

הפקולטה למדעי המחשב

בניית יישומים מאובטחים

מס' קורס 3536

סמסטר 1 (חורף) 2014-2015

ד"ר דוד מובשוביץ

הגשת תרגיל בית מס' 5

**תרגיל תכנות**

|  |
| --- |
| **מגיש** |
| שלו דוד |
| Shalev David |
| ת.ז. 036764751 |

1. מפתח פרטי וציבורי עבור צד א' נוצר ע"י Keytool ע"י שימוש בפקודה

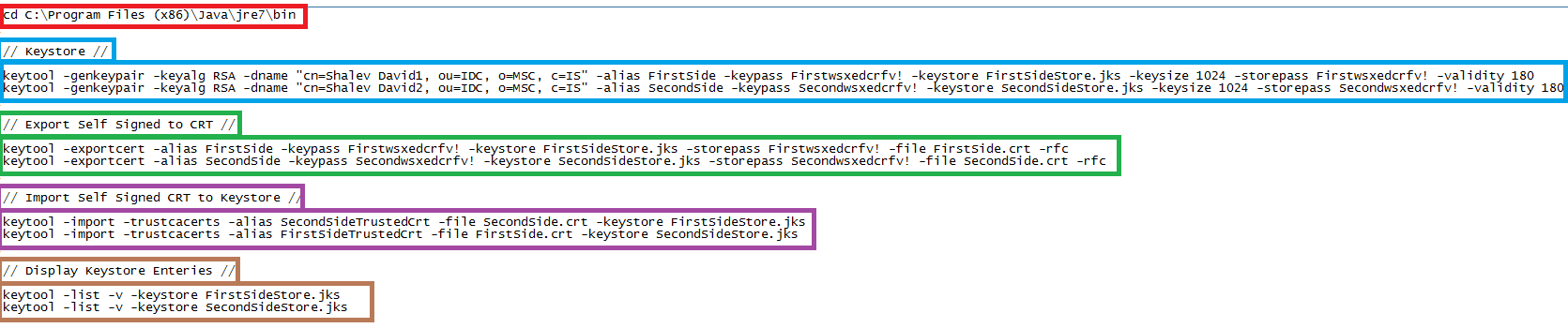
keytool -genkeypair -keyalg RSA.

פקודה זו מייצרת צמד של מפתח פרטי ופומבי, כאשר עבור הפומבי היא מייצרת self-signed certificate.

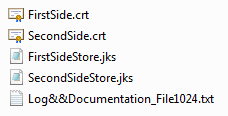
תיעוד מלא של הפקודות בסדר כרונולוגי מופיע בהמשך.

1. מפתח פרטי וציבורי עבור צד ב' נוצר ע"י באופן דומה
2. יוצר self-signed Certificate לכל אחד מהצדדים בנפרד ע"י פקודת keytool עם פרמטר ,-exportcert אשר מייצא את ה – self-signed certificate מה-Keystore לקובץ crt. את קובץ ה – crt מייבאים לאחר מכן ע"י פקודת keytool עם פרמטר –import, ל – keystore של הצד הנגדי.
   1. הערה: self-signed הינו certificate הרגיש להתקפת MITM (אדם העומד באמצע יכול לייצר מפתח פרטי וציבורי ולחתום על הציבורי באופן עצמאי עם הפרטי ללא קושי ושלוח המפתח עם החתימה ללא קושי), אך הוא נוצר כך היות וזו מה שהתבקשנו בתרגיל. בפועל ניתן להפעיל פקודות ב – Keytool ליצירת certificate מאושר (CA) וזהו דבר נוסף שחזק ב – keytool ואשר מעניק מוטיבציה להשתמש בו.

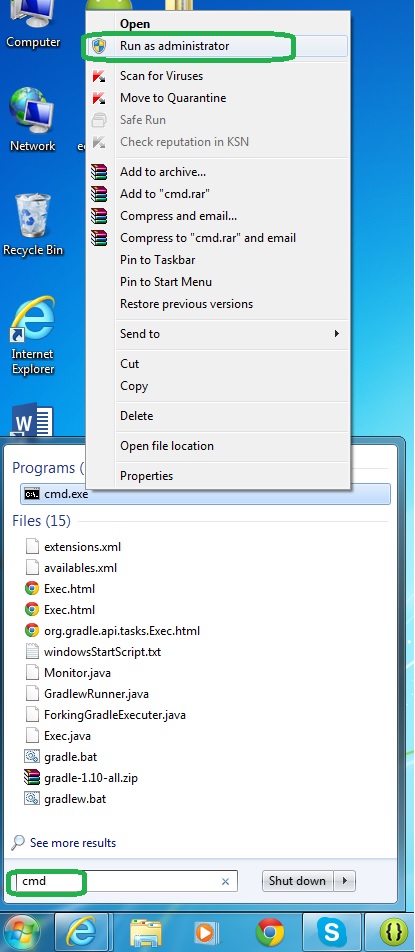
להלן סיכום הפקודות בסדר כרונולוגי ולוג של תוצרי keytool:



תוצרי ה - keytool הם קבצי ה – jks וה- crt:



אופן פתיחת חלון cmd לפקודות keytool יבוצע ע"י הרצתו בהרשאות מנהל כמתואר:



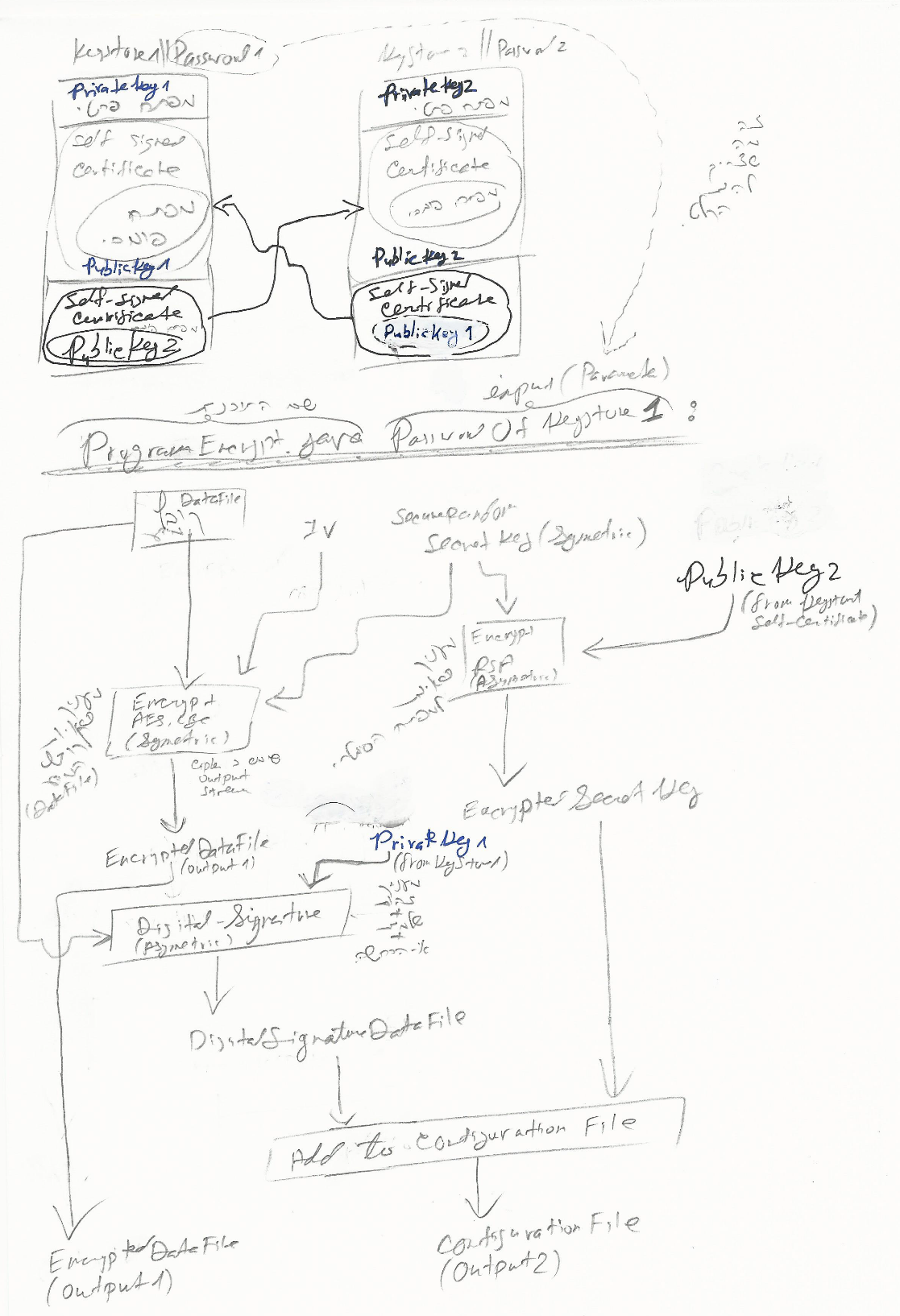
תכולת קובץ הלוג בה ניתן לראות את התכנים של ה – KeyStore-ים:



1. תוכנת הצפנת וחתימת קבצים:

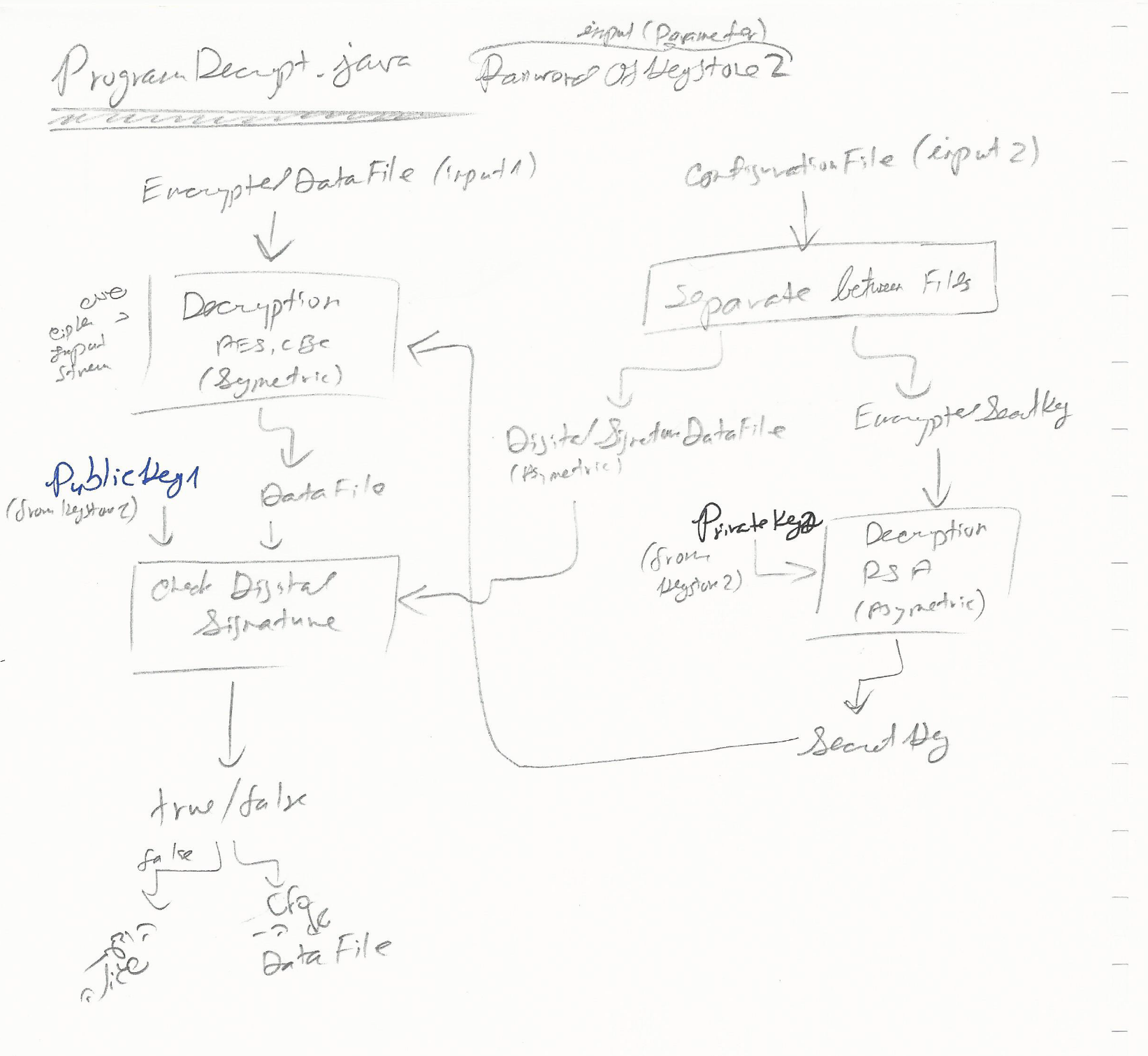
זהו בעצם צד א'. צד זה, בהינתן ה-keystore **שלו בלבד** (לאחר סעיפים 1-3( מייצר מפתח סימטרי שיוצפן אסימטרית ויועבר לצד ב' (תוך שימוש בפומבי של צד ב' השמור ב – keystore של צד א'). מפתח סימטרי זה משמש להצפנת קובץ plaintext שיישלח לצד ב'. המפתח הפרטי האסימטרי של צד א' ישמש אותו לחתימת קובץ plaintext (חתימה דיגיטאלית אסימטרית) שתישלח לצד ב' גם היא.

