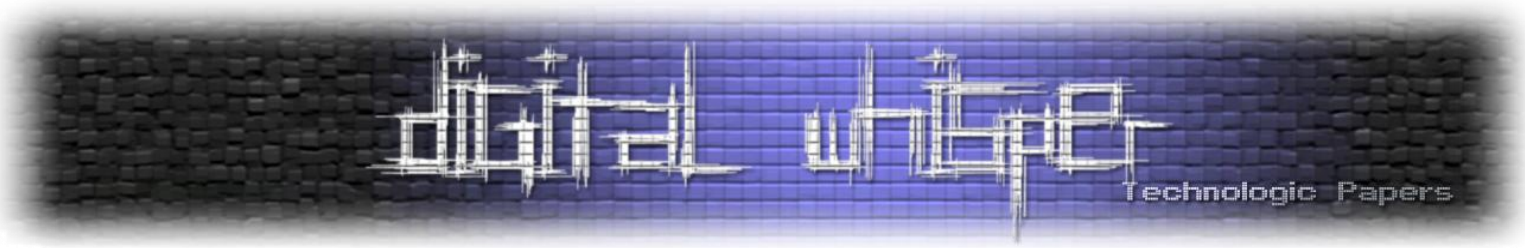


PKI, תעודות, כרטיסים חכמים ומה שבינהם

מאת הלל חימוביץ' (HLL)

הקדמה

אחד הנושאים המעניינים ביותר שקיימים , לדעתי, הוא נושא אימות והצפנת הנתונים, והפתרונות המוצגים לכל סוגיה או פרצה בתהליך זה. כבר שנים רבות שקיימים פתרונות מקיפים לזיהוי, אימות, חתימה והצפנה של נתונים ושל גורמים, בהם ניתן להשתמש ללא חשש שהמקור זויף או שונה.

במאמר זה אציג לכם מספר רעיונות מרכזיים הקשורים ב'ארכיטקטורת מפתח ציבורי' או – Public Key Infrastructure.

בעיה מספר 1 – חיסיון המידע – Data Confidentiality

תיאור הבעיה

בעולם ההצפנה ישנם שני חברים טובים מאוד ומוכרים מאוד, אליס ובוב. במקרה שלנו, אליס ובוב רוצים לדבר אחד עם השני במסנג'ר (אודיגו, IRC, מה שתבחרו) ולתכנן לחברם דני מסיבת הפתעה. לרוע מזלם של בוב ואליס, דני עובד בספקית השירות של בוב והוא יכול לראות את כל הנתונים היוצאים ונכנסים בשיחה בין בוב ואליס. בוב ואליס מחפשים פתרון איך למנוע מדני להבין מה נאמר בשיחה שלהם.

לבעיה בסיסית זו וריאציות רבות:

- מצב בו בוב מחובר ברשת אלחוטית ציבורית.
- בחברה שבה עובדת אליס, יש רכיב שמצותת לתקשורת בין המחשב שלה לאינטרנט (במקור תפקידו נועד למנוע ממידע רגיש לצאת, או ממזיקים – להכנס).
- אליס חוששת מרשת ה-Ethernet כי היא יודעת שדני שעובד בעמדה לידה, יודע לבצע מתקפת ARP Poisoning \ Man In The-Middle.

בעית חיסיון המידע: כיצד נמנע מגורם זר שמצותת לקו או לערוץ הפומבי, לצפות במידע שעובר באותו הקו/הערוץ.

פתרונות לבעית חסיון המידע

באופן בסיסי הפתרון לבעית חסיון המידע הוא הצפנת המידע או הסתרתו בצורה כזו שדני לא ידע מה המפתח על מנת שיוכל לקרוא את הנתונים מאליס או מבוב.

פתרון ראשון הוא שימוש בצופן סימטרי. **צופן סימטרי** הוא אלגוריתם הצפנה המשתמש במפתח אחד הן להצפנה והן לפענוח הטקסט . הבעיה: כל הצפנה סימטרית תחייב הסכמה מוקדמת של אליס ובוב על המפתח – ולא, יהיו חייבים בוב/אליס לציינו בתקשורת, ובהנחה שדני מאזין לתקשורת, לדני יהיה את המפתח הנ"ל והוא יוכל לקרוא ללא כל בעיה את השיחה ביניהם באמצעות המפתח שאיתר מהתקשורת.

אליס ובוב צריכים להחליט ביניהם בצורה מסוימת על מפתח משותף מבלי שדני יוכל לדעת מהו. שני חוקרים, Whitfield Diffie ו-Martin Hellman, הם הראשונים שהציעו פתרון לבעיה זו.

לא נכנס לעומק למתמטיקה מאחורי הפתרון, אך נציג את העקרון של הפתרון:

1. שני הצדדים בוחרים בסיס משותף g
2. כל צד בוחר מעריך משלו לבסיס זה, a ו-b.
   a. אליס מגרילה את a ושולחת את g^a לבוב
   b. בוב מגריל את b ושולח את g^b לאליס.
3. כל צד מעלה בחזקה את הנתון שנשלח אליו במספר שהוא הגריל, ועל כן אליס מקבלת את הערך (g^b)^a, ובוב מקבל את הערך (g^a)^b. כאשר לבסוף כמובן שני הערכים האלה זהים לפי חוקי חזקות. כעת בוב ואליס יכולים לנהל ערוץ תקשורת מאובטח באמצעות מפתח **סימטרי** זה.

