① 22.10.24
הקלטה

הפרויקט הקשור ואלגוריתמים. אי שם נמצא אחד מהם יהיה
לו קצת קשה.

מה זה קלוס אז? במובן הקלאסי של המילה, פשוט אין אפשרות להוסיף
ולגלה האלגוריתם של הקלוס הזה יותר מהשיר. אנחנו נסקור איזה שם
כלים וטכנולוגיות שיקלו שיהיה לכם אפשרות איזה.
במידה ואתם לא קלוס מאוכלס למה? כמו שעכשיו השינוי הטכנולוגי
הגדול שעוד פתוח ופתוח. אבל מה בעצם קלוס עושה? מתקף אתופולוגיות
שעוזר שניתן לקחת זיכרון ולקחת אותם, נפתח תוכנה שמין אנשים
שלא מכירים את התוכנה בכלל.

תכנים, אפשרות וכו' - לקלוס אין אחריות ואין מופיע תכנים. יהיו 6 תכנים
ולכיף להגיש את כולם. התכנים יהיו בשיעור תיאורטיים (לא נפתח קוד) ואם
נלכיר איזה מהתאוריות זה המתן. במתן כי בל שאלות. אין בעיה
גם שאלה קצרה מנושאו אסוים. גם שאלה יש כי- על תשובות יתקן
בכוונה. זה לא מבחן של שינון בעיף. המבחן מיפין תכונות. המבחן לא קל
והציונים לא מיפיאים (גם הלפאנות שכרוש...)

אין אתר רשמי של הקלוס והמרצה לא אפיל אתר. סביר שהחומר של
הקלוס בפלואנו יופיע ב- notes-heaven.

אתוס - האמנות איך?

מאיפה האמנות מגיעה כך מוכרת?

- מוכרות הנושאים שאנו אוהבים - אחלה אישי
- ציפורן
- מרשים סלוני
- תפוחים
- שיקור רגיו / שלים

- GPS / ניווט
- שעון מדוייק
- טלויזיה
- אינטרנט
- מכוניות

כל הדברים האלה, בדרגה כזו או אחרת, עובדים בדרך ממוחשבת ומשתמשים באלגוריתמים מתוחכמים.

מה זה בעצם אומר עבור משפחת ישראלי? - כל פרט שתיארנו כאן מאפשר נוחות ואפשרות לחיות את חייו בצורה שנראתה בלתי אפשרית לפני עשרות שנים בודדות. אבל, כמו בכל אחד מהתחומים האלה, מהנדסים של מחשבים צריכים בכל זאת להתמקד בטכנולוגיה. בדרך החשיבה והפיתוח והפרוטוקולים משתנים מתחום לתחום. הדוגמה כי אדריכל שאני אפשר לטפל בכל הדברים בדרך קלה ונוחה. האדריכלים יצטרכו עוד זמן מסוים לשתף פעולה עם התחומים האלה בדרך אחד או השני - שהרי אין ספק שהם ומחשב יפעלו ארוכים בפרויקט וכו'. אך יש מאפיינים שדרוש להדגיש. אם יש n מרכיבים יש $O(2^n)$ מאפיינים שדרוש להדגיש. והרי הדברים לא נמדד לפי מספר הפרמטרים שדרוש להדגיש. משכך, שאין תמיד נקודת מבט נכונה אחת מכל יתר הנקודות - ומשכך, שאין תמיד נקודת מבט נכונה אחת מכל יתר הנקודות.

נשאלת השאלה למה רואים סטנדרטיזציה של הדברים? בדרך שהיינו רגילים אליה. אפשר לקבוע סטנדרט לשאר שמתפתח כך מהר... אם הדבר היה מוטמע (האנשים היו טובים מספיק שאפשר היה תקנות שמקובלות בכל העולם וכו') יכול היה להיות אפשרי מיד. הרי אפשר מיד לא מחשב שנופל וכו' יכול היה להיות אפשרי מיד. הרי אפשר לא מחשב שנופל מהכיס מדרך - נהיה קשה כפל ואי אפשר למנוע. אפשר הקלות היא לא מחשב שנופל מהכיס מדרך - נהיה קשה כפל ואי אפשר למנוע. אפשר הקלות היא לא מחשב שנופל מהכיס מדרך - נהיה קשה כפל ואי אפשר למנוע. אפשר הקלות היא

איך ניתן אפשרות מידע? בדרך הפשוטה ביותר מספרים רעים מקבלים (שפעל לא יזכה מהמוקד, שלא יפרסו וירחיקו את המשתמש, שהיטק נפץ). אפשר גם כמובן פשוט לא לשים אף אחד ולא לדאוג כי אם המשתמש מתקשר מתקשה אותנו שלנו? אם המשתמש מתקשר אין לישות למידע שלנו. אומרים וכואבה לו הכי בתרון. בעל אופן לא מדובר כאן רק על מספרים יקרים שעושים עושים).

כדי לאתחל קצת את הדיבור - הפעולות נקראות מספרים רעים. נשים ♡ שמדובר בדיבות לביות סמירות - לאון שקיים יותר כסף באפשרות של משתמש שיוכה פחות.

מאפיינים / מדדי אפשרות

- אופטי מידע (היה קהל? ומה (שלם)
- איכות מידע (היה קהל סוג, ומי שאו אותו שים לישה אליו)
- השפעת מידע (יש קהל שמוכים עליו אפשר שיש שיבוש)
- אי לישה למידע (השפעת של הפרסה (פ)
- אינטרס לישה (השמישות מוכים אקטיביות מונע לישה למידע)
- הפצת מידע לאן תביא (Shares)
- התנצלות, אינטרס לבוא
- הונאה

מונחים

- Eavesdropping - האזנה. אם כך אשמו סודי גו בדיוק לא מדיה. אם כך משמו סודי גו בדיה ומדבר הפסיק פתרה את הבעיה. שלך משאנין בדיוק קרבנות Eve.
- Snooping - בדיקה. כך פירוש להאזנה רק שבהקשר הזה לאחשב ואולי האזנה היא שלקוי תקשורת. את בעיה ההדדה פותרים בדרך שונה כי הפונה כך לא פרקטי. ואין בדיוק מפורסם בהרשאות לישה וכו'.
- Tampering - התעסקות. הכוונה היא שנכנסים למצ' ומשנים את פרטים - סיסמאות, הרשאות. אפשר בדיוק הכוונה לאו עשינו סיס"א אחרת.

מתקיפים את המץ.

בקרס נפגר אדם שהוא מהנדס תוכנה. הוא מת מהתקפת וירוס. כשאנשים שמעו על מותו, תשעה מהם אמרו: "אני חושב שמתי בגלל וירוס שתקף את המחשב שלי". העשירי אמר: "אני חושב שמתי בגלל וירוס שתקף את המחשב שלי". העשירי אמר: "אני חושב שמתי בגלל וירוס שתקף את המחשב שלי".

- Code injection - הזרקת קוד זדוני. וירוסים, רוטקיט, תולעים, סוסים טרויאניים וכו'.

- Jamming - חסימה. חסימת גישה או שיבוש תקשורת ע"י הצפה למשל. למשל אפשר לגשת אוטו של הנהגים ולשבש את המכונית.

- Spoofing - התחזות. הכוונה היא להשתמש בכתובת IP/ של מישהו אחר כדי להתחזות. כאן יש גוף מתוחכם של פיקוח / אימות נתון.

- Buffer overflows - הצפת חוצץ. הכוונה היא לשתול בכתובת

שני סוגים של מץ:

- טפילית - מץ שאין לו שליטה על הזרימה - נסחף איתה (אולי התנועה היחסית אשמה, אפשר כוראה) או שהוא נישא בדרך 'זועמ'), כלומר אדמה, שלכן, נפוצה של פרסן וצוב. המץ הזה לא מסוגל ל' מישהו שמשפיע אותנו אלא ע' אשתו שיש לו סכוי מופץ, לכן בפרט ואז הזרם שמובדס ביטויים.

- יזומה - התנהגות, פרטית אישית מיוחד, שלנו מיוחד. אלה דברים שיוצר קשה ע כמה את הסיכוי שהם, את הנכס הפרטי ומדם ביטויים עם אזורים. ואז דרך ע שהוא ה לכה הנכס (או בהברה כסף) מתנגנן ההתאה הם מיוחדים. יש שם זה קצת אוטוס של סיבוי. אם יש לי, משהו שמסר זוכר רספי / פרוטי, כלומר שניהן סיכון. ואולי דברים בפל. לכן הם נתונים מסיכון יחד. ה לכה הנכס נישאר בהתאם לסוגים המזוכרים (מיטונית, רפואית וכו').

בקרס נפגר אדם שהוא מהנדס תוכנה. הוא מת מהתקפת וירוס. כשאנשים שמעו על מותו, תשעה מהם אמרו: "אני חושב שמתי בגלל וירוס שתקף את המחשב שלי". העשירי אמר: "אני חושב שמתי בגלל וירוס שתקף את המחשב שלי".

סוגי פרדיגמה

1) תוקפנים אקדמאים - אינטרס לחקור איך לפרוץ את הגבול ולהרוס את הפרדיגמה הם עובדים בדרך תוקפנית, שהם פרסומים (מדעי) ואינטרס הוא אקדמאי בעיקר.

2) נלחמי אינטרס - עובדים בעבודה תבנית אומנם בדיוק אין להם פרסומים, הם שופרים את מי, הם אומנים של הארגון במטרה לשפר הפרדיגמה.

3) הלקוחים של המוסדות - קבוצות או בודדים שפועלים אפוני מצדיקים ומי פרסומים
- - - -

היום נמשיך את הממשל. נדבר קודם כל על <u>הקפה אוטונומי</u>:
כל מי רשמה של נושאים שמעניינים אתכם להעמיק את המועדות שלנו
לבקר אוטונומי את אופן פעם מסוגים לאוטונומי מיקום, סוגי
של ארכיטקטורה של המבנים, מלי פרטים קטנים.

<u>משמעת תכנון</u> - לפי מה טובה יש בכלל משמעת שמתאר אותה. משקל
הכמה כאן בתכנון ותיעוד. יש באוטונומי מיקום נדבר שיהיה לנו
איך שמו תיאור של מה שמעניין עושה וכן הלאה. מה הדברים
שדיברים להופיע במשמעת?

- מטרות - הרשאות - ערוץ
- איומים - עדיפות - התאוששות
- מדיניות - ישויות - תקלות

הסברים:

- מדיניות - באלגוריתם שלנו המדיניות היא שקופה. למשל מסוג מסוכנות הם
הדברים הם מפעילים אחרים בשמו הם סוגיים. צריך להיזהר מהשימוש
תקין במדיניות ומהו שימוש לא תקין. אפשר צריך לשמור את עצמו מהי
המדיניות שמאפשר ליצור מסוגי המיקום שלנו. מה לא שמור או לבן.
מי שמדיניות יכולה להשפיע והמשתמש ויכולה בקלות ניתוב פלא מבוא שלים.
- איומים - צריך להיזהר מהם האיומים שמאפשרים ותן להם - הסכנה,
סריקה, יציבה, השתתפה, אי נישול ... צריך לשים לב שיכולות להיות
הרבה תת מדרגות תפץ אותה מהן איומים שונים ואוטונומי שונות. למשל
מיתולוגים לא יהיה מאוד מוטרד אם יפרצו לו למאגרי תיאורים של אתרות
מטמונים שיפרצו לו לתיקים אישיים של תלמידים.

- אילוצים - למשל סופיות - אומר הן תלוי מהמיקום יכול להיות סופי. אלה
הכוונה שיש מגבלות שמגדירים מראש ואי אפשר לשנות אותם.
- הרשאות - מי רשאי לעשות מה. למשל מי רשאי לקרוא את המידע, מי
רשאי לשנות את המידע, מי רשאי לבצע פעולות מסוימות על המידע.
- ישויות - הדברים שאנחנו מדברים עליהם, הדברים שאנחנו מתייחסים
אליהם. למשל מיקום, מסוגי מיקום, אוטונומי מיקום, מסוגי אוטונומי
מיקום, מסוגי מיקום, מסוגי אוטונומי מיקום, מסוגי מיקום, מסוגי
- עדיפות - מי יותר חשוב ממי. למשל אם יש שני מסוגי מיקום שמנסים
לבצע פעולה על אותו מיקום, מי יותר חשוב ממי? מי יותר חשוב ממי?
- ערוץ - מה הערוץ שדרכו מתקשרים עם המשתמש. למשל מסוגי מיקום
- התאוששות - מה קורה אם משהו נכשל. למשל מסוגי מיקום
- תקלות - מה קורה אם משהו נכשל. למשל מסוגי מיקום

אמיתי הוא הציון. ו עיקר שאיפת שניהם בדרך עושים אימונים לתכלית ולא לקבל.

- צ'ימות מקור מידע - שונה מצ'ימות מידע. צ'ימות-מידע זה לא מעניין אותנו אמיתי הוא הציון. ו עיקר שאיפת שניהם בדרך עושים אימונים לתכלית ולא לקבל.

- צ'ימות מידע- זה שונה מציונו, כי זה שאמרתי שמעני דינה לה לגו בהכרת אומר שלה שניהם.

- צ'ימות מידע- שזה, המידע צריך להיות אמין כמו מהקרה של ועבורה

- צ'הני ישות- מצ'מה זה תאוש, או לפי שם, או תפיסה אלבש

- סופיות של מידע- שזה, זה לא חיים להיות שניהם לפני המידע.

מטרות

סופיות של מידע- שזה, זה לא חיים להיות שניהם לפני המידע.

- צ'ימות מידע- שזה, המידע צריך להיות אמין כמו מהקרה של ועבורה אומר שלה שניהם.

- צ'ימות מקור מידע - שונה מצ'ימות מידע. צ'ימות-מידע זה לא מעניין אותנו אמיתי הוא הציון. ו עיקר שאיפת שניהם בדרך עושים אימונים לתכלית ולא לקבל.

- הרשאות - פעולה מותרות. למשל אפשר לקרוא אבל לא לכתוב מותר לעשות ואת.

- תקלות - עם כל האמצעים שלנו לא נוכל למנוע מצ' מושלמת - קודם כי הדעה לא מושלם ושנית, גם אל שכל שתקרה ע שעות המתיר ישאל פעולות. אנו צריכים להיפיד מהן התקלות שיכולות לקרות- האיומים הפוטנ' מכוסים (מסיבה כי אחרת).

- התאוששות - מנגנונים נוספים עם פוגע בדברים שלא כיסינו - בכל שפוגע מתוך למעל. למשל, אם מניין עם אבסה את עדאו ניתק כלידות אדמה ואז הם התאוששו את כיוטות יכול לעשות בכה. אנ בקצה התאוששות צ דבק שתבוד ע פעולות מתקלה.

למה הדברים האלה חשובים? כשמנים ארכון חשוב שפתיה איכו תמונת עולם כלשי. כה אם חשוב שהבינים ע קלקולות מה בפיון המצ' עושה ואיכו אומטה היכו מספקת ואיכו אומטה היכ לעו מספקת. אחרת הקלקולות ונלים לתפוץ על אנק שוש וכה תפע.

עכו גם ארכון יש מסמך כזה על המידע אנו נפאי שיחשבו בדרכה מסודרת.

- אי-התכחשות - נניח קיימנו תוכנה מתוחכמת וגילתה כי האיש חולה בו. אנחנו רוצים לגזור מניתון שאם התוכנה מחייבת לגרוק בפנינו גלגל הפנינואל שלושי - למשל בית משפט או בנק.
- צמינות מידע
- צמינות מדוכרת
- פרטיות מידע - כאשר מסוגיות. למשל, תביעת האלבץ היא לא סופית כופף היטו פרטיות. או למשל, אם אנחנו קונים מאכלות יקרים כמה אנשים אפף אנחנו לא רוצים שכל העולם יידע
- הסר מידע - כאן בדבר שיש סוגיות שונים של סופיות המידע. וגם מהתאם לסוגי המידע וכנל להגיע מסוים של רשמי או שאפשר לה להפין אותו רק בתנכרת אורך מפתח מסוים.
- אנונימיות - זה קצת מתקשר לפרטיות. אנחנו רוצים שתהיה לנו וכנל לעשות דברים מפני הפקודות. למשל, אולי מתישהו ההתכוונות יתבקשו דרך האינטרנט. אז נרצה שלה יהיה אנונימי.

אמנם יש קשר בין הדברים, הישויות וההרשאות. אפשר להגיד את זה בטבלה דו-מימדית. למשל מטריצה כזו:

ישויות \ דברים	A מחשב	B קובץ	C מסכה
admin	✓	✓	✓
אורח	—	קריאה	?
מנהל סטודנט	✓	כתיבה/קריאה	נדרשה

* הבעיה היא שלפעמים אין סדר של אי שלמה כי בדיוק יכול מדובר בשלמה אחת של סוגים שונים של דברים.

מוף פעמה היא בנק. הדברים הם למשל תשבונות או נכסים והישויות הם בעלי תשבונות. בעלי תשבונות יכול לבוא לבנק ולבקש לבצע פעולה על סוג של פעולה.

אף, הן הרבה פעמים משלבות גם מניעה וגם גילוי - למשל firewall.

עניין הגילוי שמהווה מניעון ההתקפה.

אנשים מסוימים או קבוצות עצמאיות. אפקט מניעה לא תמיד עובדת. וגון נכנס אחר, משפט מעניין - יותר מ- 90% מהפריצות בארגונים נגרמות ע"י שיתוף פעולה של אדם מבפנים.

- התאוששות - כמה מהרנו - גיבוי, יתירות, פיטורים

הדברים האלה נעשים מהרבה פחות מורכבים אפילו הם לגו פיטורי עובדים. קחלף שבו רשום מי נכנס למערכת, איתרן, files log, משפט למשל מצלמה, אזעקה, ניטור (monitoring) - למשל

- מניעוני גילוי לגו מונעים מהשפעות הקודות בדיוק אפילו הם מצמצמים שקרה

גורמים, ההפנה, firewall, ההרשאות, שומר

- מניעה זה לרואה ה"בעל הבית" ביותר אפילו הכי אתרנו שזה בעלות, סופשפט ב- 100%. פרוצאות למתקנני מניעה: מנעול, אוטו-

① מניעה ② גילוי ③ התאוששות

שיטות בקרת סיכונים

?	יש בעיות שצריכות בעלי לגו עושים כלום.
בלי בעלי	בעיות - סומכים על זה שיש שמישי שמונע רוכשים את הבעיות.
אנשים מסוימין או קבוצות עצמאיות	תוכן גילוי - למשל את כרטיס שנותנים אלקטרוניים - כפל מצוין, בלי וכו'
אלקטרוניות, אפשר לשתול זה משהו. מוקפסים. זה סתם לנו משהו ניתן לעשות קפיצ זול סוג של גילוי. ניתן לעשות קפיצ משהו אולי ניתן לעשות קפיצ יום. בלקץ ואתנו נדרים מאנו. יש לפ עשויים, אולי זה משהו שקורה	אפקן מיקץ - למשל פיסן הגוטס - גילוי: זה לא משהו שאותנו מעניין
	התאוששות

משנה לשנה.

הבעיה עם מקצוע אחד שהמורה אומרת זה שהוא יכול להכתיב את הכללים ואת הקשה ומאבק את החיים של הספרן.

אחר שבחורה באמצעה אפשר להגיש כאן קליפת בצל. אלין אפשר לחתוך ירוק. אל תשים אותו זה בלבד אחרת, ואם מתקיפים אותו הוא יוכל להתקיף תגובה ומהבה יותר להתקפה - עדיף נניח לספרן. זו מלוה כי הספרן מבלב הוא לא יעשה את הכללים של הספרן אלא הכללים של הדין הוא עצמו.

- המטרה היא כפולה - שכנוע צמן לספרן איך אפשר לעזור את הדעין - ולתפוס לקוחות אליהן, לעזור הספרן מבלב מאוד לשם. זה סוג של אתגר שהוא שנראה כאן את אחיותיו ויותר קל במידה שמ"מ לקוחות אליהן, לעזור הספרן מבלב מאוד לשם. זה סוג של אתגר שהוא שנראה כאן את אחיותיו ויותר קל במידה שמ"מ

- יתירות - סוג של יחסי שבו יש יותר מדי מחכות שאין קבלי יומי.

- הדפנה - זוהי המשתנה מחיים את שם נקבה לפני או הדפנה אותית, אפשר לעשות לחתשב זומע עם תהיה לושה עקלפף.

- הרשאות גישה - כי הכתבה שם היחס פיסית - כי היחס אלקטרונית. הדפנה ניש לחלם היא לא שווה כלם.

פיסית לא מי שרוצה את כולם יכולים לעשות בהם שימוש. אם אפשר - היחס פיסית - לעשן קבלים, אפשרות, סיכומות - כי אם היו ניתנים - אחרת הסברים:

ט) היחס משפטית.			ה) ניטור
י) התקפה			ו) קריסטלוגרפיה - אולי בהיר
יא) הטעיה			ז) הרשאות גישה
יב) יתירות			ח) היחס פיסית
יג) אחר:			ט) מוזלות משפטית

שבועות אמטחה

- least privilege - לתת את הפעולה עם ההרשאה הנמוכה ביותר שמאפשר לנו לעשות אותה. לא נותן למשתמש גישה להרשאות הנמוכות ביותר שדרוש יתן לו גישה אליו. למה? אם הקוד נפרץ ויש לו הרשאות גבוהות יותר הוא יכול לעשות נזק. אם נכנסים ב superuser ומוחקים את כל delete all זה לא ישאיר לנו מחשב ואולי אפילו יכולים את כל הקבצים של המשתמש. זה מאוד משתנה אפשר לתת לו את הקבצים שלהם.
- trusted components - בכל מיני ספריות שמורים עליהם. כדאי לבדוק אותם, לפעמים מישהו שהם מאוד מפורסם ואז לא כדאי לפתוח אותם כי הוא עושה את הדברים בהלכה, אלא רק מהי פעם. למשל, מתחזה הקליד שני שניים היה אינה סיפור עם מקלדות שהיה מהן כיוב שהיה מקיש את הקוד סיסמאות ושלח אותן לאנשהו. אז כדאי לקנות מוצרים מחברות מוכרות שיש להן מוניטין שהוא.
- Simple Design - אחד הדברים שיכים לעולם הזה שמע פשוטות יותר הן טומות יותר. מה הפשלה שאיכה יכול להיות בקוד סבוכ, היא כנראה פחות טובה ממע' הפשלה מובנית. לא נשים לב לתוכה. כמו שמע מסובכת יותר קשה יותר לתקן אותה.
- Paranoia - תמיד יבדקו אחרינו. אם יש משהו שלא מאמין שלנו. את אנשי אבטחת מידע תמיד צריכים לחשוב מפרנויה מוי"מ, לחדור סיסמאות כשהם מבצעים למן, שבודק קונפיגורציה של ה-firewall וכו'.
- No perfection - זה מבוי שלא נוכל להגיע לשלמות. תמיד יהיו פרצות, תמיד משהו יתקלקל, תמיד יהיה מישהו שישתף כמה כסף מוכנים להשקיע על זה.

