ב"ה

בנית ישומים מאובטחים - תשע"ה: תרגיל תכנותי ב Java Crypto API

א. יצרו מפתח פרטי וציבורי עבור צד א' (עבור אלגוריתם RSA) באמצעות השימוש בתוכנית keytool
http://docs.oracle.com/javase/6/docs/technotes/tools/solaris/keytool.html
על המפתח הפרטי שנשמר ב Keystore המיוצר באמצעות Keytool להיות מוגן באופן האוטימלי האפשרי

ב. יצרו מפתח פרטי וציבורי עבור צד ב' (עבור אלגוריתם RSA) באמצעות השימוש בתוכנית keytool , על המפתח הפרטי שנשמר ב Keystore המיוצר באמצעות Keytool להיות מוגן באופן האוטימלי האפשרי

ג. כל אחד משני הצדדים יוצר באמצעות ה Keytool תעודה דיגיטלית מסוג Self-Signed Certificate ו"מעביר" אותה לצד השני. כל צד שמקבל את התעודה הדיגיטלית של עמיתו טוען אותה ל Keystore כ Trusted certificate.

כל הפעולות בשלבים א-ג יעשו באמצעות command line ויש לתעד את ה command lines בהם השתמשתם עבור כל אחד מהשלבים לעיל. את השלבים הבאים יש לעשות תוך שימוש ב keystores שיצרתם.

ד. בנו תוכנית להצפנת וחתימת קבצים.

- התוכנית קוראת קובץ גלוי, יוצרת קובץ חדש המכיל את תוכן הקובץ המקורי מוצפן
- התוכנית מחשבת חתימה דיגיטלית אסימטרית של תוכן הקובץ ושומרת אותו בקובץ קונפיגורציה המצורף לקובץ המקורי
- התוכנית מבצעת כתיבה של קובץ באופן מוצפן באמצעות שימוש ב CipherOutputStream
- את ה Cipher יש לאתחל לשימוש באלגוריתם AES במוד CBC תוך שימוש ב IV אקראי
- את המפתח לאלגוריתם ההצפנה יש להגריל באופן אקראי תוך שימוש בפונקציה המתאימה
- את המפתח יש להצפין בהצפנה אסימטרית תוך שימוש באלגוריתם ה RSA
- את המפתח המוצפן כמו גם פרמטרים מוספים אפשר לשמור בקובץ הקונפיגורציה שיצורף לקובץ המוצפן (שבו תשמר גם החתימה הדיגיטלית של הקובץ המוצפן)
- את המפתח הציבורי שידרש להצפנת המפתח הסימטרי יש לקרוא מה KeyStore המתאים
- את המפתח הפרטי שידרש לחתימה הדיגיטלית יש לקרוא מה KeyStore המתאים (את הסיסמא יש לקבל כפרמטר לתוכנית)

ה. בנו תוכנית לפיענוח ובדיקת חתימה דיגיטלית של קבצים מוצפנים וחתומים.

- התוכנית קוראת את הקובץ המוצפן מפענחת אותו ובודק את השלמות שלו
- את הפרמטרים לפענוח הקובץ ולבדיקת החתימה הדיגיטלית התוכנית המפענחת קוראת מקובץ קונפיגורציה שהוכן ע"י התוכנית המצפינה

- התוכנית מבצעת קריאה של קובץ באופן מוצפן באמצעות שימוש ב CipherInputStream
- התוכנית תבדוק את השלמות של הקובץ לאחר הפענוח שלו
- את המפתח הציבורי שידרש לבדיקת החתימה הדיגיטלית יש לקרוא מה KeyStore המתאים
- את המפתח הפרטי שידרש לפענוח הקובץ יש לקרוא מה KeyStore המתאים (את הסיסמא יש לקבל כפרמטר לתוכנית)
- בהנחה שבדיקת השלמות של הקובץ (על פי החתימה דיגיטלית) תקינה, התוכנית תיצור קובץ פלט עם התוכן הגלוי שפוענח (אם בדיקת התוכן נכשלה התוכנית תכתוב למסך ולקובץ הודעת שגיאה)

הערות:

- התוכניות צריכות להכתב על פי הכללים המקובלים של הנדסת תוכנה
- התוכניות צריכות להכתב באופן שניתן להחליף את האלגוריתמים ואת ה Crypto providers בקלות
- יש לצרף את קובץ הקוד ולתעד אותו באופן שמסביר כיצד השתמשתם בכל אחד מה API ומדוע
- יש לבחור Crypto Provider ולהסביר מדוע בחרתם מדוע בחרתם להשתמש דווקא בו
- יש להסביר את הבחירה של האלגוריתמים בהם בחרתם להשתמש
- יש לשלוח את קובץ הקוד המתועד, את ה JAR המאפשר להריץ את התוכנית כולל הוראות כיצד להריץ את התוכנית וכולל ה Keystore
- יש לשלוח פלט ההרצה של התוכנית המאפשר לראות שהתוכנית עובדת כהלכה
- הניקוד של התרגיל יהיה כדלקמן:
  - נכונות המימוש מבחינה קריפטוגרפית (כולל אופן השימוש ב Java Crypto API) – 35%
  - קוד בנוי באופן שמאפשר לבחור אלגוריתם ולבחור Provider בקלות – 7%
  - שימוש נכון ב Keytool וב Keystore –8%
  - הסבר מפורט של השיקולים שהנחו אתכם במימוש ובבחירת האלגוריתמים –10%
  - תיעוד של הקוד –10%
  - הנדסת תוכנה - 15%
  - העובדה שהקוד רץ ומיצר תוצאות נכונות (כולל דוגמא לפלט הרצה)- 15%

הנחיות הגשה:

- ניתן להכין את התרגיל בזוגות
- יש להגיש את התרגיל עד יום א׳ ה 28 לדצמבר 2014
- את כל התוצרים הנדרשים בתרגיל יש לשלוח בדוא״ל לכתובת tausec2014@gmail.com
- יש לציין את השם ומס׳ תז של המגיש(ים) בשם הקובץ המוגש ובכותרת הדוא״ל

בברכה

ד״ר דוד מובשוביץ