הסיבה שדני לא יכול לעלות על מפתח זה מכיוון שהוא רואה רק את התוצאות הסופיות של החישובים שבוב שולח לאליס, ואליס לבוב. דני יכול לעלות על המעריכים אולם לצורך כך הוא יזדקק לכוח מחשוב עצום לבצע פעולת log (ההופכית לחזקה) עבור מעריך ראשוני גדול מאוד.

לבעית חסיון המידע, ובפרט, למציאת מפתח משותף על גבי ערוץ פומבי, קיימות שיטות נוספות בהן ניתן להשתמש. המתעניינים יכולים להעמיק בשיטה בשם Quantum cryptography. המתעניינים בהרחבה נוספת בנושא שיטת Diffie-Hellman יכולים לקרוא עליה כאן: Diffie-Hellman_key_exchange

ומה קורה במקרה של תקשורת לא ישירה?

נניח ואליס ובוב לא מדברים במסנג'ר, אלא במייל, כאשר כל אחד בודק את המייל שלו פעם ביום. אם הם ישתמשו באלגוריתם Diffie-Hellman, היה צורך ב שלושה ימים רק כדי להקים את ערוץ התקשורת. במקרים בהם אין תקשורת ישירה, ובמצבים נוספים אחרים, ישנו פתרון אחר לשליחת הודעות סודיות על גבי ערוץ פומבי, והוא נקרא הצפנה א-סימטרית. דוגמא להצפנה כזו היא RSA.

הצפנה א-סימטרית משתמשת בכלים מתמטיים הלקוחים מתורת המספרים לייצור של זוג מפתחות, כאשר האחד משתמש להצפנת הנתונים, והשני לקריאתם, כאשר:

$$C = E(K_{pub}, P)$$
$$P = D(K_{pri}, C)$$

- C – טקסט מוצפן
- P – טקסט קריא
- Kpub – מפתח ציבורי של נמען ההודעה
- Kpri – מפתח פרטי של נמען ההודעה

ובעברית פשוטה, נאמר שההצפנה מתבצעת עם מפתח אחד - המפתח הציבורי, והפענוח של הנתונים מתבצע עם מפתח אחר - המפתח הפרטי.

בוב ואליס כל אחד מייצר להם זוג מפתחות כזה, וכל אחד שולח את המפתח הציבורי שלו לצד השני. כאשר אליס רוצה להגיד לבוב מה היא מתכננת במסיבת ההפתעה, היא מצפינה את ההודעה **במיוחד עבור בוב** עם המפתח הציבורי שבוב שלח לה, ושולחת אותה לדרכה.

כעת, האדם היחיד שיכול לקרוא את ההודעה הוא בוב מכיוון שבוב הוא היחידי שמחזיק את המפתח שמסוגל לפתוח את ההצפנה. כאשר בוב רוצה לשלוח הודעה חזרה לאליס הוא מצפין את ההודעה **במיוחד עבור אליס** עם המפתח הציבורי שאליס שלחה לו, ושולח אותה לדרכה.

בעיות מספר 2 ו-3 – מהימנות המידע ושלמות המידע – Data Authenticity and Integrity

בפתרון שהצגנו מעלה, בין אם באלגוריתם Diffie-Hellman ובין עם הצפנה א-סימטרית, יש בעיה נוספת. כאשר בוב מדבר עם אליס הוא אינו יכול לוודא שדני אינו שינה את תוכן ההודעה ששלחה אליס. לדוגמא, אם אליס שלחה הודעה שהמסיבה תתקיים בשעה 4, אך דני ישנה את תוכן ההודעה ויציין שזו שעה 8, בוב לא ידע שההודעה שונתה.

בעיה זו נקראת בעית **<u>שלמות המידע</u>** – כיצד ניתן לוודא שהמידע שנשלח אכן לא שונה בדרך (או נפגם)?

בנוסף, בוב הוא בחור מאוד ספקן. כאשר הוא מקבל הודעה - איך הוא יכול לדעת שזו אליס שבאמת שלחה את ההודעה? אולי זה בעצם דני שמתחזה לאליס, ובמצב כזה בוב יחשוב שהוא מקים ערוץ תקשורת מאובטח/לא מאובטח ישיר מול אליס, אבל בפועל הוא משוחח עם דני המתחזה לאליס. דני ישלח לבוב את כל מה שהוא צריך בשביל שבוב יאמין שדני הוא אליס, ויקים עם בוב תקשורת באותה דרך שאליס הייתה עושה זו.

בעיה זו נקראת בעית **<u>מהימנות המידע</u>** – כיצד ניתן לוודא שהמידע שנשלח אכן נשלח מהמקור שהוא מתיימר להיות?