- הוספה/מחיקה מתיקיות מועדפים
- הגדרה של הורדה נקי עם מיקומים מסוימים
- סקירת מיקומי אבטחה
- הגדרת תזמורי נקיים (אופציה סיכויים מנקים
- הגדרת סיכויים ונקיים (למשל **to** ספציפית כמה פורטים יש למבקר)
- ניהול אימות/גישות
- הגדרת נכסים (תוכנה, תוכנה, מידע) - בדיקת הקמאיד שווי?

ניהול סיכונים - מיפוי איומים

- ניתוח הסתברות של 0-יום ו-1-יום כדי שלא ניתן יהיה לשארר. קלפט **T**. זכר ב שמנו לאחוק מאש בדין כאה בלאים ערוכה סלוחי מאומלים אלוהיים אם ס שכתוב בפוסק הכומאוס ס או שהיה אלקטרו-מיגנטי אמק אם אמא אפשר לעשות דברים.
- פוסקים - רשאורקים קבלת מפוסק הכולל מאש נעתק. אפשר לבדוק אלקטרו-מיגנטי אמק אם אמא אפשר לעשות דברים.
- מסכים רחוק עם כוכה אלקטרונים פוגעים קרינה אחוזה אפשר שלקוח מי שתלה את המיגן. מסכי LCD ו/או פוגעים קרינה שלוטים **Tempest** מיגן שאם אחישים שאתלקים אותם.
- שלוטים את אלמה בחיצ שפותחים אותם, אלו יש רצוה ג'יפים יותר ויותר מטוחים. למשל יש ג'יפים שיש להם מיגונים אי'כה אלחוריתם סדי, אל ג'יפ כזה לא יהיה טוב. את תכונת אי בדיקת מיקרוסקופ ורצוה שפוגות את הפרוכה שלהם. זכר אם יש ג'יפים שהתוכנה שהתוכנה. פרוצים אוכלים יוקסם שקוף אחסון מפסם, פוסטע ה'אינפורמציה שהתוכנה. פרוצים אוכלים יוקסם שקוף אחסון מפסם, פוסטע
- רכיבים - מי מיני ג'יפים, לוחות וכו - אפשר לפרק אלוהים מהלכים את סוג כמו תושב רצי הכולל אחוזה מפלי?
(יש בדיק פעמים כלה הכא להלפין את המיגן אכל כה הכולל פוסיק) אב בדיק להיות אל הכפלים. כאומן רק שאפשר טובים יותר הם אלוים יותר. ש ויש מצ, שיכולות לפרוס את הבדיקה הכאלו תלכה לפאינפורמציה. אפשר לחוש. התוכנים וכלים לחוש את הבדיקה שנוכלת מה לכם
- תוכנים שפכסם צורם מיגן אוכלוסי, יש להם בדיקה שאופיינית ואותה

כשאנחנו שמים הרבה מאמץ בלימוד – אנחנו רוצים שהציון שלנו יהיה גבוה. אומנם כל שאלה מציגה אתגר ייחודי משלה, אך ישנם כללים שאנחנו יכולים ליישם בכל שאלה.

- לחץ
- מנוע שרת
- וירוסים
- ג'ימיילים
- ג שמתרע מידע
- אומדן מידע

} עומדים הם שניות שנדרש בשמעות ומשמש אותו בקריפטוגרפיה, אומץ וכן יהיות שיש איזה כלי שפיתר בעזר אלה הקרובים.

מהי הקריפטוגרפיה עבר מטפלת?

- טרנסאקציה מטמה
- הקלאר שיתה
- תיאום מפתחות
- הזדהות בפלן ונוסף לפחות כ- 30 שנה.

מסוים היה לך ונון עסיק מה התחייבות. לה לאן היה מוכר ב עולם

- אי-התכחשות - אפשרות ליצור מנגנון שמי מישהו מתחיב על משהו
- אימות מקור } ב- 2000 שנים אחרונות
- אימות מידע }

ג שמ מעבר קריפטוגרפיה.

- סתירות - הלפנה של מידע - זה השמוש הכי נפוץ וזה הדבר היחיד שבעבר

שימושים בקריפטוגרפיה

רוצה מה שמשתמשים מו היום בשיטה הקלאסית.

מאמץ אלפי שנים. יש מספר פרובלמות בזה ואולי בהן בהמשך, אבל

והכי רואו היום. הטכנולוגיה השתנתה אמנם ולהבנות נשארו בי עדיים

מסתבר שבהצפנה אמוץ מוכנים (חוץ מכאלה תלויים) הקריפטוגרפיה הקלאסית

סתר ← כתיב ← קריפטוגרפיה קלאסית.

קריפטוגרפיה זה תחום אמוץ עתיק (אלפי שנים). השמוש נפוץ עד

אתוקדמה אמוץ ביום והסתיר את ה הפעיות.

בשאלות התחום ותאקד בנושא אחד בקריפטוגרפיה. לה רק בי אחד

29.10.07
הלפנה

וכן לפעמים את כדי מקרי קצוות. וכאשר מגיעה, כי אם האלגוריתם הוא סוף כאן הסוף האלגוריתם. אחרי שאנו יודעים את האלגוריתם הוא אחד פשוט מאוד שיש לו מתו.

כל מפתח את שלושים מלוסם ה מיוחד. לכן מצורת אותה שונקציה כאמור

ATTACK FROM EAST → DWWDF...

הפונקציה שמשמשת להצפנה היא

$$E(M) = \underbrace{(M+3 \pmod{26})}_{\text{כלומר יתו שמורחים מאקום } M}$$

- E הפונקציה
- M message הודעה
- encrypt הצפנה

ומשים פשוט Shift לאחור:

$$A \to D$$
$$B \to E$$
$$C \to F$$
$$\vdots$$
$$Z \to C$$

צופן קיסר Caesar Cipher

וכן לכפלם וזה מתקבלים מהתשפ אותיות -

$$\begin{array}{c} \text{א} \leftrightarrow \text{ג} \\ \text{ב} \leftrightarrow \text{ד} \\ \vdots \\ \text{כ} \leftrightarrow \text{ו} \end{array}$$

אמם לא הייתה יכולה צד מאלכה כלשהי. אלא מסתכר שמלפני הכוונה מתוך - מספר ירמיהו - מופיעה המילה "ששך". ראוי איכו מאלכה מעמן שעוד לפני אלפי שנים היקריפטוגרפיה הייתה קיימת.

קריפטוגרפיה מודרנית

אפשר לחלק את מה שקורה מיוחד לא, בעידן הפ' המחשב, וכעת מתוכן תמונה משתנים מידע. הקוסה הנאמני / לא נאמני לגן יהיה שיש משתמשים ברכישה וכו'. פונקציה פסטאלוגיה לה רשאירשאו שיעוץ פעולה מיניהן - זה יהיו להלבישו עד אשתדלינו שקורה מינהן - אחוק יפרוה מינהן - זה יהיו להלבישו עד אשתדלינו שקורה מינהן - לעשן, אם יש שתי תכונות לא קשורות שמאחם יש תקשורת אלקטרונית תקשורת טשתי (אפילו מתי יפרוץ את התוכן) יהיה להסתיר משהו. מתפקידים מהטתכרת שיפתח קוסו המידע. הכי כך הדוגמה שיש

סטגנוגרפיה - קריפטוגרפיה מתעסקת מהסתרת תוכן. פסטאלוגרפיה

אופן תיווי. למעט באותיות E אחוז נפוץ - 12% ו-Q זניח - 0.29%.

– מהנה השפה: השפה הזו אקראית. בשפות יש התפלגות

הלה שני טכניות:

מרחב החיפוש הוא בגודל $26!$ שזה מספר די ענק. שמיתרון

אשר π_k היא התמורה ה-k על $\{A, B, \ldots, Z\}$. ראו

בשם "ץ התמורה" ראו האלגוריתם הזה $E_k(M) = \pi_k(M)$

אותנו רוצים שסדר החיפוש המחשב תהיה קשה מאוד.

התהליך שבו מנסים אפשרויות נקרא "חיפוש מחלה" exhostive search.

נשים ♡ שאפשר לזה שאתנו מפעים את הטקסט אותנו מפעים את אור

למשל כפי שיהיה אפשר לשפור את הקשר כן שאלה מעניינת!

על בנין "ית משהו ב-על משמעות הוא מאוד קטן כאשר טקסט ברוך

שמור. לזה שפעיר כן כשיש הרבה תלות פנימית - הסיכוי שאפשרת

שמקבלים משהו משמעות. אם מצפינים רבה רק אות אחת לזה פעמי

96 ניסיונות אותנו ורוצים לנחש את האפשרות. מנסים אפשרות שנים אך

והתוצאה לא בוחרים הקוד המוכן הוא שאחרי השפוש הקטן. אחרי עד הימר

שהיא פעם אפשרות אפשרית קל מאוד להתפלל אלגוריתם.

היתרון של השיטה הזאת שלא האלגוריתם סטטיסטי עליהם שנים.

נשמע פשוט אפשרות אי אפשר! האלגוריתם יפעל עד, יש תור לא בזה וביום

למעט - פעם אפשרות. האלגוריתם יפעל עד, יש תור לא בזה וביום

אומנם יש פה פרמטר k שלנו נקראו האפשרות.

מאלגוריתם תמיד, אומנם התוספת היא הסוף. האלגוריתם אחד יכולם

מ אשתה של פונקציות. אפשר לספר עליהם שמשתמשים

בפרט הקוד אופן: $E_k(M) = M + K \pmod{26}$

$D_k(C) = C - K \pmod{26}$

כי יש פרכים לפעמים אותו.

הקלטות הבין אחר שאין טעם לשמור מסוימות את הקלטות,

נוסף בפני האחר את אותנו מהקלה. זה אפר אפשר תשובה. הקלים

הסטטיסטיקות הללו תלויות מאוד בטקסט - שיקספיר, ספר מתמטיקה, תנ"ך למשל...

כפי שמור טקסט שמוצפן כבר אושים ניתוח התפלגות אותיות וסטטיסטיקות שעשוי לעזור לפיענוח כמה אפשרויות שנשאר שמצוין - מכאן הרמה פחות מ- 26.

יתר הסטטיסטיקות שאפשר להשתמש בהן הן תדירות זוגות אותיות (aa, th, oo, ng), תדירות מילים (or, the, a), סוגי הצפנים הללו נקראים מונו-אלפביתיים - פ"א אותה מוצפנת תמיד לאותו סימן. לפנים מונו-אלפביתיים תמיד שומרים ע"י הסתכלות במאגרי השפה.

מאחר ומספיק אותיות שיהיה הרבה יותר מ- 26! אפשרויות שהיוצר משתמש בה לפנים מונו-אלפביתי אמת עושים את זה בצורה "ג' האמורה" יש שוד תוסרון - תוסר פרקטיות. עצר כ"כ נוח להתגיא פרמוטציות. הרבה יותר נוח שאפשר יהיה להציב מספרי וכה בכדי לחזור לנו את הפרמוטציה. הפלט היא שלה משהו שקשה לנשות בדרך ישלה. צפה הוספת עוד יותר קשה אם יש פרמוטציות מסויימות שאותנו לא רוצים שהמשתמש מהן.

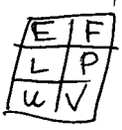
אחת מ הטכניקות שניכנס אל הן נקראות Substitution - Ciphers (צה"ל מכור למה).

Play fair מפרים את האב הטכנה בצפה וקלעים מלא קוד שקלעת את המילה של הטכנה. בניך מוה בי אכובה שאון מה נכה על אותיות רושמים אותה בטכנה וכל מתאים את השאר עם אותיות שעוד לא הופיעו לפי הסדר.

למשל, מילת קוד MONARCHY.

(בניך ים לוותר על אות אחת. מתכרה לה נ)

M	O	N	A	R
C	H	Y	B	D
E	F	G	I	K
L	P	Q	S	T
U	V	W	X	Z

אין מפתחות? למשל FUNNY.
אלתונית רוצים תכנה שלקחת את מחרוזות את כה ב- FUNZNY. ואז מתלמים בזוגות של אותיות.
F ו-N יודעות שמכן מטמלה: 
את מתלפים
את F ב-E ואת N ב- V.
באותו אופן ZN מתרגם ל- RW. אפשר עוד יותר "שמכן מנוון" את מה שעושים הם שפשוט יורדים כיום אותה שטכה ומקפלים כך. הפיענוח נעשה באותו אופן!
כאת דוגמה של קוד פולי-אלפביתי. ישפה תתכוף כלינו ואו אפשר להשתמש בה בניתות התפלגות של אותיות.
אפשר להתייחס לצירופים - למשל יש 26^2 צירופים שמותם אליהם הקוד הכה הוא מונוגלפי בלבד ואפשר לחתך אותו ל' התפלגות של צירופים. המקיה היא שישמתות המפלים סטטיסטיים מהתפלגות של צירופים. רבי שכאות הכפל בריק לקחת טקסטים הרבה יותר ארוכים. את כה לא שכה מאתי אפשרי לשכור את כה אפילו כה הרבה יותר קשה.
את השיטה הבאות אפשר להכחים למשהו עם שלוש אותיות או רביעיות ושם אפשר יהיה לשמור את כה אפילו פ פעמים שמחזורים את חוכן הקבלות כה מוסיף המון לאנלק ה טקסט שבריק כפי לשים לב להתפלגות סטטיסטיות.

Hill Cipher למשל ניקח פרמטר $m=3$ ונמנה מטריצה $m \times m$, שהקטנו לה מטריצת מפתח $K = \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix}$
נניח שהכה ופשה היכו $M = p_1 p_2 \dots p_m$ קטכפ בקוק $A=0, \dots, Z=25$
והכה פשה היא $E_K(M) = (K)\begin{pmatrix} P_{3i+1} \\ P_{3i+2} \\ P_{3i+3} \end{pmatrix} = \begin{pmatrix} C_{3i+1} \\ C_{3i+2} \\ C_{3i+3} \end{pmatrix}$ (הכפל מופנה 26)
מפוחאה כי אפשרים לפי שלושה אז כפל מטריצות ומפתח.
כה לגמי מונוגלפי בתי מכוכה הפשוטה. הציפור הוא לקחת m יפורים אחת כה לגמי שניקטי כי לריק לתשב את כה!

על ידמן) ובאיך אושים נעזרת הצפינה-
את זה בקצרה ומדגים את אורך מילת הקוד (ע"י תפוש מילים קודמת שמוצרת
כך שאורך מילת הקוד יהיו ההפרש הכי גדול (במקרה שלנו
ה אפרת ר).
מתברר שזה נכון אלפבית. (מכך שניו, אותנו לא יעניין את אורך
אפשר לומר שזה מונואלפבתי כי מילת הקוד המתאים ל- A הם פשוט
הראשונה מוצפנת ע"י O ושנייה ע"י N.
כך קוד פלו אלפבתי, כי למשל ל- ATTACK ה- T

מוצפנים ע"י O
מוצפנים ע"י N
מוצפנים ע"י A
מוצפנים ע"י R
מוצפנים ע"י C
מוצפנים ע"י H
מוצפנים ע"י T
מוצפנים ע"י ...
מוצפנים ע"י ...
מוצפנים ע"י ...

מהן - ומשן ההגדרות של הצפנה.
ולפי כלום יפה זה ענין שיש תמורות שאותן לא נוכל לשתמש
את התמורות (שפשוט משמעות לעניין פ "התמורה הרבעית")
- כאן שפשוט כתוב, יש כ-ם של תיבות, צריכה להיות בדיוק ע"י
- כלי ש-M יפעל יותר ההצלחה כלשהי מסוימת יותר (זו התמורה)
שפי הנתונות מאלגוריתם:
למה שפרים כמתכתב שלנו אותם!
אותם, את הצלגוריתם הלה כפני עצמם מתוארה - כי ע ההגדרה מובנים
בכמן סמוך.
אותנו כדי בהצפה יפעלה - אנו יכולת טרנספורמציות עסוק !מסע אפשרויות
אפשר לפקוד בחיפוש מאלה. אמת אם ההנקץ שלנו היה כאלף מסע
ההמורה לא לא אפתחות אפשרים. לפ שלה בפרק כמה מלינים
ה"אלגוריתם יפרק, בסוף היא ההמורה שמשמעשים בה, ע"מ המפתח ליכו
את הצופן! ע טכסט מסוים M יש לא תמורות אפשריות.
אפילו כך שטכסט ארוך יחד, יש יותר תמורות-ויותר קשה לשבור
ע אות מסוימה אותו מספר פעמים יש מטכסט גלוי ואם ם מוצפן.
שים לנו קלף מוצפן בהצלה? ע"י סטטיסטיקות של ההלתיות, שפרי
לא מתרפים את ההלתיות לגלא את ה תיאום. אומן אפשר ע להיות

טכסט מוצפן
טכסט גלוי

הצפני החלפה - Transposition Ciphers

פיעתו תשיעור הקודם ע קלף הצפנה. היום נעשיק בתיאור צפנים נוספים.

תרגיל 1 יוסף היום בשעות הלילה. הישה שלוש שעות מכותה?

תמורות על 5 הצמודות

אבל ג-אם קטנים מזה נניח עושות k כאות של תשוש
שלה קטן מ- $\frac{1}{2}$. כשיגיע ל- k כזה ש- $\frac{1}{2} < (\frac{1}{5})^k$ מה
זה $(\frac{1}{5})^2$ תמורות מקסם שבתישוש המלא בפקנו
$\frac{1}{5}$ תמורות מקסם שבתישוש המלא בפקנו
ולשכור את כה. אמם, אם נקפיא אנשים $\frac{1}{5}$ תמורות אחת אחרי מהן
כל אחרים - יש $\frac{1}{25}$ כאלה, וכל אשר נעשות תשוש מלא
אפשרים שהם ושום אחדים מקום עוברים בין פרמוטציות אל
נשים לב שאם בטבלה יש 25 תאים בצע את השאלות
שהשתמשנו בהם כי יש לנו מספיק זמן את התיים.
אביר את האלגוריתם שלנו, אפשר היה עכשיו לנסות את האפשרות
אם נעשה את זה כמה פעמים יש מספיק מזל. אבל האיב אולי
ולהפסיק אותו שם נראה אפשרות אחר. יש מספיק את השביבה.
אפשר אפשר שהתחכם. אפשר לקחת את הטקסט המוצפן
לנו מספר יפה, ואפשר לעשות תישוש מלא.
שקלאות. סוף כאן היה התאמה של הצמודות אבל 5 לה
כותבים את הטקסט בטבלה ואפשרת היא בסדר של הצמודות

3	5	2	1	4
A	T	T	A	C
K	F	R	O	H
E	A	S	T	A
T	D	A	W	N
A	N	D	T	H

$\longrightarrow$ AOTWT|TRSAD|AKETA ...

זוכר משהו דומה לזה שימוש בטבלה:

הקטר של הענין.

אבל אין הסתם לזה סוף בשפן הגלגל הזה הוא

ואבל כותבים אותו מספר הצמודות.

מספרים השתמש מאין ענין כזה שהיו כותבים מה זה האחר בשורות

Rotor Machines

כל מכונה מכאנית / אלקטרונית הבאין הוא כלי:
מעשה אנימה שימוש שימוש ע"ה הלקח מתחתית הגוף הרשייה.
משנה ה- 3c של המוח ה- שם צילה ירמני אותך ונאמץ אכוף

אז יש 26^3 אפשרויות שונות שלא רבה כ"כ אפשרויות
אבל כאן ה גיוונים יפה עץ ומפתח זאת המלא ההתחלתי של הפעלים.
מעלץ סימום הפעלים).
לתיקה ש 'aa' לא תניח תקאה זה שפי אותיות להיא -
לעולם ההתחלתי. זה קוד substitution פוי אלפבית (שפרי
אחרי ע לעולם הפעלים היא מתחלפת ואחרי 26^3 לחיצות היינו חוזרים
לעולם ההתחלתי ומפעיל נוזה מתחת לאות שפיא ההצפנה שלה.
השעמי היא מפעילים אחרי reflector - מיוצ ע ותוכר תלהה
איך היו מפעילים את המכונה? היו יוצרים על אות מאלפת. ההאות
שאפשר.

אפף זה מספר הפעלים קטן יותר סבבי שקטף את רוב מה
ירצ לדעת שהוא ההתחלה של תאורה ויתנו את אותה תוצאה,
עבור 26^3 תאורות אפשריות שקטף סבבי
עתה 26 תאורות סבבי הפעלים ירצ
להתחלה של ע אחד מ-26 (טוב, הדיוק אנו
ע אחד מהפעלים ירצ היה יכול משהו, אולם
מדוב)

מסיימת - פרוטוקול קבועה- ככה היו מיישרים את לה מאפן.
יש תאורים מין ההצפנים עם פרוטוקול
על ע אחרון יש 26 פינים ומתוכם העליון

שאלת השאלה - האם יש שיטת הצפנה שאי אפשר לשברה?

Perfect Cryptography

One-Time-Pad - ההודעה היא רצף של N ביטים

המפתח K הוא באורך ההודעה.

הטקסט המוצפן C הוא גם באורך N.

פעולת ההצפנה והפיענוח היא XOR

$$E_K(M) = M \oplus K$$

$$D_K(C) = C \oplus K$$

זאת הצפנה וזהה פיענוח לה, כי $A = K \oplus K \oplus A$.

מאותם שנים הראשים לאו דווקא שאפשר לשבור את הצפנה,
שהראשים יכולים לוודא שם האחרים שלהם מפוצצים.
לפעמים נקבע ביניהם שהם יבחרו ביחד. לא יהיה לאחרים הברירה
פעמיים נקבע ביניהם שהם יבחרו ביחד. לא יהיה לאחרים הברירה
לפעמ

שימוש

- אפיתת k מאורך הודעה M
- אפיתת k אחראי בפני מיכל
- אפיתת תג-פלאי
- אפיתת סודי.

הטענה היא שאיתות ההנחות הבאה אי אפשר לשמור את השיטה (גם בלי שלין טכנולוגיה, לה פשוט תכונה אתמטית).
למשל, נניח שאחנו יודעים לשלוח לפקיד הבנק מיכל אחד - 0 או 1.
ובנתנו מאריכים אפיתת באופן איזיד, מצפינים ושולחים. אין שום צורך שאחנו יודעים לשלוח לפקיד הבנק מיכל אחד - 0 או 1.
שאלה: פיתחנו את מיכל אחד. ברור שלי מעיה לפתח. מה אם שלחים על מיכל שמקלקלים משהו בקוד משמעות? האם אפשר לשלוח חלק מהאפיתות כי מבחוץ מתקבל משהו חסר משמעות? לא!
אפשר לחבור את כל 2^{10} האפיתות האפשריים ולשלוח את האפשרויות חסרות המשמעות. אבל באותה מידה אפשר לבפוק מהן כל האפשרויות שם טעות בתן משמעות באורך 10. ובכך כל האפיתות הסתברות הסתברות הזאת, אז אין לנו יכולת ביקורת מהקלקלות המשמעות היא כי המלית. אז הקריפטוגרפיה לא הסיבה טוב? אין שום מנוף. כל אחרי הה בפנים. הסתברות של כל הודעה היתה להיות בדיוק אותו אוקאי.