להלן תרשים המתאר את אופן פעולת התוכנית וזרימת המידע בה באופן תמציתי:



1. תוכנת פענוח ואימות חתימה דיגיטאלית:

זהו צד ב'. צד זה, בהינתן ה – keystore **שלו בלבד** (לאחר סעיפים 1-3) מפענח את המפתח הסימטרי (הנשמר בקובץ קוניפיגורציה). לאחר מכן, באמצעות המפתח הסימטרי המפוענח, הוא מפענח את ה – ciphertext, ומאמת התאמתו לחתימה הדיגיטאלית (שנשמרה בקובץ קונפיגורציה).



**הערות כלליות:**

* תוצאות של שתי הרצות שונות ניתן לראות בקבצים המצורפים בתיקיית Inputs&outpus – ניתן לראות קידוד מוצלח ופענוח מוצלח וכי המפתח הסימטרי וה-IV אכן הוגרלו אקראית.
* תכנת ההצפנה ותכנת הפענוח, כל אחת מהן מקבלת כארגומנט את סיסמת קובץ ה – keystore ובודקת התאמה.
* בתצורה הנוכחית של התרגיל, קובץ ה – ciphertext וקבצי הקונפיגורציה נשמרים בתיקייה משותפת לשתי התוכנות העונה לשם Input&Output. יצוין, כי קיימת מודעות והבנה כי בפועל התוכנות יושבות על שני מחשבים שונים ומומלץ לשלוח את המידע אחד לשני באמצעות SSL/TLS ולהשתמש בהגנות נוספות בשכבות הלא אפליקטיביות.
* בוצע ניסיון להשתמש ב – XML Encryption לניצול תכונת ה canonicalization ולשמור את כל המידע הרלוונטי בקובץ קונפיגורציה בודד (שחלקו מוצפן וחלקו גלוי) אך המחשב ו/או תכנת ה – eclipse שהותקנו עליו לא תמכו באופן מלא והדבר הסתבך.
  + לעומת זאת, ניסיון לבצע XML Encryption ב – C# צלח היטב ובנספח מופיעה דוגמא לביצוע XML Encryption ב – C#.
* בחירת provider-ים:
  + בוצע מחקר מקדים על רשימת Provider-ים מוכרים והשוואה ביניהם, אך באופן מאכזב, המחשב ו/או תכנת ה- eclipse שהותקנו עליו, לא אפשרו בחירה מיטבית של Provider בהתאם למחקר זה. מעבר לכך, גם בתוך Provider מסוים היו אלגוריתמים שלא נמצאו (נזרק Exception מסוג Unsupported Algorithm) למרות שבדיקה באינטרנט מראה באופן וודאי כי האלגוריתם אכן קיים עבור אותו Provider.

יצוין כי באופן כללי השאיפה היא לנסות לעבוד עם Provider BC (Bouncy Castle) מטעמי תאימות, פופולריות משתמשים, וקיומו הן ב – Java והן ב – C#.

מצ"ב קישור לרשימת ה – Provider-ים המומלצים כפי שנמצא במקורות האינטרנט השונים:

* <http://en.wikipedia.org/wiki/Comparison_of_Cryptography_Libraries>
* <http://stackoverflow.com/questions/2927952/why-do-people-use-bouncycastle-instead-of-java-cryptography-extension-what-is-t>

**רשימת מקורות:**

Key-tool:

* <http://www.herongyang.com/Cryptography/keytool-Export-Key-printcert-Command.html>
* <https://www.sslshopper.com/article-most-common-java-keytool-keystore-commands.html>
* <http://alvinalexander.com/java/java-keytool-keystore-certificates>

Provider-ים:

* <http://en.wikipedia.org/wiki/Comparison_of_Cryptography_Libraries>
* <http://stackoverflow.com/questions/2927952/why-do-people-use-bouncycastle-instead-of-java-cryptography-extension-what-is-t>

Java API:

* <http://docs.oracle.com/javase/tutorial/security/apisign/index.html>
* <https://docs.oracle.com/javase/7/docs/api/javax/crypto/package-summary.html>

C# .Net XML Encryption:

* <http://msdn.microsoft.com/en-us/library/ms229740(v=vs.110).aspx>