לצורך הבהרת העניין נעזוב לרגע בצד את ההצפנה של ההודעה. נניח שקיימת הודעה T שבוב רוצה לשלוח לאליס. פתרון נאיבי יעבוד כך: בוב כותב את ההודעה במלואה, ובסופה מציב נתון בן 10 בתים שהוא הסכום הכולל של כל התווים בהודעה. כאשר אליס מקבלת את ההודעה, היא יכולה לוודא שאינה שונתה באמצעות ביצוע חישוב זהה לבתים הנמצאים בהודעה והשוואה של התוצאה שיצאה לה, למה ששלח בוב.

האומנם? למרות שבוב מנסה לעזור לאליס לוודא את שלמות ההודעה, דני הוא בחור חכם. דני רוצה לשנות את ההודעה. הוא רואה את הערך הנקוב ב-10 הבתים ששלח בוב, ומשנה את ההודעה (או כותב הודעה חדשה) בצורה כזו שסכום הבתים בהודעה יהיה זהה לערך שנקב בו בוב. כאשר אליס תנסה לוודא את שלמות ההודעה (המפוברקת מדני) של בוב, היא לא תגלה שום פגם. דני מסוגל לעשות את זה, מכיוון שחיבור היא פעולה הפיכה בצורה פשוטה, וכוח המחשוב הנדרש לחשב אותה הוא נמוך למדי.

אם כך, הוספת הסכום הכולל של תווי ההודעה לגוף ההודעה אינו פתרון לבעיה שלנו. יש צורך בטכניקה מתוחכמת יותר.

Hash או Message Digest הוא אלגוריתם מתמטי שבעזרתו ניתן לייצג הודעה שלמה במספר מועט של בתים, בצורה כזו ש:

- קל יחסית לחשב את ה-Hash עבור הודעה מסוימת.
- בהינתן Hash מסוים, בלתי אפשרי לחשב את ההודעה המקורית בזמן סביר.
- לא ניתן לשנות את ההודעה בלי לשנות את ה-Hash שלה, אולם, בהינתן זמן ארוך מאוד, ניתן לשנות את ההודעה בצורה כזו שתניב Hash זהה.
- לא ניתן למצוא שני הודעות שיש להם אותו Hash, בזמן סביר (דומה לסעיף הקודם)

כל אחד מהסעיפים הנ"ל הוא **אפשרי לביצוע**, אך לא במסגרת זמן סבירה. מכאן אנחנו יוצרים פעולת Hash שמכוונת להיות פעולה חד כיוונית שלא ניתן להפוך אותה. עבור אלגוריתם Hash אידיאלי – תמיד נצטרך לבצע מתקפת Brute-Force על כל הקומבינציות האפשריות כדי למצוא הודעה עם Hash זהה, או כדי למצוא הודעה מתוך Hash נתון.

הרחבה: כאשר אומרים שאלגוריתם Hash נפרץ, לא הכוונה היא שאפשר להפוך אותו. לדוגמא כאשר MD5 נפרץ, הכוונה היתה לא להשיג את ההודעה המקורית מתוך ה-Hash , אלא – קיצור זמן משמעותי של מציאת הודעה שונה בעלת אותו Hash כאשר ההודעה המקורית זמינה (נקרא גם Hash Collision).

מנגנוני ה-Hash חיסלו את האפשרות שדני יוכל ליצור הודעה שונה, אך בעלת 'סכום תווים' זהה. כפי שבטח הספקתם לשים לב, 'סכום התווים בהודעה' הוא גם סוג של Message Digest, אך איננו עומד בקריטריונים שכתובים מעלה.

אבל מה הבעיה פשוט לשנות את ה-Hash המצוין בהודעה?

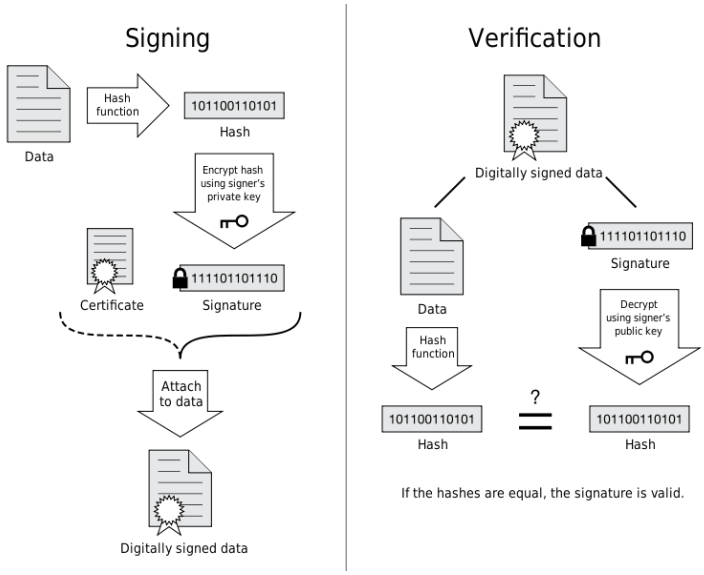
גם לאחר שימוש ב-Hash יכול דני פשוט **להמציא הודעה חדשה, וגם לשנות לה את ה-Hash**. כך אליס לעולם לא תוכל לשים לב שההודעה שונתה, למרות שאלגוריתם ה-Hash הוא בטוח לחלוטין.