אימות הודעה: One-Time-MAC

message authentication code

מאימות לא מספיק תוכן. רק רוצים לוודא שההודעה נקייה ושלא אחד שלא יודע שהשקיד בבנק "לא שלח אחד בדיוק לה לו סופי, אפילו נניח שלקח את המקום של מי השליחות אותה. לעומת מספרים כפל בין השליחות לא שינה את הסכום המופיע או מספר

נראה תחילה כי:

p ראשוני, יהיו a,b כך ש- $0<M<p$ $0<a,b<p$ אחרת $K=(a,b)$ אפשרות סודית - בין A ל B ראשי a,b אקראיים.

(אם $M>p$ נחלק את ההודעה לחלקים ונעשה את הפעולות כמה שנאים p פעם אחת שנוכל).

הפונקציה שמשתמשים בה היא כלהלן:

$$f(M,K)=(a\cdot M+b) \bmod p = y$$

אותה משהו השתמש ויאומת דותה את ההודעה.

ההשערה היא: שאי אפשר יהיה להודעה פשוט או שאנו את המפתח.

טענה: אחרי שמתקבל כאה M,y הוא יכול רק לנחש את K מהסתברות $\frac{1}{p-1}$

(למעשה ההסתברות שיהיה לו מקרים הלה $\frac{1}{2^{(1-d)}}$).

הוכחה: עם $d>0$ אפשר לחלץ את השוואה a בקצב. p אחד מהם

טענה: אחרי שמתקבל כאה M,y הוא יכול רק לנחש את K מהסתברות $\frac{1}{p-1}$

הוכחה: אם מתקבל כאה M_1,y_1 ו- M_2,y_2 עם אותו K (שאי אפשר לדעת) אז אפשר לחלץ את (a,b)!

הסתברות 1 (שאי אפשר לדעת) שני זוגות משפיע $d\neq$).

הערה: השקפים מאתר הם רק תוסף. חשוב שמרבו עשיעורים!

נושאים עיקריים מהצפנה הקלאסית ונראה איך הקריפטוגרפיה המודרנית עונה מלה.

- קריפטוגרפיה מותנית / תיאורטית - הקושי של השגרה מותנה בכושר החישוב של היריב, מבחינת הטקסט, בחינת הזמן וכו'.
- קריפטוגרפיה עם מותנית / מושלמת - עוז משנה כמה ואוו משאבים יעמדו לרשות היריב הוא יוכל לשבור את הצפן.

הקריפטוגרפיה המודרנית היא מותנית - ובאופן נקודתי ניתנה שניתן אין מספיק זמן או משאבים או מיקח כדי לשבור את הצפן. הצורך של הקריפטוגרפיה המושלמת היה לנו פרקטי משום ש המפתחות יקחו מאוד ארוכים. אם המפתח צריך להיות באורך ההודעה כי ככל שניתן לאחסן מידע. תוך מצי, קשה לתחזק מפתח כ הזמן. היינו רוצים שהתלות מפתח פחות אנו. אכ יש אוצה trade-off בין השאות שבין הפרמטרים.

הקריפטוגרפיה המושלמת היא שימושית רק למי שיודע לשמר את המידע (לפחות זמנית, ובנים שמאוד ככה הזכרון מפחות מידה- כי ברוב מרק ששלית "מוכן פשוט-שמקומה יש" ויתקין שם את המפתח ההצ פעמי).

הצפנה סימטרית:

קריפטוגרפיה סימטרית

אימות סימטרי:

פתרון בעיה זו הוא סיפרית שהמטרה קיומה היא אך ורק שלמות המידע (בניגוד ל- Check-sum שמטרתן את התוכן של ההודעה וכמו כן גם תקיפות מתקשרות). כאן אותנו מעניין שהקוד יהיה מסוג שקל לחישוב אך קשה מאוד לשבור אותו ובמה יבדיל ביניהם ההודעה כך שמבוי יהיה מטרות בנכונות שלו. שהם, ההנחה כאן היא שני הצדדים תיאמו מפתחות.

אין בעיה אם ליבשין ואת קוראת אמת ביניהם להשתמש במפתחות שונים.

פונקציות גיבוב (hash)

הרעיון הוא שנוכל לעבוד עם קלט באורך משתנה ולקבל פלט באורך קבוע. כך אשמש מפיקי לאימות.

"תנאי על בן- אחים ומשמש בכך לאמת הנתונים שאנחנו מניחים שאנו מקבלים הודעה שלא שונתה בדרך.

קריפטוגרפיה אסימטרית

הצפנה:

אם A רוצה שישלחו לו הודעות מוצפנות היא צריך שני מפתחות - ציבורי והמפתח הפרטי שלו. M או שרוצה לפענח הודעה ל- A ישתמש במפתח הציבורי של A ורק כך שאפשר ל- A לפענח את ההודעה ואז שאין אף אחד אחר.

תזכורת:

כאן יש לך אתה יכול לחתום באמת כולם יכולים לוודא ...

קריפטוגרפיה סימטרית / לא סימטרית

- מאפיין הא־סימטרי לא פריט יחשוב תיאום מפתחות אלגו הפלא מפתחות ושוב נפגש את לכך.
- מאפיין הסימטרי יקר יותר כי לכל שני אנשים פרים מפתחות ואלו מאפיין הסימטרי יש למקלף מפתח אחד והמה אנשים יכולים להשתמש מאותו מפתח ביחד.
- לה הבעלי קריטי: מאפיין הסימטרי אי אפשר לחשוף אי התכתשות. מאחר שלאליס אבוב יש את אותו מפתח הם שולחים. כאן שאליס יכולה לחתום על ההודעה עם מפתח A כך יום בוא יכול לחתום על אותו מפתח ואז אליס באמת יכולה לטעון שבוב חתם ולא היא. מאפיין הא־סימטרי לך לאליס יש את המפתח הפרטי שלה אחד אחד ויש מאגר המפתחות ציבוריים שלכולם יש גישה אליו.

יש לך איזה אלגוריתמים קריפטוגרפיים

- יפוצים לפחות
- פונקציות ג'. חשבה
- פונקציות ג'. מאשלה
- פונקציות ג'. גיבוב
- אלגוריתמי פיתוח:

מה שמפיב נשתמש בו זה אלגוריתמים יפוצים לפחות - אלה אלגוריתמים שמפיב אלגוריתמים שאנחנו פיתחנו הם לא מאמינים ואנו כאלו שנשתמש בהם. מאיזה מהם נשתמש? מאוד קשה לפתח אלגוריתם קריפטוגרפי טוב, לכן

ההבדל הוא באופן בו מגיעים לאותו מישהו לשתף פעולה) ומעניין אותנו המפתח.

- chosen plaintext - כאן יש לנו את היכולת לשנות את הטקסט שיצפינו
- known plaintext - כאן מעניין אותנו למצוא את המפתח הטקסט הגלוי.

השורה היא אם מהטקסט המוצפן אפשר להבין את המפתח או את הטקסט הגלוי, הרי אם היא לא ניתן שהאנטן מה שהוא בסך הכל צופן

- known ciphertext - ההנחה היחה שלה תמיד קיים - מישהו מאזין
- קריפטואנליזה (תוקף אלגוריתם נאמן את התקשורת מסוים כלשהו)
  - ניתוח מפתח / סיסמה
  - מציאת מפתח / סיסמה
  - פריצה לאתר / שרת

סוגי התקפות

ההצפנה או אפשר להשתמש מניתוח תקינות נפוץ לשנות את הצופן.
מצאתי מהתוקיות שלו מאה שקשור לסטטיסטיקות עם השפה וגם אחרי רגשות עלעוטין וגו אפשר לפתוח אותו. השניה, אחרי הפתיחה הקובץ נראה מאוד את תוכלות עם שאני מטקסט וגוו אם ההצפנה טובה הקובץ נראה אשונה, הכוונה מוצפן או אפשר לפתוח אותו שפתיחה בפ"כ משתמשת את זה שיש קצת פונטים וגב מצפינים. יש לזה שתי סיבות: אב אפשר לפתוח ואפשר להצפין. אבל אם נדעים יש לפתוח ואת להצפין אלגוריתם הפתיחה אנחנו יכולים לשנות את הקובץ. בפתיחה אין אפשר שיפוץ שפתוח כי לא ההצפנה מאוות שקובץ פתוח לבן נראה אין בכוונה. בניגוד לשיפוץ

הצפנה לגמרי

על ב-1000 מיליון וגוון מה הצפנה סימטרית אפשר של כ- 1000 מיליון.
- בא ארוך אפי - כי אחרת לא נוכל לבא פתקטי. בהצפנה סימטרית אפשר
- בא קצר אפי - כי אם יש אפשרות טקסט אל תפוש מאלה ומאז אין את המפתח.

מהלה אפשרי טבלה

שכן, האלגוריתמים המודרניים בדר"כ תשפים עוד הקפדות נוספות.

בסיס מודרני

יש שני סוגים מרכזיים שהעולם המודרני משתמש בהם:

- Block Ciphers - המשמעות היא בלוק שמגיע קבוע ידוע למדי. ההנחה היא מעבר מאגף מסוים. אחרי הצפנה יוצא מפתח מוצפן. כך אפשר להצפין מפתח מעולה.

- Strean Ciphers - ההנחה מאיפה בכלל שם מגיעים שאנו מתוך מפתחות.

כפי מפתח מרכיב בסיס ההצפנה מונו-אלפבתית. E קטן יחידה שניים ו-K קבוע ליניית תודש. אם מכאן הלה בא אותו מפתח פעמיים אז נשתי הפעמים ההצפנה של המפתח תהיה לחה. וגם לה אומר שאנחנו לא יעשבור את לה?

את |M| קבל ואנחנו יודעים את ה - plaintext כי אנחנו יכולים מקורות רמה פשוט לעבור על כל ההצפנה. אפילו אופטימלי |M| הם 256/128/64. בוודאי |C| לא יכול להיות קטן מ- |M|. הוא יכול להיות יותר גדול מ-|M| אבל בדר"כ לוקחים |C| = |M| עניין שעובדים עם 64=|M|. כי מספר התצורים הוא 2^{64} מתוך מספר התמורות הזה $2^{64}!$ ⇐ אנחנו מדים את כל התמורות אל $|K| = \log 2^{64}!$. כמה ערך!!! אל בדר' 256/192/128/64 = |K| טוב לא עובדים עם כל התמורות. לא כל הפרמוטציות הן טובות. יש מלא לכן טובות? להיות, התפלגות בדי שאנחנו מיתי, סימום של הסיכוי מבחינה מציאותית, ואוד הגון. אל עולם התמורות הטובות הוא יחסית שוברות. איך מוחרים משם עולם הקטן אומר יותר?

מהם המפתחות לתמונה טובה?

- שתי הדוגמאות למעלה יפיקו באופן שונה מאוד
- שני מפתחות פושעים יפיקו שונה מאוד
- ועוד!

המסקנה היא שאנחנו מתעניינים בתמונות איכותיות - תמונות שהעין שלנו רק אומרת הואו איכותי. לתמונות איכותיות אין מתכון. על מה שראוי אין שום דרך לתכנת את הזיהוי.
הבעיה עם כל היגו שניסינו לאפיין מהי תמונה יפה קשה.
אבל מה שעושים היגו שמשתמשים בתמונות פסאודו-איכותיות.
הן נמצאות איכותיות אפשר יש אינטרס פשוט שמזכר אותן מבחינה דטרמיניסטית...

מבנה של בסיס בלוק

שמייצר אותן. ככה מוותרים על הדרישה שהאלגוריתם שבעזרתו אנחנו מצפינים יהיה סודי. מה שאנחנו צריכים עכשיו זה שהמפתחות יהיו אקראיים ובלתי תלויים.

מבנה הבסיס מורכב מסבב של פונקציית סיבוב שחוזרת על עצמה r פעמים. בכל סיבוב מפתח אחר. הפונקציה מקבלת כקלט את הטקסט הגלוי ואת המפתח הנוכחי ומחזירה את הקלט לסיבוב הבא. בסוף הסבב מבצעים טרנספורמציה סופית שהיא ההופכית של הטרנספורמציה ההתחלתית. כך מתאפשר לבצע את ההצפנה וההפענוח באותו מעגל בדיוק.

סכמת Feistel

מחלקים את M לשניים. את החלק הימני מעבירים לפונקציה שאנו רוצים שתהיה חד-חד ערכית. את החלק השמאלי מכניסים לפונקציה f שמקבלת כקלט את החלק הימני ומפתח סיבוב K_i. בתוצאה מבצעים XOR עם החלק השמאלי ומכניסים לפונקציה אחרת. את כל הפעולות האלה עושים r פעמים ומסוף מתקבלים את הפונקציות ההופכיות שעושות את ההצפנה.

סיבוב 2=r

מה קורה אם מחליפים את K_1 ו K_2? M שווה את C? את אותו הדבר? פריצת פונקציות בסיס בלוק בדרך כלל מתבצעת על ידי ניחוש המפתח.

לסיכום חלק ראשון...

שמאוד יותר נקניע מי נשמע אמן ג'ן אוטומט. מה זה אפילו מנשמע אוטומטי אפשר לו פתור כוונתי אמן לפני 30 שנה היית לזה משמעות. היום יותר קשה מה יותר קשה מה יותר מסובך. היה אפשר לשקול מן מספר התוצאות האלגוריתמים. כמובן היום שאנשים מה כאן מהשפעה מבקשה מישהו לשלוח. מי נשמע מה יותר קשה הוא שהיום יהיה יקר מאוד מהבקשה. הסיבה לכך היא שאין התקינה ברור שהאלגוריתם יהיה יקר מאוד מהבקשה. אמת אומרת מאוד מאוד שהיה שם היה יהפוך לסטנדרט IBM היתה בריכה עמוד עם הסטנדרט. משהו מאוד מאוד שהיה שם היה לנו מסר שומרים. ע אחד יכול לעשות את כל הבדיקה הזאת שמשתמשים אלגוריתמים. אגב, כפי שלפני שומרים סטנדרט DES. היא מפורסם באלגוריתם, אולי זה שם שמכר סודי. שיניים ה-DES פותחה סטנדרט DES. היא מפורסם באלגוריתם, אולי זה שם שמכר סודי. לכתבה זה. אבל האלגוריתם נהנתן מאשר כמה שנים. ה-NSA הכניסו באלגוריתם כמה ה-DES היא מאוד האלגוריתם הראשון שמכרים שה-IBM ומכרה שהאלגוריתם משוכ האלגוריתם. ותמכרה כאן IBM הבינו אינטרנט לשוק האלגוריתם. בשנות סוף שנות ה-90 של המאה הקודמת תמכרה אלגוריתמים היו שיש מכר להשפיע באן קצת בשוק

DES - Data Encryption Standard

כאן OTP.

אבל הבעיה של k יחד עם x היו לפחות לפחות שמכרה אלגוריתם, אשהו שנכתב מכונה פטרוניסטית ואותנו יש בריכה שאותו בשבר יהיה אבן הבית השני. בדפים שה-k יכול אשהו אבריו. או אפשר לעשות אשהו לעתיד אלגוריתם כי כל מה שנדרש עושה את הבעיה להם פעולת ה-XOR וכך מביא אותנו לאפשרות. היינו הבעיה $0=k$ פשוט אומרת להשתמש בהתאם לנוכחות של x.

אבל באותה מידה. למעשה, אפשר לעקוב אחרי ההנחות וראות שהפונקציה אז לא, שכרי אותנו תמיד משתמשים בה באותו הבית. כמובן לא ע הפונקציה לפחות בקלות על הפונקציה - האם פונקציה מוחלטת ל כן! לא משנה אם היה הפיכה מסובך היטב או מסובך מאוד...

פתרון אפשרי, רק בדיק שיהיה אלגוריתם שאומר אם משתמשים במפתחות קטנים את המפתחות. אמן מה שאנחנו תאום מה סכמה היא שאותה קלסה יכולה אם סכמה הכאות היא לא אלגוריתם. היא לא אומרת מהו האלגוריתם, מהי k ואיך

נתון שב DES: $|M|=64$, $|K|=64$ ו- $r=16$

האם נכון שהמפתח K_i הוא באורך 48 ביט.

- יש פרמוטציה התחלתית וסופית

- יש לבצע את ההצפנה בתוכנה כך שיהיה מאוד נוח ויהיה ניתן לשנות את ההצפנה מבלי לשנות את החומרה.

- הטקסטים שעוברים בפלט אוכלוסים מספרים ואותיות, אך הקלט ASCII שהוא יכול להשתמש בכל הערכים השמיניה. כך יהיה אפשר לתאם את הקלט ובכך לשפר את ההצפנה.

אחת, הפרמוטציות ההתחלתיות והסופיות אינדיקציה ל DES בקיצור ואפשר לכאורה אותן מאיתר.

הפונקציה f מקבלת $32=\frac{64}{2}$ ביטים ומשנה expand. 48 ביטים ואז מתבצעת עליהם את ה-XOR עם K_i. ולאחר מכן הם עוברים S-box שמקטין 6 ביטים ל- 4 ביטים. סה"כ יש $8 \cdot 4 = 32$ ביטים. אלו הם את שלב ההצפנה עוברת ל- 8 הקופסאות שמקטינות 6 ביטים לאינדיקציה 4. כלומר יש 32 ביטים. ואז הם עוברים Perm ומתקבלת תוצאה סופית.

אחד הדברים שמקשים על ההצפנה הוא שהיא מתבצעת בסיבוכיות זמן לינארית. כלומר אם יש לנו מחשב שיכול לבצע מיליון פעולות בשניה, אז הוא יכול להצפין מיליון ביטים בשניה. אבל אחרי 16 סיבובים של הפונקציה f הקלט כבר לא נראה כמו הקלט המקורי. ולכן אי אפשר לשחזר את הקלט. אבל אחרי 16 סיבובים של הפונקציה f הקלט כבר לא נראה כמו הקלט המקורי. ולכן אי אפשר לשחזר את הקלט.

תיק כאה ימים ...

10^9 פעולות בשנייה ויש כ- 10^4 שניות אפשר כבראה לשבור את זה אבל היום אפשר בקלות לבחון 2^{56} אפשרויות. $2^{56} \cong 10^{17}$. אם משהו עושה התנגדות שהוא לא בשלב שהוא קיים, היהו מלחצן את הזיקט מקין שום שקבר מאוחרים הלה לא בעתר סתם! היום הרעיונות הלה תלוי, אבל לא בשלב שעתה $*$ הוספים על DES על סיבוכים את מספר הסיבובים הכאופציות. הבעיות שיש לי היא שהוא יכול להיות מסובך. (עשה מתקיר יכול) ואחרי כן אפילו לבוא אחד ששוב יש ה שפלה אפלה. אותו הדבר עם הסתה. בשלב ס מה מושג ב"י ה-expand וה-permute. אחרי זה סיבובים ס בוא מגיע לכן אפילו

מה עושים היום בהתחום הקריפטוגרפי? רוצים ש- DES לא יהפוך תלוך (כי המפתח שלו קטן מדי...). אז מה עושים? יש שתי אופציות:

- להחליף את DES
- להחליש את DES } במידה מסוימת את שיטת...

הבעיה את DES היא גודל המפתח. האלגוריתם עליו הוא מתבסס טוב. אז אפשר לעשות מ- DES שינויים אינדירקטיים. למשל, אפשר לעשות איטרציה של הפעלת ה- DES על הקלסאות ה- S ואז לבצע את התפוש הנמלא. היתרון הוא שלא משנים את האלגוריתם עצמו, אלא רק את המפתח שלו. אפשר לעשות שינויים חיצוניים.

DES - הפעלה - Double DES

אפשר לנסות לבדוק את האלגוריתם פעמיים עם מפתחות שונים ואז המפתח הוא באורך $|k_1| + |k_2| = 112$ ביטים. אבל $2^{112} \gg 2^{56}$ אז התפוש מאוד "יקר" מבחינת זמן. זה לא אידיאלי לחפש 2^{112} מפתחות...

אלא, במידה שאפשר לעשות מ- אלגוריתם - פעמים לחפש אותו פעמיים, או בהתחלה אלגוריתם אחד ואז אחר...

<u>מלכודת:</u> עם אלגוריתם הצפנה E האלגוריתם Double-E הוא לא הרבה יותר טוב מ- E!

<u>Meet in the Middle</u> בדרך הוגתות לשמש פרמטרים של DES אמנם לא יהיו לנו אלגוריתם אחר.

עניין שהמתקפה היא known plaintext. נניח שיש לנו את M ו-C.
K_1. נרשום את כל האפשרויות לחלקים של M ו-K_1:

$$X_1 = E_{K_1^1}(M),\ E_{K_1^2}(M),\ \dots,\ E_{K_1^{256}}(M) = X_{2^{56}}$$

ואז נעשה פיענוח ל-C לפי כל האפשרויות האפשריים ל-K_2:

$$Y_1 = D_{K_2^1}(C),\ D_{K_2^2}(C),\ \dots,\ D_{K_2^{256}}(C) = Y_{2^{56}}$$

אם $X_i = Y_j$ אז (K_1^i, K_2^j) מועמד להיות המפתח.
כאן משתמשים בידיעה זו שיש $O(2^{56})$ כדי למצוא את ההצפנות והפיענוחים. אז בסיכום יש לנו את הבעיות המתאימים.
כל צורך למיין את הרשימות, אז סה"כ כל התהליך לוקח משהו כמו $O(2^{56})$
הסיבוכיות היא שיש כ- 2^{48} כאלה מועמדים. ואז I הוא כמו אוסף
מפתחות שמתאימים ל-M ו-C. יש לנו 2^{64} אפשרויות. השאלה מלפני את M ו-C
2^{56} אפשרויות ל-K_1. אם אנחנו אומרים שבגלל שאנחנו יודעים
אנחנו שאלאחים אפיל את I האופן אחד אחד אם אחד מהם נמצא של
C יש לנו 2^{56} אפשרויות ל-I אפילואחד יקח $2^{48} = 2^{56} \cdot \frac{2^{56}}{2^{64}}$.
הכלל ואחשבה: אם נמצא את ההצפנה עם הזדמנות ספר K^* והכלל
שלה C^* אז בהסתברות כמעט 1 נוכל למצא את המפתח הנכון
אנחנו יכולים $2^{56} \cdot 2$ אפשרויות. ואם עדיין ההתוצאה לא יקחת תכי
את המפתח אז נעשה את זה שוב. כמו אומר, אתרי מספר מסוים לא יצלח שם
תלכות אפשר לנחש את המפתח.

