**יישומים מאובטחים – תרגיל תכנות**

**פקודות הרצה בעבור שלבים א-ג**

**Question A**

keytool -genkeypair -alias NadavKey -keyalg RSA -validity 365 -keystore Nadav.keystore -storetype jceks -storepass Nadavkeystore -keypass nadavkey

**Question B**

keytool -genkeypair -alias MiyaKey -keyalg RSA -validity 365 -keystore Miya.keystore -storetype jceks -storepass Miyakeystore -keypass miyakey

**Question C**

keytool -exportcert -alias NadavKey -keystore Nadav.keystore -file NadavCertificate.cert -storepass Nadavkeystore -keypass nadavkey

keytool -exportcert -alias MiyaKey -keystore Miya.keystore -file MiyaCertificate.cert -storepass Miyakeystore -keypass miyakey

keytool -importcert -alias NadavKey -file NadavCertificate.cert -keystore Miya.keystore -storepass Miyakeystore

keytool -importcert -alias MiyaKey -file MiyaCertificate.cert -keystore Nadav.keystore -storepass Nadavkeystore

**כיצד מריצים את התוכנית –**

על מנת להריץ את התוכנית יש להריץ את קובץ ה-Jar באמצעות הפקודה הרלוונטית ואליה להוסיף שלושה פרמטרים –

1. מצב התוכנית – הצפנה או פענוח, על מנת להיות במצב הצפנה נקיש – encrypt, על מנת להיות במצב פענוח נקיש – decrypt.
2. סיסמת ה-keystore הרלוונטית. בעבור הצפנה נרצה את הסיסמא של ה-keystore של נדב – Nadavkeystore, בעבור פענוח נרצה את הסיסמא של ה- keystore של מיה – Miyakeystore.
3. סיסמת ה-key של ה-alias המתאים. בעבור הצפנה נרצה את הסיסמא מה-keystore של נדב – nadavkey, בעבור פענוח נרצה את הסיסמא מה- keystore של מיה – miyakey.

דוגמאות לפרמטרים–

encrypt Nadavkeystore nadavkey

decrypt Miyakeystore miyakey

**דוגמת הרצה לתוכנית –**

הרצת קובץ ה-Jar:

**A screenshot of a cell phone

Description automatically generated**

קובץ הטקסט שנרצה להצפין:

A screenshot of a cell phone

Description automatically generated

הקובץ המוצפן:

A screenshot of a cell phone

Description automatically generated

הקובץ המפוענח:

A screenshot of a cell phone

Description automatically generated

קובץ הקונפיגורציה:

A screenshot of a cell phone

Description automatically generated

**הסבר על הקוד ובחירת האלגוריתמים**

תוכנית ההצפנה עושה שימוש במספר אלגוריתמים –

ראשית, כמבוקש בהוראות התבצע שימוש באלגוריתם AES על מנת לאתחל את אובייקט ה-cipher שמצפין את הקובץ הגלוי ובחרנו לעבוד במוד CBC. בחרנו לעבוד עם אלגוריתם זה משום שבמקרה בו ישנה חזרה של בלוקים בגלוי הם לא יבואו לידי ביטוי בבלוקים בסתר. כלומר, יהיה יותר קשה למצוא דפוסים חוזרים בהצפנה ובכך להצליח לחדור אותה בקלות. זה נעשה באמצעות IV שמאפשר את מניעת החזרות.

את המפתח איתו אותחל ה-Cipher הצפנו באמצעות הצפנה א-סימטרית RSA (על מנת להעביר את המפתח מבלי לתאם פגישה), השתמשנו במפתח הציבורי של צד ב׳ על מנת להצפין את המפתח כך שרק צד ב׳ יוכל לפענח בעזרת המפתח הפרטי שלו את המפתח אותו הצפנו ולהשתמש בו ולא אף גורם אחר. השימוש באלגוריתם מאפשר לנו זאת. (בפענוח יעשה שימוש במפתח הפרטי של צד ב׳ שגלוי ושייך רק לו ורק הוא יודע אותו).

בעבור החתימה הדיגטלית חתמנו באמצעות ShA256withRSA על מנת לוודא כי ההצפנה מתבצעת כראוי, ותישמר השלמות והסודיות – כלומר שהקובץ שנשלח מצד א׳ אכן תקין ולא שונה בדרך לצד ב׳.

תוכנית הפענוח מבצעת שימוש באותם האלגוריתמים על מנת לפענח את המפתח לצורך פענוח הקובץ – שמירה על סודיות, והשוואה התוכן והחתימה הדיגטלית – שמירה על שלמות.

שימוש באלגוריתמים שונים בין התהליכים יוביל לתוצאה לא נכונה.

* בחירת האלגוריתמים התבצעה דרך קובץ ה-properties במידה ויש רצון לשנות את אחד מן האלגוריתמים ניתן לבצע שינוי בקובץ הנקרא – prop.properties.