איך יכולה אליס לוודא שה-Hash שנשלח לה אכן הגיע מבוב ולא מדני? – כאן נכנסת בעית **מהימנות המידע**. כדי לפתור בעיה זו ניצלו את מנגנוני ההצפנה **הא-סימטרית** כפי שהוסברה קודם לביצוע של פעולה חדשה נוצצת בתחום הקריפטוגרפיה ואבטחת המידע, והיא נקראת **חתימה דיגיטלית.**

חתימה דיגיטלית – Digital Signature

חתימה דיגיטלית היא סוג של 'הצפנה הפוכה' שעובדת רק על ה-Hash של ההודעה, משמע – המידע עובר פעולת Hashing ואותו ה-Hash **נחתם** עם המפתח הפרטי של שולח ההודעה, מקבל או מקבלי ההודעה באשר הם יוכלו לאמת את ההודעה, את מקוריותה ואת שלמותה ע"י בדיקת החתימה – ביצוע Hash בעצמם להודעה, ולהשתמש במפתח הציבורי של שולח ההודעה יחד עם החתימה הדיגיטלית ע"מ לאמת את ה-Hash הנ"ל.

פעולת החתימה הינה בעצם פעולת ההצפנה של ה-Hash באמצעות המפתח הפרטי של המוען, היא אינה נקראת הצפנה מכיוון שהיא לא נועדה להסתיר את הנתונים מאף אחד, אלא רק לאמת את אותנטיות הנתונים, וכי כל אחד יכול לפתוח את הצפנה זו כי המפתח הציבורי הוא פומבי לכל.



נלקח מ'וויקיפדיה'

סיכום ביניים

אז מה היה לנו עד כה? אליס ובוב רוצים לנהל שיחה פרטית. לצורך כך, בתור התחלה, הם רוצים להיות בטוחים עם מי הם מדברים . לפני השיחה הם מתחילים באימות אחד את השני באמצעות חתימה דיגיטלית. לאחר מכן, על בסיס האמון הזה אליס ובוב מבצעים את החלפת המפתחות (ניתן לראות זו בצורה כזו שתוצאות החישוב של בוב ותוצאת החישוב של אליס בתהליך ה-DH כאשר הם שולחים אחד לשני – חתומות דיגיטלית באמצעות המפתח הפרטי של הצד ששלח את תוצאת החישוב).

אחרי שהבנו מה מהות החתימה הדיגיטלית, מה הרעיון מאחורי אלגוריתם Diffie-Hellman ומהי הצפנה א-סימטרית נמשיך בהצגת בעיות נוספות.

הפתרונות לבעיות שהוצגו קודם יפות ואסטטיות, אולם אינן מושלמות, הפתרונות האלו עדיין לא פותרים לנו את הבעיה הראשונית – בעיית חלוקת המפתחות.

הצגנו פתרון לבעיית המהימנות, אך אליה וקוץ בה, חדי השכל ישימו לב, שע"י פתרון החתימה הדיגיטלית הבעיה לא נפתרה, אלא רק קיבלה פנים אחרות. השאלה המתבקשת היא **כיצד נחליף מפתחות פומביים, ושהצד השני ידע שאנחנו, זה אנחנו?** או אם להשליך את הסיטואציה על אליס ובוב: כיצד בוב יכול לוודא שהמפתח הציבורי שנשלח לו כביכול ע"י אליס, הוא באמת של אליס. כיצד תוודא אליס שהמפתח הציבורי שבוב שולח לה הוא באמת ובתמים של בוב, ולא של גורם זדוני המיירט את התקשורת?

אמון - Trust

במדינת ישראל הדרך שבה מזוהים אנשים לראשונה הוא לפי תעודות הזהות המנוילנות שלהם (לפחות נכון לאוקטובר 09). ננית לרגע **שלא היה ניתן לזייף** את תעודת הזהות, ושהיא יכולה להיות **מיוצרת רק ע"י משרד הפנים**. האם כאשר מישהו יגיע אליכם, ויטען שהוא האדם הנקוב בתעודת הזהות שהוא מביא עימו, האם תאמינו לו? ואם כן - מדוע?

התשובה לשאלה מדוע אתם מאמינים לו היא פשוט מאוד – **אתם סומכים על הגורם שהנפיק לו את התעודה** - אתם סומכים על משרד הפנים.