נשאלת השאלה - האם $DES_{K_3}(M) = (DES_{K_2}(DES_{K_1}(M))$. מסתבר שיש כמה
הסברות שלא לא שם ...

3-DES

Mode EEE

יש פה יתרון ביטחון של $2^{3\cdot 56} = 2^{168}$ אבל המוסן פוגע יותר שעשינו
ב-3DES-כפול אפשר לשמור את כל ה- 2^{112} כי נשתמש את K_1 ו-K_2
ו-(K_1, K_2) ל-K_3 וזה הפרש מן K_2 ו-K_3.

מה הגדה ארוכים 56 אפתחות.

מאחז ג'דר 8 אפתחות. אל ססים 256 מיליארד גוד 8 אפתחות
הפסים. כך מיליארד 8 אפתחות. אל ססים לבים ה- 2 ואתחים.
הפשוט הזה פשוט ב-91 מ"ב הראשונים. האפתח השני, הוא 91 הבתים
מהאפתח האחרון מיליארד 256 אפתחות קבוצה של 91 ב"ב. האפתח

- 0 הפך אופנו $T+_{91}2$
- + תיבור מפנו $_{91}2$
- $\oplus$ XOR בין בתים
- אין סימנים (או ללאים / זוגיים)
- אפן אפתח 321
- אפן פתוח 64

האפיינים של האלגוריתם:
באופן פורמלי, אפשר אשתמש בו בדרכה מקומות
לה אלגוריתם שפותח ע"י האילים משנת ה- 80. הוא נקרא

International Data Encryption Algorithm - IDEA

ענין האיומים לקבלים ישנים מאותה הדרכה...
אל נדה בקפסה אחת יש עוד פתרון שאותו נקרא פתרון ... זה נותן
אל נדה בקפסה אחת יש עוד שפ. פתרון שאותו נקרא פתרון ... זה נותן
ההצפנה והפיענוח הם שדה. שפק אם $K_1 = K_2$ אל לה פשוט DES.
האן, אם $K_1 \neq K_2$ אל לה 3-DES כי הדי ב- DES פתרון

אופציה נוספת הוא mode - EDE :

מיטון של 211. אל אה שעושים מפ"ב כה שמשתמשים ב- K_1 פעמיים:

או אפיתוח פומיס.

פתרון את הבעיות" רק שמאחר והיו שניים מין הזדמנות פומיס
את מ לה עושים זו פעמים. מ ההפוך המקומות זה- אסא-ים לה משנע
הכנסה יפומצה היו בו I.
$p_y \oplus y_c = p_x \oplus x_c$ ⇐ מין אם מתנים מ- q_x, b_x או עם q_y, b_y
עמה ב נשים ⌣ שמתקים $q_y \oplus a_y = q_x \oplus a_x$ ין
סימום כלה יש תכוף שאיא הכופני שם קבאו אפיו עם אותם האפיתוח?

סימום ציאי

הזקקה מאוחרת

סימום ציאי:

הזקקה מאוחרת

ב שני אפיתוח

ב סימום אינאי משתמשים ב אחת אפיתוח סימום ציאי משתמשים

AES - Advanced Encryption Standard

ב-1997 הוקלו הקולות לסטנדרט החדש ובשנת 2000-1999 נמצא אלגוריתם שעונה על שני מתמטיקאים מלגיום. כך אלגוריתם שקוראים לו Rijndael, ויכול מאוד שאתם כבר מכירים. הוא נבחר משיקולים של מהירות.

המאפיינים שלו הם:

- מתקדם מאוד עד
- אפשרי 256/192/128 ביט
- סיבובים 14/12/10 בהתאם לאורך המפתח

מפתחות סימטרי מאפשרות יעילות, אין נפוץ אין.

יש מגוון אירועים שמהווה של מלא state

שם רשומים Byte, אל סדר יש 256 ביט. מהתחלה מעתיקים את הטקסט הראשון. ואז:

1. XOR עם מפתח סיבוב
2. ההחלפה (S-Table) יש 91 Bytes ויש טבלה שאומרת איך להחליף אותם. פשוט מחליפים byte בטבלה!
3. Shift Rows - לשורה ה- i עושים Shift ימינה i פעמים. (כאשר $0 \le i \le 3$)

4. Mix Columns - עם סור מייקרים ל תמיד הפשוט מדליק טבלה שיודעים. ואז עושים XOR של 1,2,3,4 ואל 1,4,3,2 וכן הלאה וזה נותן 4 בתים חדשים:

לא קשה לראות שהאלגוריתם הזה הפיך. בדרך תהליך ההפוכה אפשר להשתמש שהאלגוריתם הזה אותו אין הסתם לך שלא לא זה...

הצפנת טקסטים ארוכים

עד עכשיו דיברנו על הצפנה של בלוק אחד. מה עושים אם יש יותר מבלוק אחד?

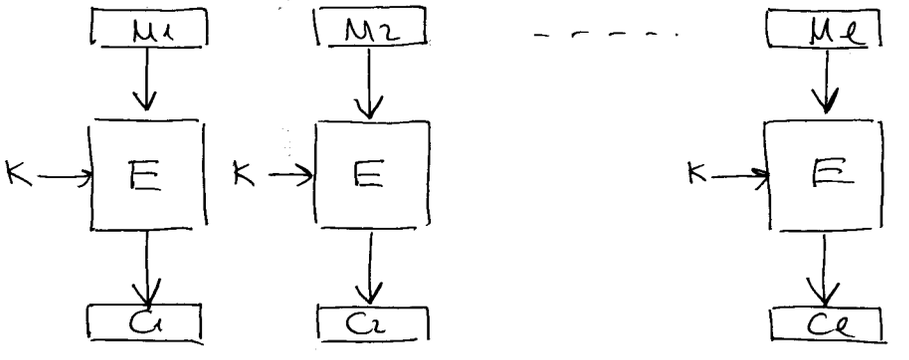
Block Ciphers

מסתבר שיש הרבה כאלו הניתן לחלק למספר בלוקים. אם אורך ההודעה קבוע מתבקש באורך הבלוק לחלק ברצף ולהצפין על כך.

יש 3 שיטות עיקריות להצפנה בבלוקים:

- ECB - Electronic Code Book
- CTR
- CBC

ECB נניח שההודעה מורכבת מבלוקים $M = M_1 M_2 \dots M_e$



כל בלוק מהודעה מוצפן באופן עצמאי, תלוי בעליות האלגוריתם. את רמשו, אם הבלוק באורך 64 ביט את כה כאלו שיש עוד קוד מוגן אלף בתי, יש 2^{64} אותיות באלפבית.

יתרונות: + הצפנה מקבילית
+ פענוח מקבילי
+ הצפנה/פיענוח בבלוק בשהו. (למשל, אם רוצים להצפין רק חלק מהקוד)

תכונות:

- מפתחות להם יכולים גישה וצפה נוסף מידע. למשל אם המשכורת של כל מוסיפים במפתח ואנחנו יודעים שהמשכורת שלו ושל ה' מוסיפים מאותו אופן אז המשכורת של כל ושל ה' להות, שצפה אין שהו מידע. אולי כל לה אני וכל המשכורת של ה' יודעה.
- אין הגנה של סדר ותוכן ההפתחים. לה מתקשר יותר לאומות תוכן מאשר ההפנה, אפשר שהאריך / קיצר / לשנות/ שהפתח אנו הידעה ושום דבר שכן מונע את לה.

CTR (קיצור של המילה "מונה")

כאן מנסים שהתכתב של הפעה שפתוחים להיות מוסיפים מאותו אופן.

למשל "1" יכול להיות ה"ז הבינארי של 1 ב- 64 ביט.

תכונות

+ מפתחים להיות מוסיפים באופן שונה
+ יש הגנה מסוימת של סדר ושל התוכן - למשל אם מתוך טקסט מוצפן ואנו מפתחים את המסר מהתה שהמפתח הזה הוכו קיבלנו שם אנגלי וכול להיות שיבוש משהו שכן הגיוני ואנו נדע שמשהו שכן מסדר. אלא יש סיכוי שיהם שכן נשים ש שמשהו שכן מסדר מכן הגיאה לכן אותה.
+ פיצוח מקבילי

הערות:

- פרוש סינכרון חיצוני
- זהו משמעו XOR עם המספרים הפי שהם כי אנו נמשיך ההבדל בין 1 ל- 5 הוא בגודל אחד מלבד ואנו יודעים קל לתקן את השורה. לכן בד"כ ה- XOR משתמשים כפונקציית ערבול וה XOR הטוב ערכים $h(1), h(2), \ldots, h(l)$.

CBC - Cipher Block Chaining

כיה השיטה הכי נפוצה.

הערות:

- ההצפנה סדרתית

יתרונות

+ פיענוח מקבילי, כי M_i תלוי רק ב- C_i ו- C_{i-1}
+ פיענוח אקראי - ניתן אפשר לפענח רק בלוקים שרוצים
+ בלוקים לחים יוצרים שונה
+ סנכרון עצמי - בריא שיש שגיאה מתפשטת רק שני בלוקים
שיוצא לנו אופה אותנו הפענוח.

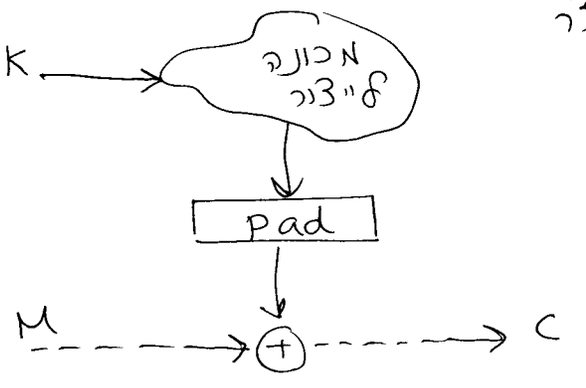
---

initial Vector
IV הוא ה- XOR של הבלוק הראשון. אמת יתכן הבלוק הראשון אותנו מאופן אקראי. בדיק ע"מ שנות את התקשורת כי אותה ההפצנה פעמיים יוצרת ענין אותה.
בדיק ש- IV אקראי אינ שיהיה מאופן שונה. ע"י פעם שמצפינים את הקובץ הוא יוצפן באופן שונה.
IV אקראי, מאוכל את ההצפנה. כמעט ע"י האלגוריתמים משתמשים בלה בדיק כזו או אחרת.

נשים ♡ ש- L_2 תלוי בכל מה שפנינו. לכן אפילו אם משתנה
מיט אחד מהדרגה מהסתברות מאוד יבוה L_2 ישתנה. לכן
אפשר להשתמש ב- L_2 בשביל לאמת. אבל אפשר לשנות אותו
פעמיים אומר כה לכן תמיד כי כה לכן אחיד מפני התקפה
שמותקת את L_2 האחרון של ההודעה ומתלפה את ה- L_2
של האימות ב- L_{-1}. לה נקרא truncation attack
אבל אה שבדיק עשוית כה שהשתמש בשני מפתחות - אחד
לה בפנה ואחד לאימות.

Stream Ciphers

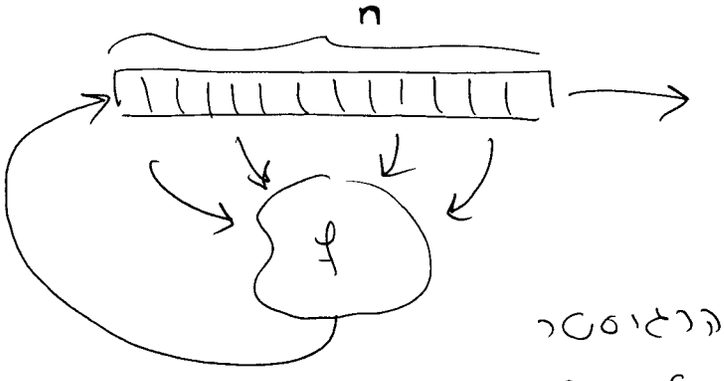
ההגדרה מבוססת בבסיס של מוטים ועושים XOR עם
איזה PAD שאיזהר
ע"י מכונה סופית
דטרמיניסטית אחלק
מאוד מסוכנת



יש שלושה מצבונים כלליים:

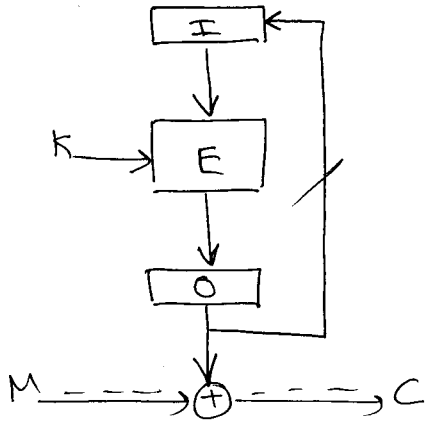
- LFSR Linear Feedback Shift Register
- OFB Output Feedback Mode
- CFB - Cipher Feedback Mode

LFSR



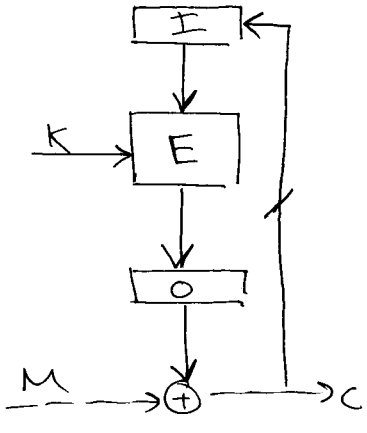
מפה הפונקציה f
היא XOR.
הערך הראשוני של הרגיסטר
הוא המפתח. אבל לא נלחה את 0 בתור מפתח כי אז כל
הסמן "שאר 0. אז יש 2^n-1 אפשרויות שאינן 0.
הפולינום האופייני שאם נסתכל על הרגיסטר כפולינום $1+x+x^2+\dots+x^n$
אם הוא מתחלק פריק אבל לה יפחה על 2^n-1 האפשרויות.

CFB



אתגרים מעניינים התחלתי של תוכן מעשי.
מגבילים אותנו עם מפתח סודי וכך
משמש להצפנה של M. השיבוטים
של O עוברים הלאה תוכן I והתהליך
ממשיך.
מנגנון "בזבז" ה- O הוא אוטומט סופי.
X הוא סודי. מבנה שאת I השורת והמקלט יודעים עליה.
נקודה למחשבה: האם I בהכרח להיות מתואם באופן סודי או לא?
מאחר שהוא סופי, אפשר לחשב את מספר המצבים שלו וכך בקלות.
יתרון נוסף הוא שאפשר להכין O-ים מראש וכך כשמגיעה הבקשה להצפנה וכלה עקרונית מהר מאוד תלוי
במהירות של E.
חסרון של השיטה הוא שהוא שמרני סנכרון תזונתי בין שולח ומקבל.
אם מצפינים מעט מיטים אפילו תוך כמה דקות אז בהכרח
או משבו כדי, עדיף את ה- I להתחיל כדי שאפשר יהיה
להסתנכרן מחדש.

CFB



כמו קודם אוטומט סופי והוא יחסי דטרמיניסטי,
שהרי כל תלוי ב- M - מחזה ההוד עוד
הוא אינפוטי, ואין מתכוונת לחוד עוד.
זו אפשר להכין מיטים מראש, אופן
ואין בעיקרון יש סנכרון עדיין בדומה ל-
OFB)
נשים ♡ שבאן I הוא לא סודי בכלל. הרי I מגיע מה- C
ואת C כולם מכירים מראש.
מה רותם פס ומשונה? אפשר מיט מוכן ואפשר מיוחס התוכן.
משהו קטן אולי מטות יותר אפל מלמלנו מתושה.

נניח שאתנו עובדים עם מעבד של 64 ביט ונניח שאנחנו
מיטים מתקשורת, אפשר אתנו לא יודעים כמה. ברגע
שהתקשורת תגרה ואתנו מקבלים 64 מיטים חדשים
תקינים אצ אתנו לראורה שום מסונכרנים. אפשר לדיק
לשים לפ שהמיטים החדשים הם בדיוק מעלון אצלו
תגי מעלון אחד ותגי מהמעלון אתרון.
אם המשפ הנוא של ביט אחד אצל אתנו מיד נהיה
מסונכרנים.

אימות

- ישות - אימות אחד הצדדים בהודעה
- מקור - מהי ההודעה ומיכן
- תוכן - אימות התוכן של ההודעה
- סדר - אימות סדר ההודעות או התוכן (מניעת התקפות כי שם מתקיים את ההודעה לעתיקים)
- זמן - אם ההודעה הגיעה בזמן המוגדר (למשל אם היא "השעון היום מהונג")
- אי התכחשות

שיטות באימות

1) אימות ע"י הצפנה - אימות של תוכן אפשר להשיג ע"י הצפנה.
   בתור תזכר לוואי יש פה גם סודיות.
   - הצפנה סימטרית
   - הצפנה אסימטרית
2) אימות ע"י פונקציית MAC (message authentication code)
   כאן השימוש הוא תמיד סימטרי.
3) אימות ע"י פונקציית hash
   - סימטרי
   - אסימטרי

אימות ע"י הצפנה

נתון) באימות סימטרי.
יעקוב יפהפה יש מפתח להה K. איך הוא יכול לאמת
את ההודעה ? יש כאן 2 אנשים שיכולים לראות את
ההודעה היום ולשלוח אותה מחר הפי שהיא, אפשר לקבל הודעות
עשרים הודעות וכן הלאה. הבעיה העיקרית היא אבל שיכול

אלה שני מנגנונים שונים.
אל מקביל האסימטרי אי אפשר להשתיל חתימה בלבד אותה.
אל הנכון שהמפתח הציבורי של A יפרץ ויתן ל A יכול לחתום, אל אפשר
להניח ש- A אכן חתם.
אל מקביל הסימטרי – אי התכחשות – כי אם באים לבית משפט
אל התוכן מישהו צריך לדעת לזהות מי חתם ואת מי
אי-התכחשות. אקלר - כי רק A יכול לחתום. תנאי - כי כפי שהוסיף
אותה נפיץ אל הסוכן ל המקבל ב אקראי מפל משמעות מפיקנות
שהוצפה. כל שש יותר אפשרויות כאלה, אזי שהוצפה יש מפה
משמעות. הפרק להקטין את הסיכוי שכזה היא להוסיף מפנים

אזמת ג': הצפנה אסימטרית:

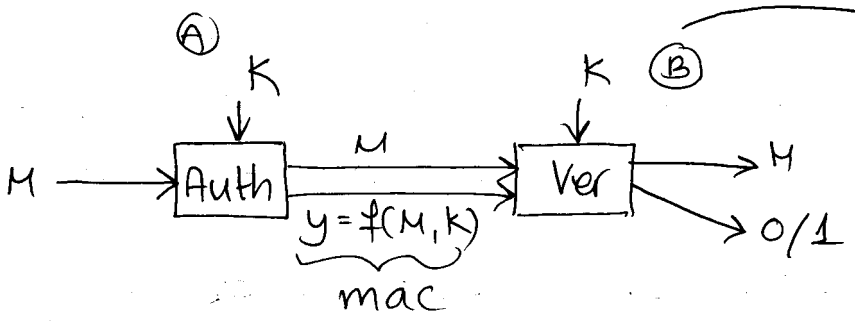
אזמת תקן? אזמת אקלר ולכן ברור שאין שם כלי

מאחר שמפתח הצפנה הינו פומבי ל אחד יכל לשלוח הודעה
אל אין מושג מי שלח ואין מושג אפ אין מישהו שהפס אות
ההודעה באמצע ושינה אותה! אל כך לא שווה פום.
מה שפרק לשלוח בעולם הא סימטרי הוא כך:
כך נקרא תתומה
דיגיטלית. את תתימה
יכול רק A לשלוח
ופל אחד יכל לוודא

שכך אכן הפא. רק אפשר להשתיל אזמת אקלר, אזמת תכיל גם
אי-התכחשות. אקלר - כי רק A יכול לחתום. תנאי - כי כפי שהוסיף
את התוכן מישהו צריך לדעת לזהות פומבי פשל, משמעות ואת מי
אל A יכול לשלוח. אי התכחשות - כי אפ באים לבית משפט
אל המפתח הציבורי של A יפרץ ורק A יכול לחתום, אל אפשר
להניח ש- A אכן חתם.
אל מקביל הסימטרי אי אפשר להשתיל אזמת והצפנה בפת אחת.
אלה שני מנגנונים שונים.

הנתונה שמשתמשים בהן:
- מרחב ה- M-ים האפשריים הוא גדול מאוד. פירוש הדבר שזהו מושג מאוד $\subset$ וקריאי את הסיכוי שהפיענוח שלו ע"י משתמש הוא קטן מאוד
- אלגוריתם ההצפנה והפיענוח הוא טוב
- קל לבדוק במחשב אוטומטי M-ים תקינים.

אימות ע"י MAC



אין פה סודיות בתוכן. התוכן עובר יחד עם איזה תג מאחר שיכול לבדוק אותן רק מי שיש לו את המפתח K. הצד המקבל קושר שכח את התוכנה של $f(K,M)$. אם יבדוק y את כך בסדר. מחדש כנראה שמשהו פגע בתוכן. Auth ו- Ver קושרים בדיוק את אותו הדבר. שניהם מחשבים את $f(K,M)$ אלא ש- Ver משווה את התוצאה עם y שהגיע מהתקשורת. f איננה פונקציה הפיכה. לרוב, לא כדאי שתהיה הפיכה.

תכונות:
- מתקיים כי (y,M) ומחלץ את ה מפתח
- מתקיים כמ"ז $(y,M) \neq (y',M')$ הפולים.

ראינו שרוב משתמשים ב- MTO (Mac one time) אותן לפני מטרידים מהדברים האלה. הבעיה היחיד שאותן נדרים MTM (Mac many time).

תכונות:
- פולים y-ים אותיך / אקראי
- M-ים דומים שומרים ל- y-ים שונים.

פונקציות הגיבוב הן טובות לייצור MAC מי ישלהן התכונות הנ"ל.

נוסחה: $y = E_k(M_1 \oplus M_2 \oplus \cdots \oplus M_n)$

כאשר M_i זה המלוקים של ההודעה. כלומר אם פונקציה טובה, מרגע שפורים ה-y-ים אקראי וחיד כי E_k פונקציית הצפנה טובה אפשר יהיה M-ים שונים כאלו אותו y. למשל אפשר לשנות סדר אחד משני מלוקים וה-y לא משתנה, או שאפשר לשנות את סדר המלוקים וה-y לא ישתנה.

נוסחה: $y = (IV, C_L)$ ואת $C_L = CBC\text{-}E_k(M)$

IV אקראי כי כדי שהוא אחיד, C_L חייב להיות אקראי כי כך יוצא ב-$CBC\text{-}E_k$. אפילו אם כלומר אפשר לנחש מה הודעה (תיאור של כמה קלות).

מסקנה: יש צורך שהוסיף שהודעה יהיה את הפרטים הבאים:

- כמה
- מספר סידורי
- אורך
- זמן.

