקלט? איך נעשה שימוש בזיכרון דינמי? האם מתבצעת המרת טיפוסים (casting) וכו'..

5. **לפני ההגשה**

א. בדקו שהפרוייקט מתקמפל ורץ בצורה תקינה ללא קריסות או תלויות בספריות שונות (למעט הספריות הנדרשות לתרגיל)
ב. מומלץ לייצר תיקיה חדשה ולהעתיק לשם את הקבצים המיועדים לשליחה. לייצר פרוייקט VS חדש, לקמפל ולהריץ
ג. העבודה תבדק על מ"ה חלונות עם Visual Studio Community 2019

דגשים לקוד שרת:

1. השתמשו בפייתון **גירסה 3**
2. עשו שימוש בספריות פייתון הסטנדרטיות בלבד!
3. תוכלו להעזר בספריה **struct** על מנת לעבוד עם נתוני התקשורת בנוחות

דגשים לקוד לקוח:

1. מומלץ (אבל לא חובה) לעשות שימוש בספריות STL
2. ניתן ורצוי להשתמש ביכולות C++11 (לדוגמא פונקציות מסוג למדה, שימוש ב- auto וכו'..).
3. למימוש התקשורת עשו שימוש ב- winsock או בספרית boost

הגשה

שרת

1. עליכם להגיש רק את קבצי הקוד (כלומר קבצי py.).
שימו לב! על התוכנית להטען ולרוץ בצורה תקינה (ללא צורך בתוספות קבצים וללא קריסות).
2. יש לכלול פונקציה ראשית בשם **main**. פונקציה זו תהיה הפונקציה הראשית של תוכנית השרת והיא תעבוד לפי אופן פעולת השרת המפורט לעיל.
טיפ:
תוכלו להשתמש במנגנון הבא כדי לאפשר עבודה אינטראקטיבית **וגם** הרצה של הקוד

```
if __name__ == "__main__":
```

לקוח

1. עליכם להגיש רק את קבצי הקוד (כלומר קבצי h. ו- cpp.).
שימו לב! על התוכנית לרוץ בצורה תקינה (ללא צורך בתוספות קבצים, ללא קריסות)
2. עבודתכם תיבדק במערכת הפעלה חלונות, באמצעות Visual Studio ולכן מומלץ לעבוד עם סביבה זו.

ניתוק והתאוששות

המטלה דורשת מהלקוח לשמור את פרטי הרישום שלו בקובץ me.info, כדי לאפשר חזרה לפעילות של לקוח שכבר נרשם לאחר סיום פעילות. מסיבה דומה השרת מחזיק database הכולל מידע של לקוחות שנרשמו, כדי שבמקרה של נפילה והפעלה מחדש, לקוחות רשומים יוכלו לחזור לפעילות על בסיס מפתח RSA קיים מבלי שיצטרכו להירשם מחדש.

עם זאת, כדי לשמור על פשטות המטלה והיקף עבודה סביר בסמסטר קיץ, אין צורך לממש בפועל את תהליך ההתאוששות או חידוש תקשורת (דבר שהיינו דורשים במערכת אמיתית) אלא רק לשמור בבסיס הנתונים את הנתונים שנדרשו.

שאלה 2 (20%)

עליכם לנתח את הפרוטוקול המוצע בשאלה 1 ולמצוא בו חולשות פוטנציאליות. יש להגיש מסמך מחקר המפרט את החולשות שמצאתם, התקפות אפשריות והצעה לתיקון.

הגשה

מסמך word או pdf .

הערה: את כלל קבצי המערכת יש לארוז לקובץ zip.