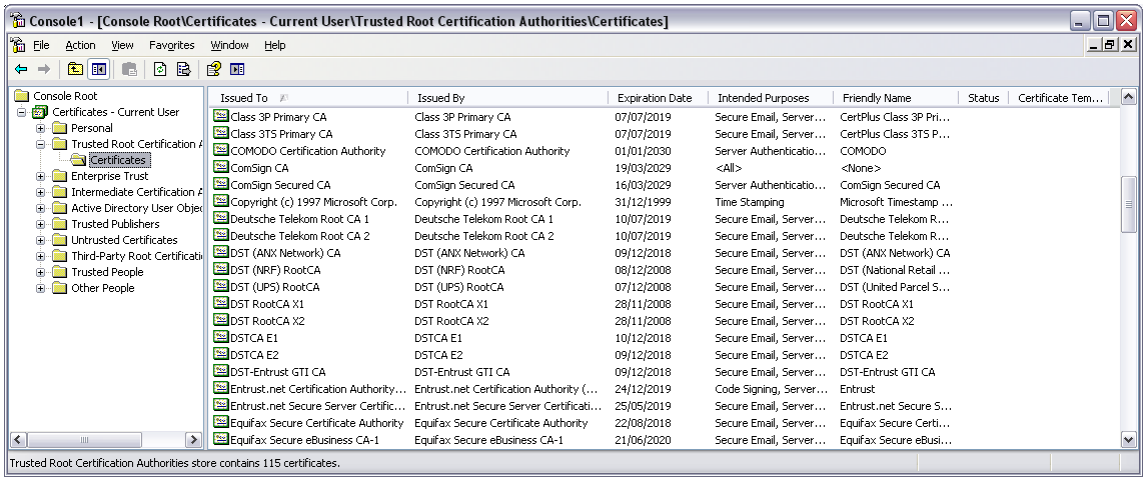
אם רק משרד הפנים יכול להנפיק תעודות זהות, ואם אתם סומכים על משרד הפנים שיעשה את העבודה הנדרשת בשביל לזהות מי זה מי, אתם יכולים **<u>להיות בטוחים ב-100% שהאדם העומד מולכם הוא מי שהוא מצהיר שהוא</u>**.

בעולם ה-PKI, קיים אותו רעיון, קיים גורם צד שלישי, Trusted Third-Party (להלן 'משרד הפנים') שהוא מפיק תעודת זהות דיגיטלית לגורם המגיע אליו ומבקש להנפיק לו תעודה, אותו גורם אחראי לוודא שהאדם שהגיע אליו הוא אכן הוא ומפיק לו תעודה.

תעודה דיגיטלית - Digital Certificate

תעודה דיגיטלית אינה ניתנת לזיוף בזמן סביר בגלל השימוש במתמטיקה שהוצגה בפרק הראשון. הגורם שהנפיק את התעודה, גם לו חייבת להיות תעודה כזו, וגם לזה שהנפיק לו את התעודה יש תעודה משל עצמו, וכך הלאה עד אשר מדובר בשורש העץ, באבא של כל התעודות –Root Certificate Authority.

תעודת האבא **לא ניתנת לאימות** ללא מידע מוקדם, על כן, תעודות Root CA צריכות להגיע **בדרך כלשהיא** למחשבים השותפים לאותו האמון, בין אם באמצעות Disk-On-Key, Floppy, Group Policy, הורדה והתקנה ידנית (אך מסוכנת מאוד) מהאינטרנט. הדרך שמקובלת היום באינטרנט היא **שהתעודות של ה-Root CAs העולמיים מגיעות מובנות עם מערכת ההפעלה של המשתמשים.**



האוסף הדיפולטיבי של Trusted Root CA המגיע עם Windows XP 64bit.

אז מהי תעודה דיגיטלית בעצם?

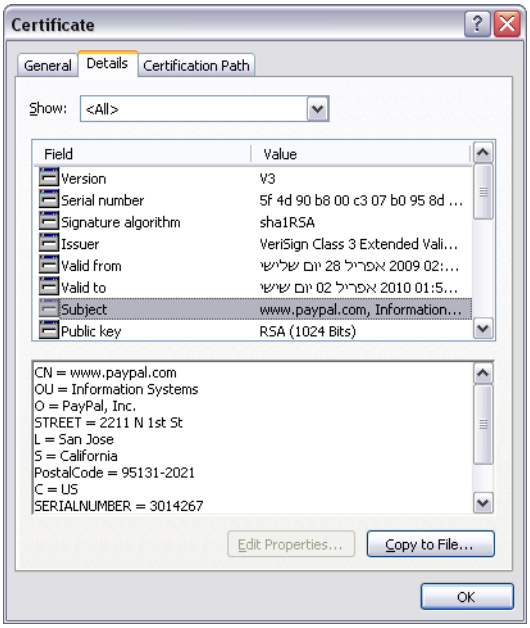
תעודה דיגיטלית הינה אוסף של פרמטרים ונתונים החתומים דיגיטלית ע"י מנפיק התעודה. בין הפרמטרים האלה חובה להיות קיימים **שם בעל התעודה** וכן **המפתח הציבורי שלו**.

אם שם בעל התעודה מופיע יחד עם המפתח הציבורי של בעל התעודה, והנתונים האלה **יחדיו** חתומים דיגיטלית ע"י גורם אחר שעליו אנו סומכים, ויש לנו את המפתח הציבורי שלו לאימות החתימה. **בהכרח** שהמפתח הציבורי המצוין שייך לאדם המצוין (ראו בעיית שלמות מידע ומהימנות מידע).