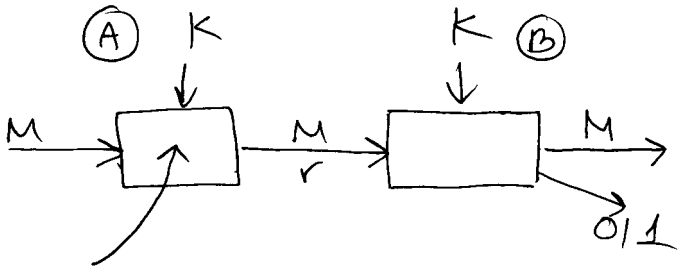
כפי שהמין על הודעה ע"י MAC צריך להוסיף לה זמן הרבה הרבה פרטים משמעותיים.

התכונות הדרושות של MAC:

- חישוב מהיר
- MAC רב-פעמי
- MAC קצר

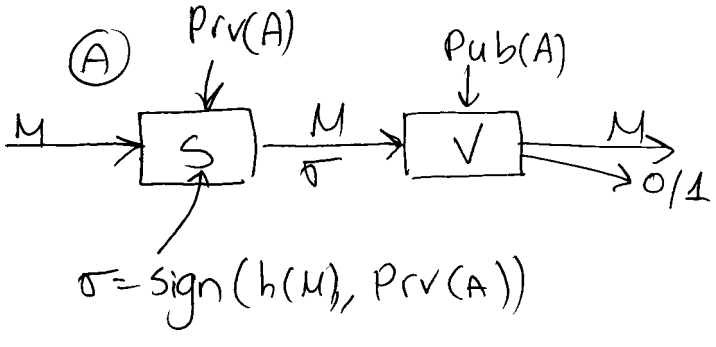
אומת ע"י hash

סימטרי:



יתרונה מזהיים את הפונקציה הדרוש שלה (איך שהוא מאופיין קדם שם ההודעה) ואז על המאפיין הזה ועל התפיסה מסתמכים איזו פונקציה. במיפון מסוים, אפשר להסתכל על זה כ"דגר של MAC. מתחילה תידונים לה להה. אלא שב- MAC משתמשים בפונקציית הצפנה וכאן לא שונה. שכן מצפינים את ההודעה אלא מקדים אותה (בדיב לשלוק אחד) (על פונקציית הדרוש נפתר בפרק הבאה). לה נקרא HMAC-h.

אסימטרי:



האומת היא שמעולם הסימטרי לא מה שקרא תתיימה דיגיטלית. בדיב שכן תתיימים על ה- M במשלואו אלא רק על ה- hash כי הפונקציות אוטיות שתתיים על הודעות ארוכות לוקח אחלאמן. שנה, ההבדל היה שקודם מקדים את ההודעה ועכשיו אותה או משהו כזה.

HMAC - סכמה סטנדרט שמשתמשת ב- 3 פרמטרים: פונקציה h, מפתח k והודעה M. מה שעושים הוא פתוח או ואת כזה:

$$h((h(k \| M)) \| k)$$

שרשור

הרבה סוגי ניתקפות - אי אפשר לגלות מקדים ההודעה

פונקציות hash

המטרה של פונקציית hash היא למפות את קבוצת הקלטים (הקבוצה הגדולה) לקבוצת המספרים. קבוצת הקלטים היא תיאורטית כל מחרוזת בינארית (ייתכן שיש מגבלה פיסיולוגית של המחשב) קבוצת המספרים היא סופי וקטן (משהו כמו 500 ביט)

תכונות בסיסיות של האלג':
- מהיר לחישוב
- אלגוריתם דטרמיניסטי (מפעיל אותו פעמיים בדיוק אותו דבר)
- קשה לניחוש

מצב שהפונקציה היא לא חח"ע (שהרי יש יותר מספרים קטן מאלפי הקלטים) נקרא התנגשות (collision) ורצוי שיהיה כמה שפחות התנגשויות.

תכונות רצויות של הפונקציה:

1- h תלויה על כל הקלט
2- h אינדקס מקום באוסף קבוע מראש כך שכל איבר (ת) } תכונות רצויות
3- h נקלה לחישוב
4- One-way : מתקיים לכל מחרוזת x נתונה ש- $y=h(x)$ את קל לחשב את הערך אחרון ולא שקל מאוד למצוא מחרוזת. קשה לי מאוחר שלא ידוע לכמה מאוד כמה } security רצוי
5- weak collision resistance : עבור מאפיין של הדומן אלגוריתם אם קשה למצוא התנגשויות: מתקיים x_1 קשה למצוא x_2 כך ש- $h(x_1)=h(x_2)$ ו- $x_1 \neq x_2$
6- strong collision resistance: קשה למצוא $x_2 \neq x_1$ כך ש- $h(x_1)=h(x_2)$

אבל במקום לשלוח את K אפשר לשלוח את $h(K)$ אבל אם כבר פעם כי מישהו יודע לחשב את $h(K)$ צריך להתעניין האם היא מספיק קשה, כי אחרת נתשל?

לפי A ו- B מסכימים ! - A רוצה לוודא שמישהו אכן מחכה לו B או שהוא A את B רוצה לדבר שלפני אלא מתחזה A. ניתן לחשוב פשוט שלח לו אומרת סיסמה. אז A יכול לשלוח את K ל-B ולהגיד בבקשה (שלו הסמא מתאים רוצה

② פרוטוקול אימות

אבל קשה לזיהוי. hash במקום בסיסמה. אלגוריתמים שמשתמש ב- אחד הפתרון הוא: אין סיסמה סוד, הלקוח או משהו כלשהו יש לו סוד שמוכר לשרת אבל לא לאף אחד אחר. הלקוח יכול להוכיח את הסוד בלי לחשוף אותו. אומרת שהשרת יודע את הסוד של הלקוח, מתקיים תכונה. אבל אומרת שאם השרת אינו יודע את הסוד של הלקוח אז לא יוכל לחשב את אפשר כמובן לקרוא את הקובץ ולחשב את $h(K)$ אבל אם כלה, אמנם יכול להיות שהלקוח K - $h(K)$ או משהו כזה, אבל יכול להיות שהלקוח יודע את הסוד אבל השרת לא יודע את הסוד. הבעיה היא שהקובץ לא יושרה מאוחסן בשרת. אפשר לפתור בעזרת חתימה דיגיטלית, שנוכל לוודא שהקובץ לא שונה.

① אופטימיזציה

שאושים לפונקציות hash

① אופטימיזציה ④ הצפנה
② פרוטוקול אימות ⑤ MAC
③ פרוטוקול להתחייבות ⑥ חתימה

אבל אותנו מתכוונים שבאמת לכן המקרה כך שלא. אבל, יכולים להיות מקרים שבהם קל למצוא התנגשויות

מ- A אל B אפשר שמאפשרת לו לעשות בדיוק מה שהוא רוצה.

מ- A מספר רנדומלי R ואל A שולח B שולח פעמי. אל מה שלושים פעמי ש- B שולח מתקיים עתיד את x מאותו שבטן פעם אחת משתמשים במספר רנדומלי אחר כך זה אפשר מתי פעמי.

③ פרוטוקול ההתייחסות

A ו- B רוצים לשתף ביניהם סוד או פרט. את הסוד שלהם הם מאמינים לשמור את הבחירה ואלגוריתם הם מאמינים אמנם לה לא מפיקים אפשרי וכל אי שניתן ראשון בטוח אף סיד.

את A ורוצה לשלוח קודם את $h(x)$.

אחרי ש- B מקבל הוא שולח באופן אחר את y. מאחר שקשה לחשוב את x מ- $h(x)$ B לא יכול לעשות.

וכעת A שולח בחזרה את x ו- B רוצה לוודא שאכן $z = h(x)$ אבל הוא יודע ש- A לא יכול לשנות את ההתחייבות שלו כי הוא מחזיק.

אבל, כפי ש- B לא יראה בדרישה תשובה (4) ואולי כפי ש- A לא יראה דרישה תכנה (5) כי אם A היה יכול לשנות את הסוד אחרי שהוא שלח את $h(x)$ הוא יכול לחשב מספר אינסופי שיש להם אותו hash אך מה יש לנו כאן? A מחזיר את התוכן אמנם הוא לא מתחייב על.

④ הצפנה

יש לנו אלגוריתמים טובים של הצפנה. למה שבעל נדרה מעוניין להצפין מדוע hash? יש 3 סיבות הרבה שונה,

אותנו יבינו וכך נמנע השניים, יש אפשרות שאחד מהם יבוא
בשנה נתשמרת מי נשלח, אבל פונקציות hash לבן; נאז אם
נדעים שהשני יהיה יהיה בעלת hash השנייה, וכך
פעמים, פונקציות hash נן יותר מהירות אפונקציות הצפנה.

A ! - B שלחים אחד לשני
מספרים אקראיים ומאחדאים
שמאחדאים הם יצפינו את
ההודעות. כך ה- PAD יהם
שונים וישב כאן סטאודו- אקראיות.

כך לאחדאים אקראי כי P_{i+1} תלוי ב- P_i רק $i \geq 1$, אשר
משפל ההצפנה של h כה קושי לאקראי.

⑤ MAC

$y = h(M, k, Rand)$

$Rand = R$ עליך לבקש שלאחה הצפנה לא יהיה אותו
ה- MAC פעמיים ...

⑥ חתימה

$\sigma = Sig(h(M), Prv(A))$

כאן עושים $h(M)$ כדי לצמצם את ההודעה שהרי חתימה על
הודעה ארוכה לה יעיל.

אוסף מונים פונקציית hash?

מהדרישה ש- h היא strong collision resistant נובע ש"קשה" למצוא x ו-y כך ש-$h(x)=h(y)$. לא נרצה לדרוש קשה מדי. למשל אם y מאורך 10 מיט אפשר לנשוא תפוש מאדה, שהוא $2^{10}+1$ הודעות שונות שהן מאותו אורך... אבל הפלט קשה לזיוף מספיק אורך. בפרט הפלטים מתוחזקים ל- 80 מיט.

מאותה עם N תגובות אפשריות. הסיכוי שלאחר K ניסיונות יש התנגשות הוא $\frac{1}{2}$ להתנגשות אם שווים $K=1.18\sqrt{N}$ ניסיונות יש הסתברות שאין התנגשות $Q(N,K)$ את ההסתברות שאין התנגשות אחרי K ניסיונות. קל לראות ש-

$$Q(N,k) = 1 \cdot \frac{N-1}{N} \cdot \frac{N-2}{N} \cdot \ldots \cdot \frac{N-k+1}{N} =$$

$$= \left(1-\frac{1}{N}\right)\left(1-\frac{2}{N}\right)\ldots\left(1-\frac{k+1}{N}\right) \leq$$

$$\leq e^{-1/N} \cdot e^{-2/N} \cdot \ldots \cdot e^{-\frac{k}{N}} = e^{-\frac{k(k+1)}{N}}$$

אז ההסתברות שיש התנגשות היא לכל היותר $1-e^{-\frac{k(k+1)}{N}}$

התקפת יום ההולדת: בכך שמנקים ש-hash תהיה אפיק טובה לא רק שהפלט צריך להיות ארוך מאוד היות שני, אלא ש"תדי באוסף לפי שורש מאוזן הטענה... אחת ההתקפה תוכל?

יש מאות טוב ומאות רע ואני חלה לחתום עם המאות הטוב, אמץ המלכיה שלי כדלה שאני אותתום של המאות רע. אומאפשר לראות ש-B "שמח מאזן A ו-B אפשר ליזוד מאזן מוקדם של A ושל B כך שהם שווים מאוזנים אפילו שניהם סולטקיה, למשל מאזנים שיש בהם מספר שונה של דוגמים כלהם מאזנים אבל שניהם סולטקיה.

מיפוף בלוק בודד בפונקציית דחיסה של SHA-256:

סכמות אלה שכיחות מאוד בשימוש בפועל: מהירות ואבטחה סבירה.
MD-5, SHA-1, SHA-256 ועוד.

נכון כמה פונקציות סכמות: מיומן מישהו פשוטות הקריפטוגרפים ניתנו את הצורך ל- 256.
אנטכניה. אמנם בחלק שלא כלים יודעים שלא מה שחיק לעשות
וכך על פתרונם אלו. את אפשר אם טוב לנו מאמת עוין יכולנו
או שאפשר לפני שנשלחים את המאמת לעשות בו שינוי טעות
את משכנו לפתרון את הפלט הלוגי אשר שיצרנו את אסף הביטים.
מאלם.

שהתאמה תהיה עם A_i (לא ראוי שה- A_i לפה סעיף ל- A ונשה לפתרון את B_j
ניצה B_j ו- A_i כך ש- $h(A_i)=h(B_j)$. ואז צריך לבצע 2^{64} פעמים של A! 2^{64} פעמים של B ואז נוכל למצוא
לא. אז יש 2^{64} מאכים שווים. סעיף ל- A. את אחד מהם
ניתן שיש 64 אלמנטים שאפשר לפתור את בלתי בפה רוח או

נושאים מתוך האלגוריתמיקה - אלגוריתם

מחלק משותף מקסימלי greatest common divisor

יהיו $a,b \in \mathbb{Z}$ מחלק המשותף המקסימלי של a ו-b הוא מספר $d = gcd(a,b)$ שמקיים את a ואת b מתקיים $d \mid a$ ו-$d \mid b$ ולכל c שמקיים גם את a ואת b מתקיים $c \mid d$.

לכל $a,b \in \mathbb{Z}$ מגדירים את הקבוצה הבאה: $LC(a,b) = \{xa+yb : x,y \in \mathbb{Z}\}$

מסתבר ש- $gcd(a,b) = \min\{x \in LC(a,b) : x > 0\}$

כדי לחשב את $gcd(a,b)$ כדאי לדעת את זה.

טענה לכל $a,b \in \mathbb{Z}$ קיימים $x,y \in \mathbb{Z}$ כך ש- $gcd(a,b) = xa+yb$.

אלגוריתם אוקלידס למציאת $gcd(a,b)$: מסתבר ש- $gcd(a,b) = gcd(b, a \bmod b)$.

כמה אפשר לחשב בקלות את ה-gcd של שני מספרים קטנים ב- מן ומעל מיליון ספרות בשורה. למשל,

$gcd(27,21) = gcd(21,3) = gcd(3,0) = 3$

ההבדל בין אלה מן לעל מיליון ספרות, מובן היטב? פשוט מכמן $O(\log \min(a,b))$. מה המשמעות? - כי אחרי לכל אחד אחנו משנים את יותר בסדר בין המספרים.

אלגוריתם אוקלידס המוכל גם אפשר למצוא את x ואת y שמקיימים $gcd(a,b) = ax+by$.

למשל, $3 = 1 \cdot 27 + (-2) \cdot 21$, כלומר $x = 1$ ו- $y = -2$.

אלגוריתם זה ניתן גם לשנות בקלות שיחשב את x ואת y בזמן פולינומי.

פונקציית אוילר $\varphi(n)$

$\varphi(n)$ = מספר המספרים שלמים n אפיי $1, \dots, n$.
מספרים $a, b \in \mathbb{Z}$ הם זרים אם $\gcd(a,b) = 1$, כלומר
אומר שאין להם מחלקים משותפים פרט ל-1. באנגלית נקרא
מספרים relatively prime, כלומר שיש להם רק מחלק משותף
משותף הגדול ביותר 1. למשל $\varphi(10) = 4$ כי המספרים $1, 3, 7, 9$ זרים ל-10.
לעומת זאת $\varphi(11) = 10$ כי 11 הוא ראשוני וכל המספרים
שקטנים ממנו זרים לו. באופן כללי, עבור מספר ראשוני p
מתקיים $\varphi(p) = p-1$. ובאופן כללי $\varphi(pq) = (p-1)(q-1) = \varphi(p)\varphi(q)$
יש נוסחה כללית שאומרת $\varphi(n) = n \prod_{p|n} (1 - \frac{1}{p})$
לעומת, כדי לחשב $\varphi(10) = 10 \cdot (1-\frac{1}{2}) \cdot (1-\frac{1}{5}) = 10 \cdot \frac{1}{2} \cdot \frac{4}{5} = 4$

חבורות

חבורה היא קבוצה S עם פעולה $*$ כך שמתקיימות
התכונות הבאות:

- לכל $a, b \in S$ מתקיים $a+b \in S$
- קיים איבר $z \in S$ כך שלכל $a \in S$ $z+a = a = a+z$
- לכל $a \in S$ קיים איבר $q \in S$ כך ש- $a+q = z = q+a$
(אם q כזה מסומנים $-a$ והוא נקרא הנגדי של a וניתן לחשבו שנוא יחיד)
- לכל $a, b, c \in S$ מתקיים $(a+b)+c = a+(b+c)$

לעומת $(+, \mathbb{Z})$ חבורה, $(+, \mathbb{Z}_n)$ חבורה,
$(\text{מטריצות}, M_n(F))$ חבורה לוגי.

חבורה חלש היא קבוצה S עם פעולה $*$ כך שמתקיימות
התכונות הבאות במקום הקודמות:

- לכל $a, b \in S$ מתקיים $a*b \in S$
- קיים איבר $n \in S$ כך שלכל $a \in S$ $n*a = a = a*n$
- לכל $a \in S$ קיים איבר $q \in S$ כך ש- $a*q = n = q*a$. אם q
מסומנים כ- a^{-1} וקוראים לו האיבר ההופכי של a (ניתן להוכיח שהוא יחיד)
- לכל $a, b, c \in S$ מתקיים $(a*b)*c = a*(b*c)$

שדה הוא הקבוצה S שעליה מוגדרות $+$ ו- $*$ כך ש-
- $(+,S)$ חבורה חילופית,
- $(*,S\setminus\{0\})$ חבורה כפל בה
- הפעולות $+$ ו- $*$ הן קומוטטיביות
- דיסטריבוטיביות: לכל $a,b,c\in S$ $a*(b+c)=a*b+a*c$

חבורת מחלקות שאריות

לכל $n\in\mathbb{N}$ נגדיר $\mathbb{Z}_n=(\{0,1,\dots,n-1\},+_n)$
↑ מספר שלמים n

- לאיזו חבורה חילופית מכר אפשר!
- מהיר שיש סיבוכיות - לאיך הגדרנו $+_n$
- שימו לב האפס הוא 0: $a+_n 0=a=0+_n a$
- לכל $a\in\mathbb{Z}_n$ מתקיים $a+_n(n-a)=0=(n-a)+_n a$
- הסוריאקטיביות ברורה.

אי אפשר לדבר כאן על שדה כפל. למשל ב- $\mathbb{Z}_{10}$ עם כפל אין הופכי. למשל ל- 2 אין הופכי. לכל $a\in\mathbb{Z}_{10}$ $a\cdot 2$ הוא מספר זוגי, כלומר $a\cdot 2\neq 1 \pmod{10}$ כי $\Leftarrow$ אין a כך ש- $a\cdot 2\equiv 1 \pmod{10}$.

מה שכן אפשר להגדיר זו החבורה הכפלית

$$\mathbb{Z}_n^*=(\{x\in\mathbb{Z}_n : \gcd(x,n)=1\},*_n)$$

עכשיו קשה לראות שכן חבורה. רק נראה שיש הופכי:
נניח ש- $a\in\mathbb{Z}_n$. אז $\gcd(a,n)=1$ $\Leftarrow$ קיים $x,y\in\mathbb{Z}$ כך ש- $1=ny+ax$ $\Leftarrow$ x הוא ההופכי של a. שאר הוא מי ש- n? זו אחרת עכשיו יכול להיות שלכל x אפשר למצוא y כך ש- $1=ny+ax$ (כי אולי יותר מתחלק ב- n אחד אולי שאנו מוצאים y).
אם n ראשוני אז $\mathbb{Z}_n^*=\{1,\dots,n-1\}$ וזו חבורה כפלית. (למעשה $\mathbb{Z}_p$ שדה לכל p ראשוני).

תזכורת (או אקסיומות) של חבורה שבה פעולה היא קומוטטיבית. כלומר לכל $a,b \in S$ מתקיים $a+b=b+a$ (או $a*b=b*a$ אם החבורה היא כפלית).

חבורה ציקלית היא חבורה שיש לה יוצר. אם S סופית אז $\exists g \in S : \{g^i : i \geq 0\} = S$. כאשר S אינסופית אז $\{g^i : i \geq 0\} \subsetneq S$. כאשר $S = \langle g \rangle$ (זה אומר "שחזקות של איבר מסוים שייכות ל- i כלשהו").

לדוגמא, $\mathbb{Z}_{10} = \langle 7 \rangle$ שים לב שהחזקות של 7 הן רק כיוון אחד של הסדרה (יכול להיות שלכל אחד מהאיברים יש חזקות אחרות שיוצרות אותו).

<u>משפט פרמה הקטן:</u> אם p ראשוני אז לכל a מתקיים $a^{p-1} \equiv 1 \pmod p$

נשים ✔ שמתקיים $a^{p-2} \cdot a = a^{p-1} \equiv 1 \pmod p$. לכן a^{p-2} הוא ההופכי של a ב- $\mathbb{Z}_p^*$. משמעות לחשב את a^{p-2} בזמן $O(\log p)$ פעולות אז זה נכון. אפשר לא לחשב את ההופכי בעזרת אלגוריתם אוקלידס המורחב.

אומר a ייקרא יותרת אם קיים b כך ש- $b^2 = a \pmod p$ נשים לב שאם p ראשוני אז אם $p-b$ שורש.

לדוגמא ב- $\mathbb{Z}_{11}^*$:

$$1^2 = 10^2 = 1$$
$$2^2 = 9^2 = 4$$
$$3^2 = 8^2 = 9$$
$$4^2 = 7^2 = 5$$
$$5^2 = 6^2 = 3$$

אז יודע שרק 1,4,9,5,3 שהם יותרות! כלא מעניין!

נניח ש- $\mathbb{Z}_p^* = \langle g \rangle$ אז $g^x \equiv g^y \pmod p$ אומר $x \equiv y \pmod{p-1}$.

בדיקת ראשוניות

לאחרונה אספר בראשונים כאן לראשונה. אבל אתה רוצה לאחרונה
אספרים ראשוניים תדשים מאוחר שלך מיוחד.
אם n שלם אפשר לאשר בשיטה שבפרק הזה ראשוניות
בזמן $O(\sqrt{n})$ - פשוט נעבור את התוקה בכל המספרים
שקטנים מ- $\sqrt{n}$. אבל זה לא אשכרה כי עדיין n
מאורך 1000 ביטים ייקח 2^{500} פעולות.

נאמן ב- $\pi(n)$ את מספר המספרים הראשוניים עד n.
הלכתי ש- $\pi(n) \approx \frac{n}{\ln n}$ $\Leftarrow$ מצרך $n^{\frac{1}{2}}$ מספרים עד
n יהיו ראשוניים.
אז סיכוי קטן מאוד בין 1000 ביטים שיהיה ראשוני
הוא ± 300 (בערך).

האלגוריתם ההסתברותי של מילר ורבין

מותרים n שקרנות או-לוגי. הבדיקה של n מתבצעת כך:

בשלבים:
1. מגרילים a שקרני בין 1 ל- n $\quad n > a > 1$
2. מחשבים את $a^{n-1} \bmod n$
3. מגלים שהוא $\neq 1$ אז מצוי n בטח ראשוני
4. מגלים התישנות של (3) אם ראינו $\pm 1 \neq r$
$1 = r^2$. אם יש כזה אז n בטח ראשוני.

מסתבר שהסתברות ש- n פריק ועובר מבחן אחד הוא קטן
ממש $\frac{1}{2}$. אז אם תגרילו לי t פעמים הסתברות
שנלשה t איטרציות ולעדין לא פסלנו את n היא $\frac{1}{2^t}$
וזה יורד ממש מהר עם t. ב"כ $t = 100$.

נאה כמו ייקת לנו למצוא ראשוני אקראי בערך 100 ביט. לעומת
$t=100$ זה בערך 100·100 אבל צריך להבדיל מערך 1000
כפולים בלא פלא. אפשר בערך ב- 90 מיליון פעולות כפל.

נשים ♥ שמפ"ב נתפוס את הראשיים מאוד-מהר- נקבל את
הפקטור שלנו $t=100$ ואפסי צמאבא ומרקם ה- 1000
כפלים כפלים 200-100 סמוכים של n. נקטע כאשון.
במרכז - מילוון פרוטוטיפי שמדוע מספר ראשוני של
1000 מוכ...

(34) 28.11.07
הכפנה

קריפטוגרפיה מפתח ציבורי
Public Key Cryptography

פריצת הדרך הראשונה היתה ב- 1976 ע"י
Diffie - Hellman פרסמו מאמר ראשון.
אותו הם פרסמו שיטת החלפת מפתח DH.
באותה שנה שיטת Merkle פרסם שיטה הכפנה שלא
כמה חודשים לאחר מכן.
ב- 1977 פורסמה שיטת ההצפנה RSA
ב- 1980 Lamport פירסם שיטת חתימה
באותה שנה פירסם El-Gamal אלגוריתם הצפנה
וחתימה. גם אפן, מאז פורסמו עשרות אלגוריתמים.
הללו הוריאנטים שמשמשים בדרך היום בעולם מאוד
קטן מהאלגוריתמים שפותחו בפועל. כ-90% שהם עדיין
שניים על גובה המפתחות, אלא שם, אמנם עדיין
שניים מהותיים (עדיין מעלה שפירסמו לפחות)

הגדרה

לכל משתמש A יש שני מפתחות
- $Prv(A)$ - פרטי של A.
- $Pub(A)$ - ציבורי של כולם.

הצפנה - פונקציות E, D כך ש-
$C = E(M, Pub(A))$ כל אחד יכול להצפין
$M = D(C, Prv(A))$ רק A יכול לפענח

חתימה - פונקציות S, V כך ש-
$T = S(M, Prv(A))$ רק A יכול לחתום
$V(M, T, Pub(A)) \in \{0,1\}$ כל אחד יכול לוודא

← חתימה את הודעה
כדי לוודא חתימה

המוצג יפה, שהוא לפי הבנתי מאפיין הצפנה טובה.
למשל $E = id$ מקיים את הדרישה, אולם אין הוא מתאים כלל
לשום טוב. אלא נדרשה דרישות של דרישות הביטחון.

דרישות

- E, D ; S, V הן פונקציות דטרמיניסטיות
- E, D, S, V הן פונקציות הסתברותיות המקבלות מפתח מתאים.
- בהינתן מפתח מתאים, קשה מאוד למצוא את M, עבור (π, M) ...
- מהינתן $Pub(A)$ קשה למצוא את $Priv(A)$ (גם אם ראינו הדרכה והדגמה)

שיטת ההצפנה של Merkle

מה שאנחנו מחפשים זה פונקציות חד כיווניות - קל לחשב אותן אך קשה מאוד. (one-way trap-door functions).
באותן שנים מתמטיקאים התעניינו איך מוצאים פונקציות כאלה.

בעיית ה-knapsack - יש שקית N עצמים $1, \dots, N$. לכל עצם i יש משקל k_i.
מהינתן עצמים קל מאוד לחשב את המשקל הכולל של העצמים. אולם אם אומרים לנו שמשקל יש עצמים מתוך כולם קשה מאוד לדעת מהם העצמים שבפנים.
כלומר $\sum x_i k_i = C$ כאשר $x_i \in \{0, 1\}$.
מהינתן $\vec{X}$ קל לחשב את C, אולם מהינתן C קשה לחשב את $\vec{X}$.

איך נוכיח מכאן שגם הבעיה הכללית NP-שלמה?

נתון: N זרמים M_i, $i=1,2,\ldots$ שמשקלות יחידות k_i

הנדרש: $x_1, \ldots, x_N \in M$ $x_i = \pm 1$

הבעיה: $\sum_{i=1}^{N} k_i x_i = C$

יש פה בעיה! לה אין תחת! אם $R=(1,2,3,4)$ ו- $C=5$ יש שני פתרונות - $(0,1,1,0)$ ו- $(1,0,0,0)$.

צריך לוודא שהבעיה תהיה חד-ערכית. שאר Merkle ו- Hellman שלקח עמוד הקורס $\vec{k}$ שהוא super-increasing - כלומר $k_i > \sum_{j=1}^{i-1} k_j$

איך נוכיח שהבעיה $\sum_{i=1}^{N} k_i x_i = C$ קשה? נניח שיש אלגוריתם A שמוצא את x_i המקיימים את המשוואה שלפניך. נראה כיצד ניתן לפתור בעזרת A את הבעיה המקורית (כי אותה אנו יודעים לפתור ב- C מעלית הקומות). את תוצרים על התהליך עם $k_i - C = C$ (זו סיבוכיות שלנו!)

את הרעיון בה תוך אותם אנו עוד אחרי שהוא סודי. אין בה שום דבר סודי.

נבחר $m > \sum_{i=1}^{n} k_i$ ונמצא w כך ש- m. נחשב את $w \bmod m$ w^{-1}. נגדיר

$$\vec{k}^* = (wk_1 \pmod m, \ldots, wk_N \pmod m)$$

ונאמר שהבעיה הקשורה ל- $\vec{k}^*$ ולמשתנה הפתוח הוא $(\vec{k}^*, w^{-1}, m, w)$.

$\vec{k}^*$ גם היא super-increasing לפי שיש את m.

הבעיה: $M = x_1 x_2 \ldots x_N$

$C^* = \sum_{i=1}^{N} k_i^* x_i$ כל אחד יכול לחשב את $\vec{k}^*$ בעזרת

פיצוח: $C = C^* \cdot w^{-1} \bmod m = \sum_{i=1}^{n} k_i^* x_i w^{-1} w \bmod m = \sum_{i=1}^{n} k_i x_i \bmod m = \sum_{i=1}^{n} k_i x_i$

$\sum_{i=1}^{n} k_i x_i < m$ כי

אנו יודעים שקיים C שבעזרתו אפשר ליצור $\vec{x}$ מ-superincreasing

אם k באורך 1000 ביט הסכום שלו יכול להיות 2^{1000} לכן m יהיה גדול. זה קשה מאוד לעשות ניחוש מיושם מוצלח

כאן מוצגת שיטה של Merkle שפיתח את השיטה זה, שעזר שמצא את הדפוס ש"ש שנוצר בהצפנה שיטת הקשה!

זוכרים הנוא שלקחים מעין אלגוריתם שמושמים שהן קשות ומכניסים להן אלמנט של סוד כדי שיהיה קל לפענח.

חתימה חד-פעמית (Lamport)

אותנו נוגעים ל"ד מעניין שמאמצותיו נעם לחתום באופן חד פעמי שיצר Lamport שאפשר לבדוק ממנו: A ומציגו וקטור מילות באורך N $(y_N, x_N), \ldots, (y_1, x_1)$ ישנו y_i, x_i שמקיימים לפחות אחת מהפרטים הפרטי. המפתח הציבורי הוא $(H(y_1), H(x_N)), \ldots, (H(y_1), H(x_1))$ ישנו H פונקציה ידועה.

בהינתן הודעה $M = b_1, b_2 \ldots b_N$ חתימה של A היא פרטים של $z_1, \ldots, z_N$ כאשר $z_i = \begin{cases} x_i & b_i = 0 \\ y_i & b_i = 1 \end{cases}$

כדי לוודא מול i: אם $b_i = 0$ בודקים $H(z_i) = H(x_i)$

ואם $b_i = 1$ בודקים $H(z_i) = H(y_i)$.

אנו מן הסתם יש פה כמה בעיות: קשה להפיק את M', קשה למדוד התנגשויות של H וכן הלאה.
אנו אומר M טובה קשה למצוא את זה. הוידוא הזה כאמור קל.
למה זה חד פעמי? אם שולחים הודעה שנית אז אותו מפתח אל זה יכול למשל עוד פעם מהצורה (y, k_i).
זה לגמרי כאמור כתוצאה אחרת בית לאורך ההודעה ופונקציות hash הן ומסות אינטרנט, ובניגוד לבלוק אחרי אחרון.

אל אם לא משתמשים בשיטה של Merkle מה לגבי ה- hash, אנו מה קורה???

ניתן משתמשים בשתי בעיות קשות מתורת המספרים
- לוגריתם דיסקרטי (H-D, El-Gamal, DSS)
- בעיית מספרים לארטיפים ראשוניים (RSA)
אלגוריתם בעיית RM- קשות אחת בין בעיות קשות למחצה.

<u>לוגריתם דיסקרטי</u>- בהינתן קבוצה ראשוני יחידה ובהינתן g יוצר של $\mathbb{Z}_p^*$ קל לחשב את $y = g^x \pmod p$ אבל קשה לחשב את $x = \log_{(g,p)} y$
נשים לב שמתקיים ה- $\pmod p$ אי אפשר לדעת אינפוש מינימלי! כאן x המפתח הפרטי! y המפתח הציבורי.

<u>פרוטוקול לאורמים רבים שניים</u>- בהינתן p וראשוניים שקל לחשב את g כך ש $g^p = N$ אבל בהינתן N כנ"ל קשה למדוד את p וגם g. כאן הסוד יהיה הפירוק של N והמפתח הציבורי יהיה N.

אלגוריתמי קריפטוגרפיה סימטרי

בשיעור הקרוב:

- RSA - הצפנה + חתימה + הנפקת מפתח } פיתוח ראשוני"ם
- DSS - חתימה
- El-Gamal הצפנה + חתימה } עוד קריפטו'
- H-D הנפקת מפתח

Diffie-Hellman Key Exchange - (DHKE)

שני אנשים, נניח שמם א' וב', רוצים להסכים על מפתח סודי משותף K. ברשותם הצפנים (אלגוריתם הצפנה ופענוח) כלשהם, אבל אין שום ערוץ סודי ביניהם. כל מה שיש להם זה ערוץ תקשורת (אלגוריתם) ציבורי שמהירותו z ו- M. אותם z או את y אותם M. אוסף הנתונים שם ציבורי הקריפטוגרפי שעוד יכול לסכל, כי מציאה קשה כאן סמיר שמתקיים יכן y מתשב את x אבל הפונקציה יפרץ את p, g, z ו- M. אמנם תחת ההנחה

הפונקציה g יכול שם $\mathbb{Z}_p^*$ שנוחה מכירים.
משהו תשתית האלגוריתם הזה שיש לו כאשהו, אפשר שיפרץ עם אותה כדי שיוכלו להסכים מידע או לתתום ביניהם תגובה.
יש שני אנשים. ניתן להם שום מפתח ראשוני, הם כאן מכירים לה אות לה וניתן לה שום מידע אחד על השני. הם רוצים לתקשר

נאפשר פרט: יהי כאן אנחנו שיש $\mathbb{Z}_p^* = \langle g \rangle$

A רוצה לדבר עם אפשר בהמרי

El-Gamal אלגוריתם הצפנה שמתבסס על בעיית הלוגריתם הדיסקרטי.

שמטרתו לאפשר שני אנשים שלא מכירים מראש יוכלו להסכים על מפתח סודי משותף. - DH מאומת - הצפנים משתתפים יוצרים משותף אחד על השני, כדי שיוכלו להשתמש בו לצורך הצפנה סימטרית של המידע ביניהם.

- קל מאוד - אם היקף מוכן, אבל יכולים הצדדים אבן לפחות גיא תוכן הפרוטוקול הזה הוא מאוד פשוט. יש לו שתי וריאציות:

שניים אפשריים מסוימים יחמור, רק שהם לא יוצרים עם מי ...

כן נשים ✓ ש- E בזמן שהוא מתחזה עבור אחרים התקשורת.

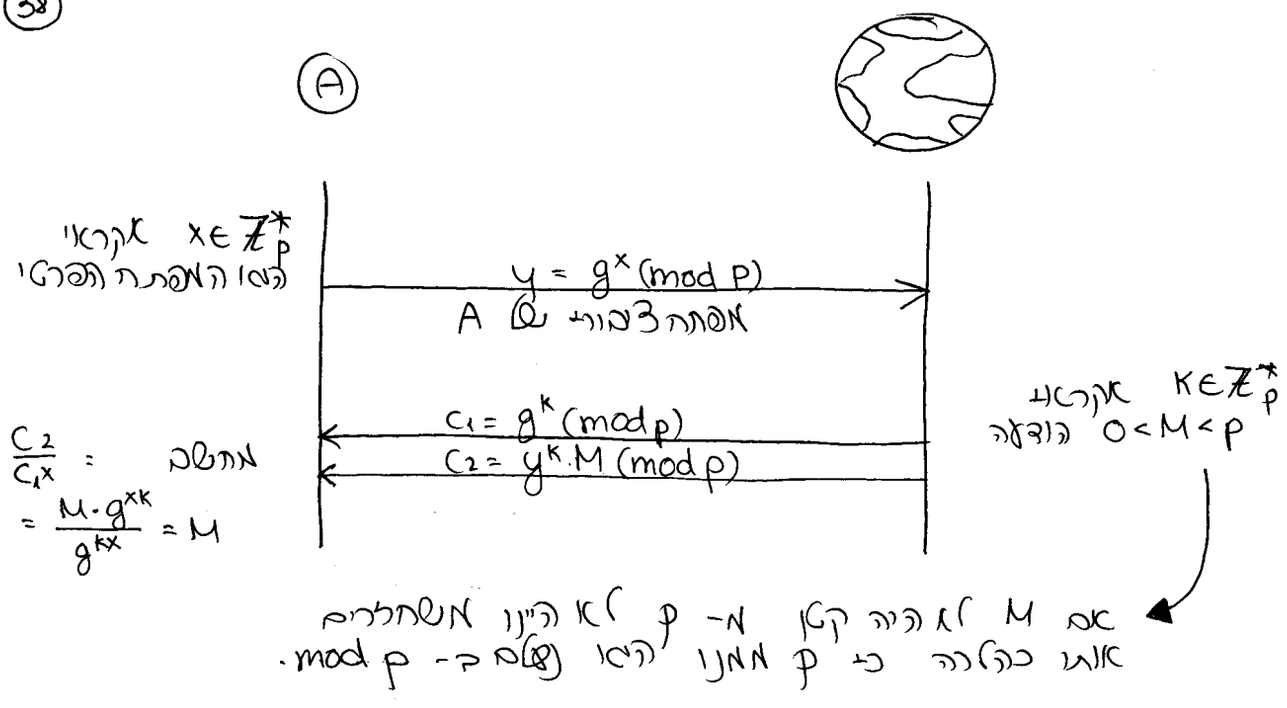
ו- B חושב שהוא מחובר עם A אבל בעצם הוא מחובר עם E.

ולעומת A חושב שהוא מחובר עם B אבל בעצם הוא מחובר עם E

Man in the Middle התקפת

פרוטוקולים כלה לא יעיל בפני שמשתמשים שלהם יכולים לוודא מיהו הצד השני.

פה הבעיה שאנחנו יש סוף נאלץ אחד לא ובין לשקר אותו. בפועל היא מרכיב יש לבעל סוד משותף. היא לה יותר מדי מבחינת עלויות כיון? ה עליון המופע נראה מוכר. אין A ו-B שיש מידע משותף ואותו יש מאוין

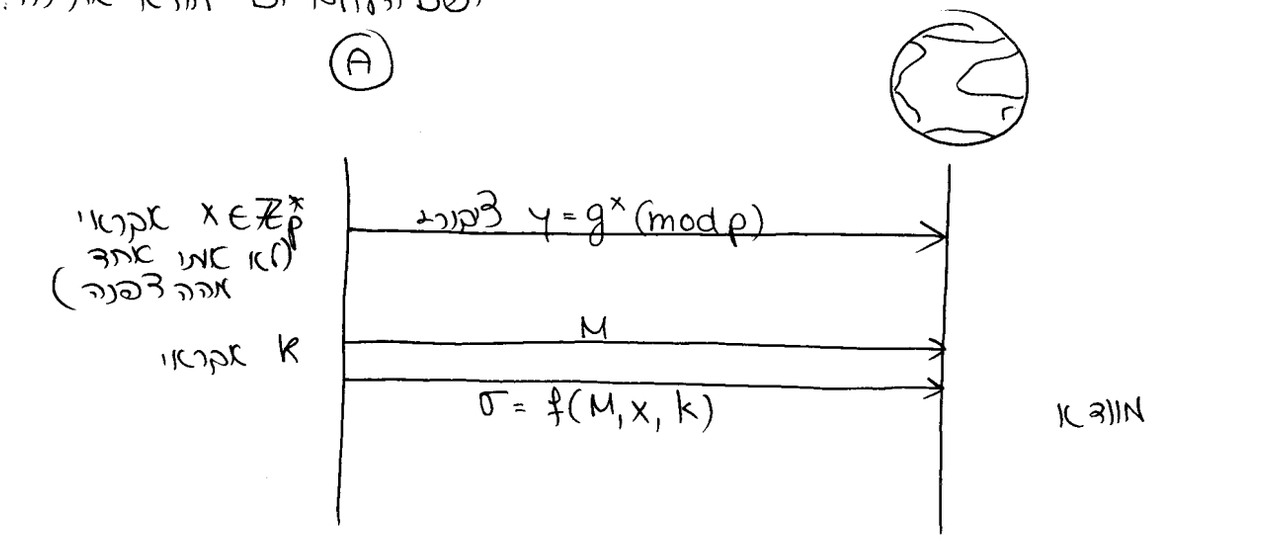


אותו כלגורה כי k מאלו הינו מספרים אם M לא היה קטן מ- p לא היינו מתקדמים

יש לו רווחים ציבורי: g יוצר של $\mathbb{Z}_p^*$. A בוחר (מתוכם) בדיקה

El-Gamal/DSS - תיאור

הוא מלכתחילה אינו מאפשר לבדוק את האותנטיות של הנקודה. אין שום אלגוריתם שמאפשר לבדוק את הנקודה. אין שום דרך לדעת שהנקודה שנשלחה אכן מקורה ב-A. אין סיכוי שמישהו אחר יכול היה לשלוח את הנקודה הזו. לכן הפתרון הוא לחתום על הנקודה (חתימה דיגיטלית) ולשלוח את החתימה יחד עם הנקודה. כך נוכל לוודא שהנקודה אכן הגיעה מ-A. אם c_1^x לא שווה ל- y^k אז הסיגנל אינו בדיקה.



מה קורה אם מישהו מתקפל? כל הפרוטוקולים מבוססים על הנחות. אחת ההנחות היא שהמפתח הפרטי נשאר סודי. מה קורה אם מישהו גילה את ה-k האקראי? כל אחד יכול לגלות את המפתח הפרטי שלנו! זה קורה הרבה בפועל. מאוד קשה לשמור סוד.

DSS - Digital Signature Standard - מאפשר חתימה וזהו חתימה, אין צורך לבצע פעולות הצפנה/פיענוח כלל. כי האתרים אינם לא רוצים שהלקוח יהיה מישהו אחר.

לדוגמא מסיבות אמו"מ. נניח שמישהו פיתח את הפלגיה של ההצפנה, ויש את כל ה- M וקיבל מפתחות סימטריים לשימוש בהתכתבויות בין מישהו לבין RSA-Labs פיתחו פרוטוקול שמאפשר לשני אנשים ליצור מפתח סימטרי ביניהם בלי שאף אחד אחר יוכל לדעת מה המפתח.

- p חייב להיות ראשוני (אפשר להוכיח שאם הוא לא ראשוני אז אי אפשר לפענח בכלל).
- אם $n > M^e$ אז בשום אופן אי אפשר להוציא שורש רגיל. במקרה כזה אפשר לפענח בקלות ע"י הוצאת שורש רגיל (אם ראינו התכתבות שם רואה הוצאת שורשים יכולים לגלות בקלות מה M היה).
- אם $M_1 \mapsto \sigma_1$ $M_2 \mapsto \sigma_2$ אז $M_1^i M_2^j \mapsto \sigma_1^i \cdot \sigma_2^j$

תכונות של RSA:

$$M = M \cdot 1 = M \cdot \alpha^{(1-b)(1-d)} = M(M^{p-1})^{(1-b)(1-d)} = M^{1+\phi(n)\alpha} = M^{ed} = C^d$$

שזה מסביר למה אפשר את ההצפנה ע"י הפעולה $\pmod n$:

★ ה- e וה-p של ההצפנה הם לא אותם e ו-p של ההתחלה.

← ניפוח:	$\sigma^e \equiv M \pmod n$	ע"י מי שיש לו גישה
← חתימה:	$M^p = \sigma \pmod n$	רק A יכול
← פיענוח:	$C^d = M \pmod n$	רק A מסוגל
← הצפנה:	$M^e = C \pmod n$	$n > M$ ע"י מי שיש לו גישה

(אפשר להשאיר את d, b ; $\phi(n)$)

- המפתח הפרטי הוא (p, n)
- המפתח הציבורי הוא (e, n)
- מחשב $d = e^{-1} \pmod{\phi(n)}$, כלומר $ed \equiv 1 \pmod{\phi(n)}$
- מגדיר e כך ש- $\phi(n)$
- מחשב $(1-b)(1-d) = \phi(n)$ מספר האספרים שלכל n $|\mathbb{Z}_n^*| = \phi(n)$
- מחשב $b \cdot d = n$
- מוחר b, d ראשוניים גדולים ואז קשורים.

A רוצה לבנות לעצמו מפתח פרטי וציבורי. מה הוא עושה?

RSA הוא משתמש בעיקר שיטת שיתוף מפתח לראשוניים.

משימות הקהילה (מלבד את אימות אנשים, פרוטוקולים וכו'.