אם אליס תקבל תעודה כזו מבוב, היא יכולה להיות בטוחה שהמפתח הציבורי של בוב הוא הנקוב בתעודה.

גם אם לא בוב, אלא מישהו אחר ישלח את התעודה, המפתח הציבורי הוא עדיין של בוב ורק בוב יכול לקרוא הודעות שהוצפנו עם המפתח הציבורי המצוין. רק בוב יכול להפיק חתימות דיגיטליות שיאומתו באמצעות המפתח המצוין.

אם גורם אחר ינסה לשנות את אחד הפרמטרים – הוא לא יוכל לעשות זאת בגלל שאז החתימה לא תהייה תקפה. **החתימה הדיגיטלית בתעודה 'שומרת' על שלמות ואמינות הנתונים המופיעים בה ומצליבה את שם בעל התעודה עם המפתח הפומבי שלו**.



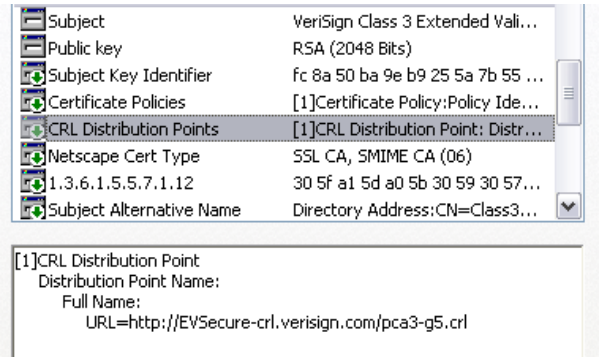
חלק מפרטי תעודה דיגיטלית לדוגמא

Certificate Revocation List

נאמר לרגע שאתם עובדים בארגון שבו יש 1000 עובדים וכולם עובדים מהבית עם תעודות דיגיטליות שהונפקו ע"י החברה, ומותקנות על המחשבים הביתיים של העובדים. בתעודה הזו משתמשים גם בכדי לזהות את העובדים מול השרת וגם את העובדים מול העובדים האחרים (השותפים לאותו אמון - Trust). יום אחד נפרץ המחשב האישי של אחד העובדים ונגנבה התעודה, בעל התעודה דיווח לחברה, והנפיקה לו תעודה חדשה, אך נשאלת השאלה – כיצד תבטל החברה את האפשרות לשימוש בתעודה שנגנבה? יש צורך במנגנון תשתיתי שיאפשר ל-CA לבטל תעודות דיגיטליות ברגע שנגנבו, פגו או במקרים אחרים.

על כן, כחלק מהפתרון הכללי, נהגה מנגנון ה-CRL. מנגנון ה-CRL הוא חלק מתשתית ה-PKI, והוא מאפשר ביטול תעודה ע"י המנפיק בצורה כזו שברגע הביטול – או בזמן הכי קצר האפשרי, ידעו משתמשים ושרתים אחרים (או כל גורם המשתתף באמון) שלא לאשר תעודה זו.

בתעודה דיגיטלית של Certificate Authority המנפיק תעודות למשתמשים/שרתים קיים פרמטר נוסף אופציונאלי, הפרמטר הזה מכיל כתובת, URL, או כתובת שיתוף (UNC Path) לקובץ CRL. שימו לב שגם הפרמטר הזה חתום בתעודה אזי לא ניתן לשנותו מבלי לבטל את החתימה הדיגיטלית. קובץ ה-CRL הינו רשימה של מספרי תעודות שבוטלו ע"י מנפיק התעודה, להחלטתו. לווידוא שלמות ומהימנות הקובץ, גם קובץ זה חתום דיגיטלית ע"י ה-CA.



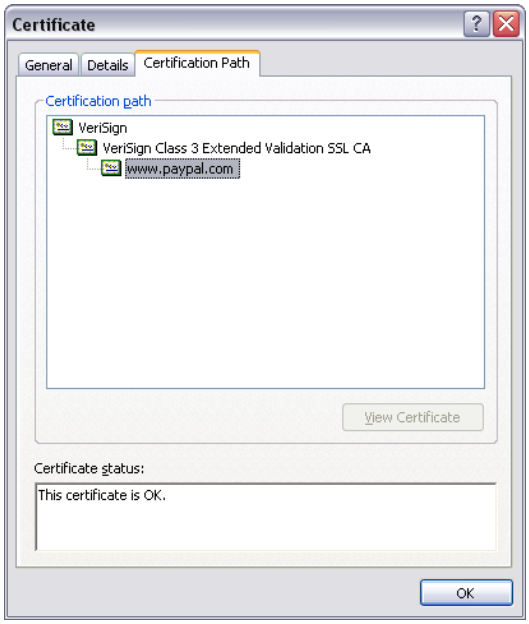
כתובת ל-CRL של VeriSign ICA מצוינת בתעודתו.

שרת שעליו מאוחסנים CRL-ים , ותעודות של שרת CA, נקרא שרת Publish.