אימות אנשים

מאז' מתשמ אפשר לחלק את שני תחומים: אנשים ומחשבים
ויש אבחנה סוגים של פרוטוקולים:

- אחד מול אחד
- אחד מול מחשב
- מחשב מול אחד
- מחשב מול מחשב

אך יש לחלק מטרות האימות - לא כאמור תלוי במערכת

- ביהוי - באופן פשוט לענות לשאלה מול מי אנחנו עובדים. לא רק ביהוי ראשוני. למשל את מוחים מול שאנה את מחליפים שלה שאה אולי יש לו את תאום. את לא ביהוי בלי אימות.
- אימות / ויזוא - זוכן אנחנו מעוניינים שאה שביהוי הוא באמת נכון. למשל סיסמה וכולה לאמת.
- הרשאות
- מזקה (log של הנשה)
- Auditing
- Monitoring

פרמטרים לבחירה מטיטת אימות / ביהוי

- כמה בטוח - האם האימות הזה ב-100% או רק 99.9%? ב
  לחפץ מהן ט וכמן)
- אותלקה (יש מערכות שאין בהן לחתלק ויש כאלה שכן
- שלות מערכת
- נוחות (למשל מקלדת ארגונית)
- כמה שינוי (למשל בקלדת מכאני אלפי שקלים) ולפעמים
  - כמן ביהוי / אוזאות - יש מערכות שבהן יהו אחסנת לנו לחתלת

שמירה - דרוש של תווים שמורכב מלעות / אותיות גדולות / מספרים
מהי סיסמה טיפוסית? מורה באנגלית - אותיות / מספרים
שמירה - סיסמאות / מידע אישי, PIN.
- קשה למתקפת לחם / לאדוג / שבלת } מהניח ה"נלחמי".
- קל למשתמש לכתוב } קשה להשיגה אנשים
של המשתמש). הפרמטרים שנבדקו:
משהו שהמשתמש יודע (- ההנחה היא שאפשר לשמור עליו בשלם
סיד קשה יותר! security תמיד יותר.
היום בדב לומדים על שלומים של פגיעה מהסוג הזה. כה מעשים
3) משהו שהמשתמש מחזיק (אפתח, כרטיס חכם, ג'יפ אמו"ם)
2) משהו שהמשתמש הינו (אזורים ביומטריים, פיזיומטריים)
1) משהו שהמשתמש יודע (סיסמה, PIN, מידע אישי...)

אימות אנשים יש כמה סוגים של איומים.

סיסמה אפשר לשנות אחת אלפיה - לא...
- שינוי / תפוגת מלהה - האם ניתן לחפש מלהה?
אותה הלידה לה עכו ממש אפשרי.
- הגבלת מלהה - למשל אם נתנו למישהו סיסמה אל יקלט
- הלכרת מלהה - למקלה שרוצים שמישהו ימצא את מקומי
לה קלט פחות נוח...
אם צריך לפלל את האלגוריתם שמחשב פונקציית HMAC
- קלות שימוש - מפיק: טביעת אלגוריתם פשוט למשתמש, אופן
ישנם פגיעים באוכלוסיה.
אנשים עלו נתונים שאינו כן מלמלת.
שגוי - המלמלת כן אנשים יכולים. לפעמים יש
אנשים נכונים. בכל אופן עלו נכונים עלו מלמלים.
תוכי - המלמלת כן אנשים נכונים אפילו לפעמים פלות
יש רמת פיקח תוכי ורמת פיקח שגוי.

מסיסמה קודמת, משהו קל לניחוש, משהו שרשמנו באיזה מקום
מקודם לה לא כדי להרחה אפשרויות, אפשר לעשות תיפוש
ממוקד על האפשרויות. סמיר שאתרי מספיק נתונים
כדי לפצח את הסיסמה.
בפועל הכלבי הזה למשל 100 מיליון איקריום. ועוד שאפשר
לנחתפק ב-69. מן הסתם כדי לעשות שאפשר לעבור.
מפוצר ב- 20-11 תווי ASCII ויצואליים. כדי להרחה.
אפשר להמיר את כל המשהו שעיתן לחזיה אפילו יותר קל לעבודה
אומלץ אל צריך 40-50 תווים. או אפשר 30-50 תווים
שמייצרים משפט - מילים באורך משמעותי עם קשר תקבירי
למשל "הסוס הלורוד רקד סמבה עם התמתה הבתולה".
נפעור בין הכלל ולדות הזה כי יפעל. אבל מה זאת, סיסמה
לה נורא נוח...

מורכבות של אלגוריתם פיצוחים

- גודל משתנים
- מספר רב של שרתים
- צורות משתמשים במערכות שונות

} יש אנשים עם סיסמאות חלשות.
גם הדבר הזה ביטוי

הסיסמה (או גירסת שלה) נשמרת במחשב (בדיסק השני), אחרת
איך נשתמש בה? יש עוד מיני אפשרויות לשמירה:

- עלגו הגלפה - אישה באופן גלוי
- עם הגלפה אישה (הרשאות גישה) באופן עקיף
- עם הגלפה אישה באופן מוצפן - למשל קובץ מוצפן
(אבל איפה הסיסמה?)
- עם הגלפה אישה ע"י hash (בהמשך דברים מפורטים)

אין צורך להגן עם ה hash ב
ישקפץ סיסמאות ששמר משתמש וה- hash של
הסיסמאות שלהם. הקבץ לכן ביד לפיות מוכפן או מלח.
ב פעם שמשתמש מסיסמה, המלח מתשב את ה- hash
ומוודא זהה!
גמגם עם בקבב הלה אפשר לעשות online dictionar attack
עניין שמתקיל יש הרשאה קריאה לקבץ הצפה. הוא ורק
למוצ דם מלאו סיסמאות וה-hash שלהם מוכנים מראש.
ואז מהמה שניות אפשר פשוט לחפש בהטואה.
אז אפשר להגבות את הניש יח לקריאה. ב- Unix
הוסיפו מנגנון salt. הקבצים נשמרים כך:

A : — : $salt_A$: $h(pw_A \| salt_A)$
⋮

salt זה מספר אקראי שמוצמד ל- A ל פעם שהוא
פותח חשבון או משנה סיסמה. אז אם יש מליון סיסמאות
פשפלכיות שם את אותו הן יש מפרים שונים שכל
ה- salt. אם ה salt ארוך מה מאוד יקשה
על התיפוש.
אבל זה לא מונע התקפה של offline. כי אז יש יותר
מן ללנות אותן אם ה- salt-ים ה מתאימים.

סיכום של סיסמאות

- נוח מאוד למשתמש
- קלות נשובה בגב
- אפשרי שהו יבור סיסמה
- או אפשר שהו תבור סיסמה
- שינוי והחלפה אפשרים
- הפונק יש התוכי, ובמה השיש הוא טוב (מהמה שאין התנגשויות)
- התמטיות לא משהו
- אתקפה נוחה

משהו שהמשתמש הניח

- אמינות אובייקט
- אוטומטיות של כל היחידות
- מבוקרת קלט
- זיהוי IRIS
- זיהוי Retina
- זיהוי פנים
- DNA

זה לגמרי משהו שקרה בעבר. זה משהו שנוגע עם המשתמש ומאפיין אותו.
זה מיצג כי אישי, כי אף אחד אין עוד אותו קלף או אותה אובייקט.
ובזה נותן מאוד למשתמש.
למרות שלפה יש חסרונות - אם שם ההקלטה קלף.

אובייקט

המתקדמת מרחיק שקשה למדוגמה שתי אובייקטים שונות. אפשר לתאומים מאותה ביצית יש חתימה אובייקט שונה (למשל היחל מתאלפים בשתיה מי השפיר, אל בה הפליק סמוסה כי)
צריך להבדיל תהליך רישום - נמצים למחשב רואה תמונה של האובייקט. המחשב שומר את האינפורמציה הזרה שה מאפיין שמצדיה יש המשתמש הרלוונטי.
אבל יש הטוענים בעניין זה: לעומק על האובייקט, לעומת מפתחי, פיזיה. אלה מציות בשימוש.
אבל יש יתרה בעניין זה אפשר לקחת תמונה של טביעת האובייקט מרוס קפה או משהו ואז לנסות לפרוץ את זה לפעמי.
אל משהו יותר מתוחכם כה תמונה תלת ממדית של האובייקט.
יש סנסורים שמזהה מעיכות.
זוכר מתחבא כה מפיקת חיות של האובייקט - טמפרטורה, עוד
דם וכו' - שבדוק שהיו אובייקט אמיתית אל או אפשר לבנות

מדקים אחרי אלפעות, מתחקים בין אלבעות,
צורה של אלפעות.
כמובן גם לגבי מושם.

מדידת קנזית IRIS

גם משהו מאוד "חודי לאנשים.

למשל, מה עושים מרספונס? מכניסים כרטיס שמגדה אותך.
בדיק לחקיס MIT ובמן הלה אם מלמדים את הדין
ומדקים. גן מדידה משופר והיש מאוד יקרה.

מדידת RETINA

נימי הדם מאוד מאפיינים אנשים ואפשר לדעת אותם עם הדין
אינפרא - אדומה.

סיכום

מאפיינים של טיבוי ביומטרי:

- אי אפשר לזייף
- אי אפשר לשכוח
- נוחות - יש גם וגם
- אחידה - בדיק לחתכן את המרשים
- עלות יחד גבוהה מסמאה
- שינוי - אפשר פ כמה שנים לחדשה מחדש
- אין בדיק לחתכיר

אימות זהויות

משהו שהמשתמש מחליק

- תלוי מהיה (או מהידוע שהוא יודע לפעמים תלוי מהיה)
- כרטיס נייר / פלסטיק
- מאפיין ביומטרי

כרטיס מגנטי

- מפתח פיסי / מכני (בעיקרון שומפתח נמצא אך ורק אצל המשתמש)

הבעיה בציור כאן היא שהסיסמה לא מוגנת. אפשר להגיע לתקשורת וכך הסיסמה לא אבד יותר.
מה אפשר לעשות? - אפשר לתת למשתמש פרק זמן לשינוי סיסמאות - כך פעמים והמשתמש מקליד פעם סיסמה אחרת - כמעין אתגר בשרת בדיוק לדעת רשימה של הסיסמאות הללו.
בשהסיסמאות מאוחרות פעמים יכול רשימה מפשר.
בשביל להשתמש בכך המשתמש יכול לשנות מספר סיפרת של סיסמה ואת הסיסמה עצמה.

הבעיות?

- את הכרטיס נאבד לו מעיה גדולה.
- פעמים הרבה מאוד ביכרון בשרת
  שרות ואז להתמר בשרת.
  להתנגדות לשפת לקלט את הסיסמה, להיות שיון
- אפשר לעשות התקפת man-in-the-middle. אפשר
- כלי (ביכרון מעמד)
- אין בעיה הגיונית כי כל מה פעמי
- סיסמאות מקלות כי זא פעמים לכתוב אותן

הכוונה?

זאת הסיסמה עצמה.
בשביל להשתמש בכך המשתמש יכול לשנות מספר סיפרת של סיסמה
כשהסיסמאות מאוחרות פעמים יכול רשימה מפשר.
כמעין אתגר בשרת פעמים להיות רשימה של הסיסמאות הללו.
סיסמאות - זה פעמים והמשתמש מקליד פעם סיסמה אחרת -
מה אפשר לעשות? - אפשר לתת למשתמש פרק זמן לשינוי
לתקשורת וכך הסיסמה לא אבד יותר.
הבעיה בציור כאן היא שהסיסמה לא מוגנת. אפשר להגיע

כרטיס סיסמאות

- מפתח פיסי / מכני (בעיקרון שומפתח נמצא אך ורק אצל המשתמש)
- מאפיין ביומטרי
- כרטיס נייר / פלסטיק
- תלוי מהיה (או מהידוע שהוא יודע לפעמים תלוי מהיה)

משהו שהמשתמש מחליק

- יותר יקר מבחינה כלכלית.
- אין תמיד אפשרות
- יותר (אפשר לקבל את התושבת ע"י שמירת נקודת ביניים)
  גם אפשר גם לחפש אותו מתדירות נמוכה
- כל ש- N יגדל יותר ההתאמה של סיסמה לנקודת יותר

הבעיות:

התכונה של פונקציית hash: קל יחסית אבל קשה מאוד אחורה.
יש סיסמה ידועה לשימוש. את כל המיס כבר מספיק שימוש
ב- N סיסמאות שונות מה פעמיות. ההסתה היא יכולה בפועל
לחשוב במיס $N \leftarrow N-1$ ולשמור $X_N \leftarrow X_{N-1}$ וכך
X_N את ע"י מפיקה פשוטה היכו אוחזא שלה שון.
את שמו A ואת $X_{N+1} = h_{N-1}(x_0)$. לשרת יש גם שכו השם
hash בדרך אחורה. כפי להפחות המשתמש שלח לשרת
מכרסים של הנקות שמורים x_0 ו- N, ומכא יודע לחשב פונקציית
$X_{N+1}, \ldots, X_1, x_0$ כאשר $X_{i+1} = h(x_i)$ אחרי פונקציית hash.
מוחרים x_0 אקראי (ספל ביט) ומתחבים שלשרת סיסמאות
לפ משתמש A שולח את הדבר המאוחסן

מיכים סיסמאות - key/s

כי אינטרס של סיסמאות.

- אם מרגים לשרת ואנחנים את קבלת הסיסמאות
  כן מעניין אחרת. (אנחנם כן כי אפשר להתאבר לפ hash)
  מכים הסיסמאות ורק להיות וקבוצי או מאוחשת. אבל בפ אושן
  אין לו שום רכיב לוגשבי. הכל יכן לפעמי פברים. לא ירק
  להיות כן מאוד.

- אפשר לעשות התקפת man-in-the-middle
- הסיסמאות גלויות
- הגיבוי הוא בדרך אחר
- השרת אתגיק מידע לא רגיש- לא יקרה כלום אם מישהו יגנוב את סוף השעה.

אז אפשר לעשות? הנה פרוטוקול אפשרי: ChallengeResponse
אף פה לא בטוחים להתמודד עם התקפת man-in-the-middle

כלומר, שהשרת "בודק מאחר את A ומשהו שמתקבל מן הצד, שהשרת "בודק שהוא מאחר את A ומשהו שמתקבל

את יודעים:
- אפשר סיסמה
- כמו
- אפשר סיסמה

יכולה להיות מחרוזת אקראית.
או רשימת הבקשות ... ש-S מנהל
התכונה של ש הבאתים הבאה היא שהם

R/C סיסמה

פונקציית חד-כיוונית
k סוד משותף בין A
ל-S

תכונות:
- הקלט יכול להיות (אפשר אותה אות פעמיים)
- אפשר סיסמאות ארוכות מאוד
- לא צריך סכנות בין השרת לקליינט. (מעין ההנחה שהשרת לא יודע את הסיסמה)

C/R מתפס למן

אפשר לעשות את אותו הדבר כך שהאתר יהיה הלמן בהנחה. לא כאן יש שתי אופציות: או שהלקוח שותל שרת או שהשרת שולח ללקוח. תיאורטית, אם הסביבה מושלמת אין לנו חיסרון בשתי האפשרויות. אבל כאן הסביבה לא מושלמת ואנחנו יכולים לזהות את man-in-the-middle. אבל אז אותו התוקף יוכל לשנות את תוכן ההמן לכמה שיותר. אבל אנחנו נשים לב שהמפתח הציבורי שלו/מיכרו שינוי לא יגרום לתקשורת. אנחנו נוכיח שפתרון זה יעבוד מעולה נגד התקפת man-in-the-middle. אבל לא נוכל להתמודד עם התקפת man-in-the-middle. אבל היא לא יכולה לעשות כלום נגד ההמן כמה שהיא תנסה. אנחנו נשים לב שאותו המפתח ציבורי שלו/מיכרו שינוי לא יגרום לתקשורת. אנחנו נוכיח שפתרון זה יעבוד מעולה נגד התקפת man-in-the-middle. אבל לא נוכל להתמודד עם התקפת man-in-the-middle. אבל היא לא יכולה לעשות כלום נגד ההמן כמה שהיא תנסה.

נקודה חשובה: ב הפרוטוקולים אנחנו שמנו לב שאפשר אי אפשר לדעת אם מי שאפשר אותם רק ב שלב האימות. אחרי זה אם נגלה אם הלקוח התנתק ואישהו אחר התחבר במקומו. הפרוטוקול רק יגלה ש-A פיבר אותו בשניות האחרונות אבל אישהו אחר.

C/R אמצע אפשר אקראי

כאן רק השרת יבין לשלוח אפשר לאשתמש. אחרת, אין פרק לפקח שהאפשר היה לא מאות אקראיות ולא השתמשו בו עדיין. השרת סומך רק על עצמו. הוא יכול את האתר. הבעיה כאן היא שלה יכול להיות הרבה אפשרים אקראיים. כשהיינו אלפי אפשרים יכול להיות שהלקוח אבל לה פותח את ההתקפה man-in-the-middle סמוך-אקראיות אבל לה יכולה להתקפה בכלל.

איך עובד ה- OTP שלנו?

פתח הוא סיכהן כאן של כפתור.
כי סיסמה חד-פעמית שמשתנה ע"י הזמן.

יש כמה אופציות:

- PIN משמש לפתיחת המרשיר (אז ה-PIN שמור רק במרשיר)
- PIN משמש כדיק מהמסיס לסיסמה (אז ה-PIN נדיק להיות בשרת)

יש מעיה: יכל להיות שסנכרון הזמן נהרס בין השרת לכרטיס. השרת יכול לנחש את הזמן הנכון של הכרטיס ואז לה צריך לעבוד. אם ובמציאות היא כאוטית אי אפשר לנחש את זה.
במקום כך שהמערכת תאפשר יותר ויכו רחש להתקפות.

את השרת מקבל נהנה מסיסמאות שאיות ל-A את כתובת שמשמו לא מספר. אבל לא נאמם לגור בסיכהן אבל אפשר לעשות מעקב אחרי ע"י הכרטיסים.

פרוטוקולים לאומתות

④ פרוטוקול אפס-ידע

③ מבוסס, אפשרי אוטומטי

② מבוסס, אפשרי סימטרי

① מבוסס סיסמה

} רב אחד יש מאפיינים משלו ומתבטאן מלכתי בדיק לתשובה מה מתאים.

פרוטוקולים מבוססי סיסמה

(סיסמה (מניחים ואפשרי) לא משתנה שקל לנחש אותו וכו כמובן מעיד ...)

יתרונות:
- פשטות /קל
- אין צורך בתוכנה בצד המשתמש

חסרונות:
- מאפשרן יכול לאלץ סיסמה (התקפה replay)
- נתן לנחש / לגלות סיסמאות
- יש מידע רגיש בשרת

בפרוטוקולים כאלה יש שימוש ב יום - לאו-אין, שמונע תוקף.

פרוטוקולים מבוססי אמת סימטרי

נניח כי את כולם יודעים שיש לכולם 256 ביטים אקראיים.

הבעיה בשני שלבים הוא שהיה ניתן לעשות replay ל-K_{AS} בלי שנדע שהיה.

הפתרון יכול להיות אחד מהשניים:

- מספר אקראי
- זמן
- מונה

$$\text{nonce} = \text{משהו שנוצר על ידינו}$$

הפונקציה f יכולה להיות:

- hash – מפה חד-כיוונית, קלה לחישוב ודורשת שיעור
- mac – קלה לחישוב ודורשת שיעור
- הצפנה – הפיכה

נניח אין וריאציה של הפרוטוקול הבא:

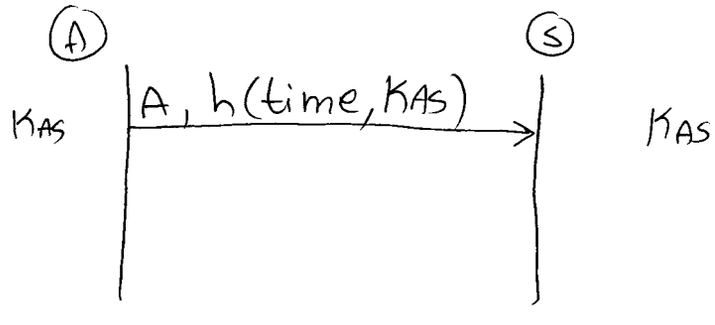
כאן כולנו S מחשבים את C מרגיע השמות K_{AS} ואז A שולח לפונה אותו ולהגיד את C. אחרי שעשינו היתוך שיודעים את K_{AS} את זה מספיק. הבעיה כאן היא ש C היה מן ת"ש – למשל תמיד כמן או מונה סדרתי. אלא C אינו להיות אקראי. כאן זה שהוא איזה לכמה את C. אומר יש פרוטוקולים של "שיר ושמח".

אחרת, עשוי לשים לב שבקשה הגיעה ממי שאינו בעלים של הכתובת. לה במידה שלא - יש איזו מנגנון - אולי header נוסף, שמור של המשתמשים אותו כלה. אל אם בפיזיקה מתקפת משתמש, בעזרת המנגנון הנכון אל כתובת שמי שלנו אותה היא מאמת השפה.

יתרונות:
- מניעה שהכוונה כי האחר הוא מאיר אדם
- אין מקום של תוכן המפתח כי רוצה אותו אקראי.

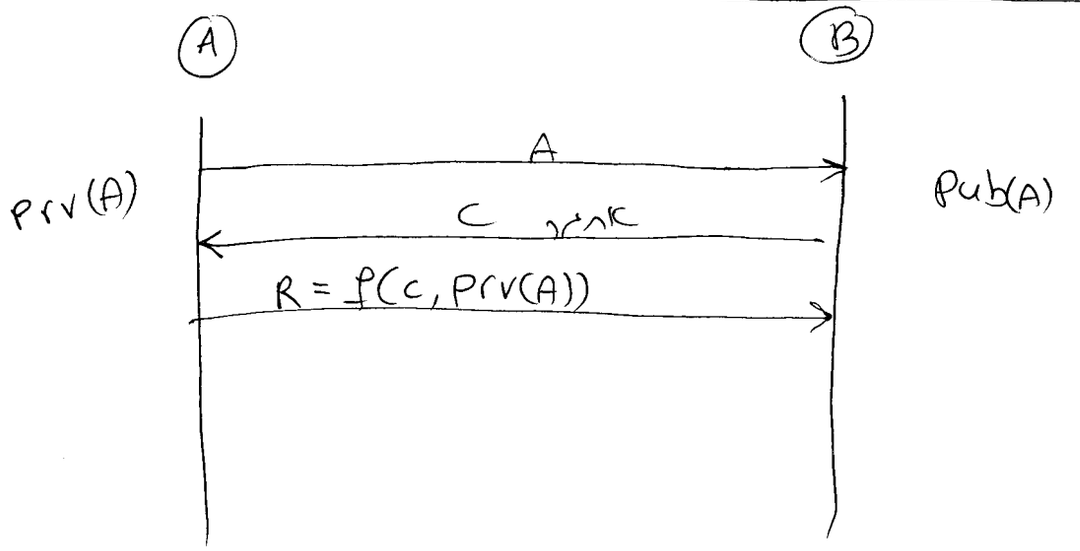
חסרונות:
- קוד שומרה את הלקוח (עמודים לתשב פעולה)
- משתמש מיוחד משרה
- זמנים יש מידע רגיש שלפ הלקוח אפשר הנה מוסר מוקם שמלוף. לשרת יש הרבה יותר מידע.
- נמנעים השיש הבנות/פיענוחים מהפרוטוקול - את אלו אפשר אפילו לעשר הדקפה chosen plain text

לפחות, יש פרוטוקול אימות יותר הנדרה אותה:



ואין לעשה יש אחר יחסי. בעזה הכה חלון הנמן ביני ובין מספיק יחסי לחישוב הפונקציה ולתקשורת. ומלפנים הנמן הרבה אותנו ישים ל התקפה replay! אך אם מתקים h הייה פונקציית הצפנה את S ית) לפעמ יחד והנמן ולבדוק אם הנכון מתאים לנו או לא.