בנוסף, בתעודה של ה-CA, ניתן לציין מרווח זמן מקסימלי לעדכון של CRL-ים. לדוגמא, שרת CA יכול לציין בתעודה שלו שעדכון CRL חייב לצאת כל 48 שעות. אם משתמש בודק תעודה של משתמש אחר החתומה ע"י ה-CRL הנ"ל, חייבת להיות למשתמש שמוודא את התעודה גישה לקבצי CRL-ים "טריים" שנוצרו ב-48 שעות האחרונות, אם לא, המשתמש **אינו יכול לוודא מהימנות של התעודה** ללא גישה לקובצי ה-CRL.

Certificate Path/Chain

בארגונים גדולים (וגם באינטרנט), מקובל לא לרכז את השליטה ב-CA אחד, מפאת חלוקת הרשאות, הפרדת רשויות, חלוקת עומס, אזורי מיקום שונים ויחידות ארגוניות שונות. אולם עדיין חייב להיות רשות אחת שמפעילה את שאר הרשויות. הרשות העליונה ביותר נקראת Root Certificate Authority, ויש לה הרשאה להנפיק תעודות ל-CA אחרים.



דוגמא לשרשרת תעודות עבור Paypal

בארגונים גדולים מאוד, מקובלת החלוקה ל-לא יותר מ-3 רמות של CAs:

- Root Certificate Authority – השרת הראשי, וקיים אחד כזה בכל ארגון, התעודה שלו חייבת להגיע באופן ידני למחשבים המשתתפים באמון.
- Intermediate Certificate Authority – שרת CA 'ביניים', משמש להורדת עומס מה-Root CA.
- Operational Certificate Authority – השרת האחרון בשרשרת שלרוב הוא שמנפיק תעודות למשתמשים, מחשבים או שרתים. לרוב, השרת הזה מחובר לרשת הארגונית או לאינטרנט.

להלן רשימה אפשרית לפרמטרים מנהלתיים שמאומתים באמצעות ה-PKI.

פרמטר	ROOT CA	ICA	OCA	תעודת משתמש
יכול להנפיק תעודות ל-CA אחרים	V	V	X	X
זמן חלוקת CRL-ים מקסימלי	שבועיים - חודש	שבוע	יומיים	אין
קישוריות לרשת	X	X	V (גם לא תמיד)	תלוי אם Client או Certificate Token
סיכון במידה שנפרץ	מוחלט, כל הרשת בסכנה	הסכנה הינה עד לחודש, לאחר מכן ימנע החוסר ב-CRL-ים אימות נוסף	הסכנה הינה עד לשבוע, לאחר מכן ימנע החוסר ב-CRL-ים אימות נוסף	תוקף יכול להתחזות למשתמש עד לטווח של יומיים
אבטחה	מאוישת גבוהה מאוד, 24/7	גבוהה	בינונית	ברמת המשתמש, או ברמת החומרה (ראו המשך)

למעשה, מעבר לזה שה-RCA וה-ICA אינם מחוברים לרשת, הם לרוב בכלל כבויים – כי פשוט אין להם סיבה להיות פועלים.

כרטיסים חכמים

כיצד לא נאפשר את שכפולו של המפתח הפרטי?

כאשר שרתי ה- CA מאובטחים עדיין לא פתרנו את נושא האבטחה הפיזית. הגנו על מספר מצומצם של שרתי CA באבטחה פיזית, אבל מה קורה אם מאות ואלפי המשתמשים שקיימים בארגון? אי אפשר לצרף לכל משתמש 2 שומרי ראש, נכון?

אין לנו שליטה מספקת על האבטחה הפיזית כאשר מדובר במשתמשים , אולם פותחה שיטה טכנולוגית בכדי להגן על המפתח הפרטי של המשתמשים גם במקרה ונגנב. שיטה זו הינה שיטת 'כרטיס חכם'.

התקן נייד 'חכם'

כדי להגן פיזית על המפתח הפרטי חייבים סוג של התקן פיזי שניתן לחברו למחשב ובעזרתו לעשות שימוש במפתח הפרטי. בנוסף, המכונה שאנחנו מחברים אליה את ההתקן יכולה לגנוב לנו את המפתח , מה שמחייב אותנו לעשות את כל פעולות החישוב עם המפתח הפרטי אך ורק **על ההתקן**, כך שרק את תוצאת החישוב נשלח למחשב. המפתח הפרטי לעולם לא ייצא מגבולות ההתקן.

אז הלכו וייצרו התקן כזה, לקחו מעבד קטן, שמו מאחוריו זיכרון (מאחוריו כלומר שהוא מגן על הזיכרון ואי אפשר לקרוא את הזיכרון ה מוגן ישירות מההתקן) ובזכרון איכסנו את המפתח הפרטי . את כל הפעולות שדורשות מפתח פרטי (בין אם מדובר בהחתמת מידע ובין אם מדובר בפענוח מידע שהוצפן בעזרת המפתח הציבורי) יעשו הכל על המעבד של ההתקן . הוא יכול להגיע בצורה של כרטיס אשראי שנקרא באמצעות קורא, או בצורה של USB Token שבכביכול הוא גם הקורא וגם הכרטיס יחד.

Factors of Authentication

אם ההתקן נגנב אזי מי שגונב את ה-Token לא יכול להוציא את המפתח הפרטי, אבל יש לו את הכרטיס, והוא עדיין יכול להשתמש בו כאוות נפשו.

כדי לפתור את הבעיה הזו הגנו על השימוש במפתח הפרטי באימות באמצעות סיסמה. לפני ביצוע פעולה **על הכרטיס שמשתמשת במפתח הפרטי** יש צורך להזדהות מולו עם סיסמה.

כלומר, כדי שתוקף יוכל להתחזות למישהו אחר, עליו **לגנוב** את ה-Token ולדאוג להשיג את הסיסמה בדרך כלשהיא. על כן שיטת הזדהות המשתמשת ב-Token ובסיסמה, קרויה 2-Factor Authentication, מכיוון שההזדהות משתמשת בשני שיטות:

1. What You Have (Token \ SmartCard)
2. What you know (Pin Code \ Password)

קיים פקטור הזדהות נוסף אחר והוא

3. What you are.

כלומר, מה אתה – הכוונה הוא לזיהוי ביומטרי, בין אם באמצעות רשתית, דגימת DNA, טביעת אצבע או שבלונת רגל (☺). הזדהות עם 'PIN ביומטרי' מתבצעת בצורה דומה ל-Pin רגיל, המשתמש מזין את טביעת האצבע שלו לקורא, הקורא מפענח את נתוני טביעת האצבע וגוזר ממנה נתונים מספריים. את הנתונים הללו הוא שולח לכרטיס החכם עצמו, אם הכרטיס החכם מאשר, הוא נותן למשתמש להשתמש בפונקציונליות שהוא מממש (חתימה דיגיטלית, פענוח נתונים שהוצפנו באמצעות המפתח הציבורי המקושר לתעודת הכרטיס וכד').

כמה מילים לסיום

נכון לזמן כתיבת מאמר זה הכנסת מנסה להעביר חוק שיחייב את כולנו להכניס את טביעת האצבע שלנו למאגר מרוכז. כדי לנסות להמחיש כמה זה לא הגיוני בעליל תחשבו על זה, שהסיסמאות של כוווולכם ל-Gmail היו מאוכסנות אצל משרד הפנים, נשמע הגיוני? לא ממש, בכלל, אז מה היה קורה אם היה לא מדובר בסיסמה שלכם לג'ימייל, אלא בזהות שלכם, בכסף שלכם, ברישיון שלכם, והדרכון שלכם?

מאגרים זה דבר רגיש שעדיף שבכלל לא יהיו קיימים. כולנו יודעים שמרשם האוכלוסין החל משנת 2000 או קצת אחרי דלף כל שנה לאינטרנט וניתן להורדה היום מאימיול. אני חושש שאם אכן יוקם המאגר, לא רחוק היום שגם הוא יגיע לידים הלא נכונות.

היום ניתן בקלות ליצור שבלונת סיליקון בכל צורה של טביעת אצבע רצויה, יותר מזה, באחד מהפרקים ב MythBusters הראו איך בקלות עוקפים את מנגנון ההגנה הביומטרי (**ביומטרי בלבד**) יחסית מתוחכם שקיים בשוק - http://www.metacafe.com/watch/252534/myth_busters_finger_print_lock

כמו שהראתי במאמר זה, קיימים פתרונות טכנים רבים, בין עם שימוש ובין בלי שימוש בטביעת האצבע, לפתרון תעודות זהות דיגיטליות (שנראה שלשם פני הממשל מכוונות). פתרונות אלו קשיחים ברמה הפיזית (של הכרטיס החכם) ומאוד קשה לפצחן. אף אחד מהפתרונות לתעודות זהות דיגיטליות לא מצריך את קיומו של מאגר נתונים ראשי שכזה.

קריאה נוספת

- http://en.wikipedia.org/wiki/One_time_password - One Time Password הינו פתרון אחר להזדהות שהיא מעט יותר חזקה מסיסמה רגילה.
- http://blogs.zdnet.com/security/?p=2339 - השתמשו ב PlayStation3 300, והפיקו סט מפתחות ותעודה דיגיטלית מפוברקת של CA ע"י שימוש ב MD5 Collision Attack. – הם כעת יכולים לחתום בצורה מפוברקת על כל אתר שיחפצו.
- http://en.wikipedia.org/wiki/Transport_Layer_Security - SSL\TLS הפרוטוקול מספר 1 בשימוש באינטרנט להצפנה של ערוץ קריא בצורה מאובטחת.
- http://docs.sun.com/source/816-6156-10/contents.htm - עוד על SSL\TLS.
- http://en.wikipedia.org/wiki/PKCS - PKCS הינו סט של סטנדרטים לייצוג של תעודות מפתחות תהליכים, בקשות להנפקות, Software API, ועוד לשימוש בעולם הקריפטוגרפיה.