פרוטוקולים עם אפשרות אוטומטית



לפי הגדרה הסכמטי, f היא או תמיד או פעמים. צריך לה יהיה שמשתמש יכול לפתוח מאותן תמיד. שהרי משתמש יכול לשלוח לנו hash של אובייקט ואותנו נחתום עליו. אפשרות לתקוף פעמים- משתמש ישלח לנו משהו שהוצפן איתשהו ואותנו נפענח לו.

קשה לחשב מטרה לשתמש באפשרות שניה. יהיה לנו אפשרות שהשרת היתודה שלו בדיוק הוא אוטומט (אולי נצפה על כך). אבל אם f בדיוק להיות "רמז תמיד" או "רמז פעם" נקבל שאנו תלויים בהנחה הבאה.

בניגוד לפרוטוקול הקודם, כאן A לא יודע עם מי הוא מדבר. A מוכיח את עצמו על מי שמנה אלגון. יש פה יתרון וחסרון: היתרון הוא שאפשר להוכיח את עצמנו מעל רשת. החיסרון הוא (קשור) ש- A לא יודע עם מי הוא מדבר.

בדרך כלל משתמשים בפרוטוקולים האוטומטיים בדרכה המשולבת שתיארנו. במקום שנשים את כל ממש רמזים אל אופשר לשים פה הרבה plaintext chosen. נוכל מתמטית לשלוח הצפנות שלנו לשרת מסוים על מנת לקבל את ההצפנה.

פרוטוקול אפס-ידיעה

כדי תת תחום של פרוטוקולים אינטראקטיביים. אפשר להגדיר שאפשר גם ניתן לשאול את המטרה הפרטי של המאמת שום דבר.

דוגמה: יש שני גרפים Y-ניתן-לי-מ איזומורפיים זה לזה, אוסף שאותות מקוריות ניתן לשנות שהמאמת יודע שהם האיזומורפיזם. המטרה הוא ה איזומורפיזם כזה. יש שם מה שניתן להשוות וזה באמת יודע אפס ידיעה. אפשר להגיד שיש כ"ט כי בזיק המון מיכתן משנה לכדור גרפים הזה יוקים.

הגדרה שפרוטוקול אפס-ידיעה

א) שלמות - A תמיד יעל ל הלנית כהוא

ב) יעילות - מתקבל ע"י A ישלם מהסתברות שואפת ל-1.

ג) אפס-ידע - כולות מרובות על מסכימות מאוזמת על $(A)\text{Prv}$

פרוטוקל אל מפוסס על דיסקרטי

משתנה: p ראשוני יצוק ויודע

g יוצר של $\mathbb{Z}_p^*$ יודע

מטרת פרטי/סמוי של A: x מקרוי ב-$\mathbb{Z}_p^*$ (פרטי)

$y = g^x \pmod p$ (ציבורי)

אומת של A מול B:

A **מוכיח**

1) **commit:** A בוחר k מקרוי ב-$\mathbb{Z}_p^*$ ושולח את $s = g^k \pmod p$

2) **challenge:** B שולח $c = 0/1$ מקרוי

3) **response:** A שולח $r = k + cx \pmod p$

4) **verify:** B בודק $g^r \stackrel{?}{=} s \cdot y^c \pmod p$

ניתן לשנות את הפרוטוקול שלעיל כך שיתקיים:

(1) שלמות - commit בלבד אם שפ"ד.

- challenge בלבד ובלתי
- response בלבד אם שניהם.
- Verify - נמצא שכנה מאותו סוג.

$g^r = g^{k+cx} = g^k \cdot g^{cx} = y^c \cdot s$

A יודע את x והוכיח תמיד יכול להשתמש
מן שלב ההוכחה נדרש.

(2) **ובפועל**: המתכנה אינו יודע את x! קשה מן הסתברות להתחיל על משהו. אבל מדקדק שוות רק על ה response. וזה את ההוכיח ורוצה לעשות מה שהוא עלו.

נניח שההוכיח מקפיד ש- $c=0$. אז ההוכיח יתחיל k כל שהוא ואלו מקומם $c=0$ אז ההוכיח יודיע $r=k$ ואלו

$$g^r = g^k = y^0 \cdot s$$

אמנם כזה רק בגלל ה"חוק" ש- $c=0$. אבל כזה קורה ברוב ההסתברות $\frac{1}{2}$. אלו $c=1$ (וכן) רוצה $r=k+x$ ורוצה לעמוד במהתן. זה k במקום אלו על x אותנו לא יודעים עדיין. אפשר רק לנחש אותו בהסתברות $\frac{1}{q}$.

אז המתקיף יצליח בהסתברות $\frac{1}{2}$ ומתוך t פעמים הוכיח מצליח רק בהסתברות $\left(\frac{1}{2}\right)^t \to 0$.

אלו המתקיף מוכיח שיודע $c=1$. אז הוכיח יתחיל $s = \frac{g^k}{y}$ אם אכן וכיח אז הוא יצליח. אולם כמובן בדמה שיודע על המהתן הוכיח $r-k=x$ ושלח אותו במקום שיודע את x.

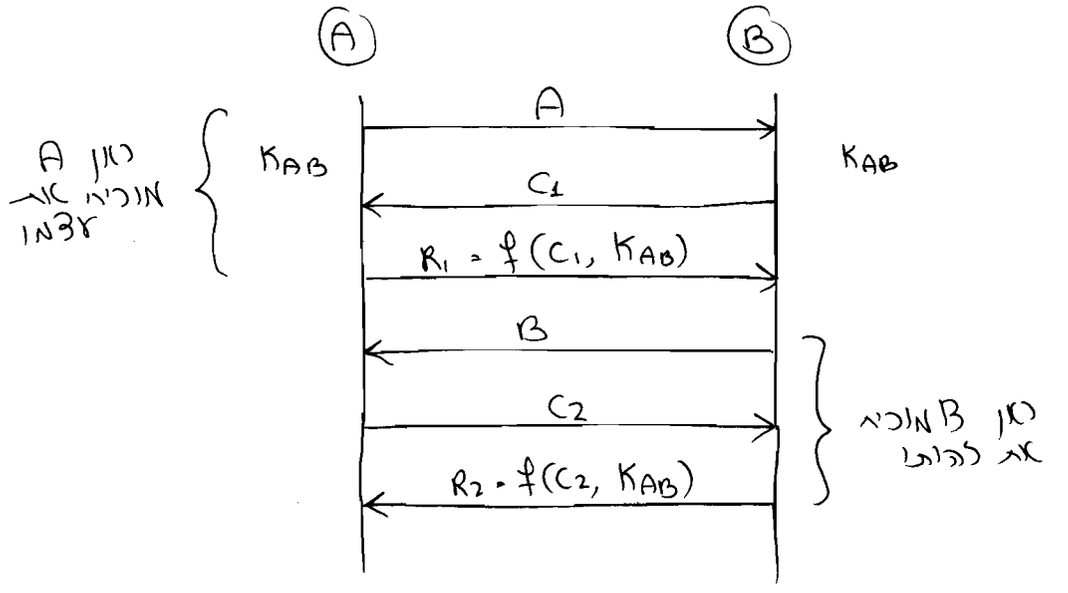
(3) **אפס ידיעה** - הרעיון הוא שאחרי ריבוי שיחות אפשר לעשות איכשהו ניתוח ולמצוא את x. מאיזו הוכיח הידוע שיחות (r,c,s) הוכיח s אחרי $g^r = y^c \cdot s \pmod q$. אולם c אחרי $1/0$! - r אחרי ב- $\mathbb{Z}_q^*$! $\pmod q$

נתשב $g^r y^{-c} = s \pmod q$ והוכיח יודיע אחרי כל הלל את x יודיע x. $\Leftarrow$ זה אפס ידיעה.

כי ראוי בשעה שהוא תביא לתוכיח לפני

פרוטוקולים לאימות + תכונה מפתח שיהיה

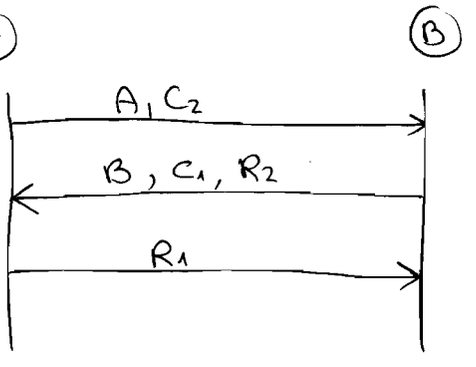
* אימות חד-כיווני



האיומים הם R_1 ו- R_2. נשאלת השאלה מי יאמת את זהות קודם? הרי אם יש אחד שמדבר עם A למה ש-A יכריח את זהות לאמת את זהותו אם הוא בכלל מדבר עם אחד כזה?

התוקפת'לי הוא כלה: מי שיתן ראיש יחתום שהדף השני יאמת את זהות קודם. אצל גאמש אם B היא שרת הוא ירצה ש-A יאמת את זהות קודם. אמנם יש פעמים שהן הלקוח יותר רציני. למשל אם אותנו מדברים עם הבנק, הלקוח קודם יכיח זהות כי טוב שהוא מדבר עם הבנק.

איש התקשורת הקאסי היה רוצה לחתוך את מספר ההודעות באופן הבא:



נראה יפה! תסכים איתי מההודעות. אלא שלא לא אותו הפרוטוקול, שהרי מתרחש התקפת Reflection Attack - מתקיף A. מתקיף מתחזה ומתחזה B מוכיח את זהותו קודם (אל A).

כאן פשוט שאנו את g האינפורמציה הכי חשוב קודם...

אך איך אפשר בכלל לאמת את הפרוטוקול ולשאול אם הוא בטוח?

בדיוק לתשב על זה...
פונה אל הלקוח כמה פעמים. כאמצע ישפה מציה אם שמישהו שיתים.
אפשר פעמים. אך היינו שלקוח יכול לראות את השרת. השרת בדיוק עכו
ראה אותו מדברים פה שפעם? כי לקוחות יכולים לשקר ולשרת
הראשונה שפוא פתח ולאת אוי האמתי R_1 ובכך להתחזות ל- A.
אבסיון השרת תשלח את התשובה R_1 ל- X וכך לתצור לשיחה

אבל הבעיה שיתה נוספת עם S:
אבל X לא יודע את K_{AB} אך הוא לא יכול לשלוח תשובה R_1.

אתקף X עשה את הפעולה הבאה:

שונים ושלח הראשון ושלח השני. בסוף השלב נעשה רשימה מלאה.
שהשתמש בפונקציה שונה בשני הצדדים, או שהשתמש במפתחות
אין אפשר להוכיח מהתקפה כלשהי שיש ביניהם - למשל,

פרוטוקול אוניברסלי:

הבעיה עדיין היא שלאן תאיד לה אפשרי. בסוף אחד יודע את כ
המפתחות הציבוריים של כ העולם".

אומת צבועים לה יהיה קריטי כפי לעשות את הדברים היותר מעניינים -
למשל ניהול שיחה מאובטחת. שיחה בדיוק מתרחשת בשלבים
הבאים: - אימות (תד/דו צדדי)
- התבפת מפתח
- הצפנת מידע

לפעמים לא עושים אימות דו-צדדי. למשל כשאנו משתמשים באינטרנט רק אותנו
עושים אימות שאותנו מעבירים עם השרת הבנק והשרת בטוח עד מעניין
אותו מי אנחנו. רק כדי אותנו רוצים לעשות לוסקה עם השרת את
השרת ים מאמת את הכתובת שלנו. למשל בתוקף של האתר של הבנק
את אותנו סתם יגלשם באתר ובוקים שעדי אכן השרת של הבנק בטוח
לכל מעניין אותו מי אנחנו. אפשר לנו יש דרך לדעת שאותנו אכן באתר של הבנק.
רק אם אנחנו רוצים להיכנס לאשמון הפרטי שלנו את אותנו משתמשים
בסיסמה כפי שהיה בהתחלה.

על פונקציית ערבול הדגש מהו..
גיבוב שונות. אומץ יש כאן את מחלקה $K_{AB}[i]$ את הדגים שלה. מאין ההנחה הקריפטוגרפיה היו מסוכן. ישנן של השיטה הלא היו שאינו לא פורשת
② שני הכפפים מחשבים $K_{AB}[i] = h(K_{AB}[i-1])$. בהנחה ש- h טובה שלה מסוכן. הבעיה היו שאם אורשאו מתגלה $K_{AB}[i]$ אז אפשר לפענח את כ מה שהיה) (אבל אי אפשר לפענח את כ מה שיהיה
① הדגר $K_{AB}[i]$ אוכפן באמצעות $K_{AB}[i-1]$ אבל הכף השני. ההנחה כאן היא פעמים או שמשתמשים בסוגי מפתח ה מון לאן בכיוון הקריפטוגרפיה ויותר מסוכן. אבל אם מצפינים את אותה

נניח ש- $K_{AB}[0]$ קיים אצל שני הצדדים. יש כאן סכמה:
פיסקרטי - יום / שנה | חלש.
נסמן $K_{AB}[i]$ המפתח בין A ל-B בסיבוב i ראשי הלאן הוא משתנה זמן ההנחה היא ש- A ו-B יכולים ליצור בהצפנה א סימטרית.

הצפנת מפתחות סימטריים בעזרת סימטרי

הרי לעיל שנצפין אינה אם יצרנו בעל כך יצליחו לפענח או אותנו אפשרים?
כאמור, נשים לב שיש לנו בתיאום מפתח רק אם היה לפחות שיחות אחד בעבר.

+ מפתחות שיחה
- מפתחות לטווח ארוך שמשתנים בתפירות יחסית ונמחקים מסוף השימוש
  + מניית מפתחות שיחה
  + אימות
- מפתחות קבועים (יחסית) שמשתנים בתפירות נמוכה.

יש היררכיית מפתחות

בדיוק לתאם.
קצת קשה מאוד. הדלות היא לעבוד בשכבת מפתח סימטרי ואותו שמוצפנים הצפנה. לפעמים יש לנו רק מפתחות שלא מתאימים מהותית אחרונו שאחרי האימות בדיוק לתאם מפתח. למה? כי תמיד יש לנו מפתחות

תיאום מפתח

מסוג. אומר לה פשוט מה שאנו יכולים. מה עם האימות? המקום
- שנשים את שניים אפשר שיהיה $g^x = h$. כאן עם הצבר מסוג איזה הפתיח
- B מאציא f אקראי ומעלה $g^f = m$
- A מאציא x אקראי ושולח $g^x = z$

הלוחות שבוצה: p ראשוני אפשר
g יוצר של $\mathbb{Z}_p^*$

פרוטוקול תיאום אפשר H-D מאומת

את g אפשרות השיחה הישנים.

אבל הצבר פס אופן אפשר, כי אדיין האתקיף וכן יכולה מ אלגי ו אפשר ישן
עם הצבר ואם ה פתיח אפשרים. אם g אתיק אפשר פרט, אלה אתיק מסוג,
כאן בשתי האופציות אם פוגשים ואחשב של g ו- g לא אוצא שלה את
מגיה עם לה היו שכן תאיב יש ל- A אפשר פרט. למשל באלה של פנק
ומלוות אפקלה אין אפשר פרט. היא אפקלה אתיק את לחותו ב אלגי סיסמה.
אפשר שמשתמש היא ששמתים את x אופן אם עושים אלו hash
מפיל אה שעושים היא ששמתים את x אופן אם עושים אלו hash
ומוצאים על ה- hash.

② A במה x אקראי ושולח אופן תחת $Pub(B)$ ותחם תחת $Prv(A)$
ו- B בדק חופש שלה את A.
באופן מיסרון אפקלה, ואלגי שאלו לפעמים לה את אים, ואחשב אם יש
מתרחש שאפטיה הלה B לא יוצא שאיא אקבל את האפשר ש- A. לה
של B.

① A במה x אקראי ושולח אומו ו- B אופן באימות האפשר הלחות
יש באה פתרון:

הצלחה אפשר סימטי בעזרת אסימטרי

(סיסמה) וכך נאפס את האלגי.
ה אפשר תסח. אם אישהו אינה אפשר אותנו בטח נפיץ אלה (כי ההתקפה היא
מואה אידאלית עם אפשרות. לה יקר אמף יש בטוח. יש הצבר אלגי ואם
③ התקפה פיסיי אחיפשת - פעם בראה לאן באמת אתקיפים בשני ו לפחים

ומשלב האתחול אין אתה יכול לעצור את הסימה.

ערשיו נפל שזה מיתכון שום פרטי הסימה B יודע שאינו אתה מתחבר עם A.

B מתניר $E_K(C_2)$

A מתניר $E_K(C_1)$

A שולח C_2

B שולח C_1

ערשיו הם צריכים לחזור שניהם יודעים את K ורשון:

ואין לו שום צרך לדעת אלה נתונים יותר טוב מנתונים אתר.

ניתשנו שון. התקוף לא יודע את X ו-Y בשלב ערייך הלוץ הדיסקרט

אם נניסה חור ע ו-M-ום בשלב שהשרטס הוא אקראיות שאינפץ אום

שעה הפרוטוקול הלה טוב? הרי מצפינים כאן פתרים אקראיים. אך אפשר

שניהם מתשתים את $k = g^{xy} \pmod p$

B שולח $g^y \pmod p$ כאשר y אקראיות

A שולח $g^x \pmod p$ כאשר x אקראיות

הנחת ההתחה היא ש- A יודע את $h = h(M)$

② פרוטוקול סימאולי תלק.

אנו צריכים שיהיה מפתח את המפתח הפרטי של B אך הסימה מצפה.

① ה תבורת RSA ל-B על ערוץ מוצפן בניהם (כמו SSL).

איך מתבוים סימה בצורה מאובטחת?

ה תבורת סימה בין A ו-B

פתרון קריפטוגרפי.

מאומתים להיות ומונים מפתח. אבל יש לזה מחיר כי מצריכות שה

אנו רוצים שה האימות כאן מתנה מפרוטוקול ההצנת המפתח. במה זתת כאן

g^x ו- $-g^y$ צריכים להיות מאומתים ע"י $Prv(A)$ ו- $-Prv(B)$.

יבוא ה- pass.

כלומר משהו כלשהו והרבה מאד לפחות בדוגמים שלהתקדם אחרי אינטן
אחת לה היא שה- pass היא לכן אחרת. לכן מוגנים מוגנים אחרים בשניה
רפ. אוד אופציה היחו תוכנה שמייצרת מספרים פסאודו - אקראיים. הבעיה
היא שמשתמשים בתוכנה "פופולרית" אבל לה עוזה הרבה
יהאקרים יכולים מספרים אקראיים מדויקה אל לה בשם.
פטרמינסטיים. אם עושים את לה פעם בשנה אל לה מסוכ. אבל אם בכל
(7) מספרים אקראיים טובים - כלומר "קרוב אינותי". אחשבים הם יקרים
- אחרת - עכ שהשורר את המפתח בביטחון שהוא מקום לכן אינו.
- כמן - לשלוף מפתח נפרד למן אינוקי, ברוש אחר מקום האינו
(8) שימוש במפתחות/אידע ביוש במוסף אינוקי+
(9) שימוש בך מפונקציות אחרות - כגון הפונקציה אפונקציה משנו
(10) הוספת שכנות - הוספת ערכים לתוכנית אקראית
(11) תוספת תיום + כיוונים שהוצעות - כשהשם header-ים
(12) שימוש במפתחות שונים לפעולות שונות } בקטע מפתח, אינוקי בפ, תתיאה וכו'.
(13) שימוש במפתחות שונים לפעולות שונות } בפונקציה / מפתח, אינוקי

תוקים נבניית פרוטוקולים טובים

נושא: אפיתוח סימטריים

יש כאן שני שלבים:

1. התקנת מפתח אחד K_{AB} של A ו-B
2. שליחה באופן מאומתת של מפתח K_{AB} מ-A ל-B (שלב זה יכול גם להיות מפתח אחר שנקבע מראש).

<u>שלב שלישי</u>: אתגר מפתח K_{AB} אפתח K_{AB}

<u>שלב שלישי</u>: אמירה באופן מאומתת אפתח K_{AB} של A ושל B.

שלב שלישי אמין - TTP - Trusted Third Party

כלי שלב שלישי מידעי לתקשורת בין A ל-B.

מפני השלב הראשון של בניית הארכיטקטורה שלב פיסי - לא הכי מסוכן שיש. אפתחות הם לא יש איזון. צריך שיש מידע שלבים שלבים שמאפשרים לארגון. אולי נשלים יותר על ② ו-③.

השמות הבאה של KDC (Key Distribution Center) שמבצע קשר לעולם הסימטרי, ושמות הבאה הבאה של CA (Certificate Authority) שמבצע קשר לעולם האסימטרי. אלה שני סוגים של TTP.

העולם שבו נכיל מחוץ היה עולם שבו יש n תחנות וכל אחת יכולה לדבר עם השנייה. כלומר, לכל שני תחנות A,B יש אפתח K_{AB}. אכ יש $O(n^2)$ אפתחות תשפיטיים בין תחנות. והאפתחות התשפיטיים ישמשו לבניית אפתחות שיהיה נקפיים. אחד האפתחות התשפיטיים נשנה בתפיחות ולכן יותר נאוכה. אלא n קטן אכ אפשר לעבוד אם n^2 אפתחות. פשוט אתקינים סוגי מפתח ועכ תחנה אחת ואפתחות הקרושים. רק - n אצל כל אחת לנהל קשה. קשה רק אדכרות-אחזקה של אפתחות (הוספה, שינוי, הגדה וכן הלאה).

אלא שני, הניהול של מרכזי היה שפיט אמולרת אין נק' כשל יחידה. אם אשתגים מל A בין התקשורת של A לפיצור ואין השפעה על התקשורת בין B ל-C.

לכן יצא (בדרך Diffie-Hellman למשל)
מחדש. אפשר את רוב הפרוטוקולים באיחוד אפשר לוותר אפשרות נוסף שה- KDC
הוא שאפשר לאותם ל- KDC. הוא לכן את יתרון. הוא לכן אלגוריתם אלגוריתם
הם יכולים לשתמש גם יכולים לוותר אפשרות שיחה אשרים. בפיל הנעשה
A ו- B יכולים לשותת באופן מאובטח בדרכים K_{AB} - לה האפשרות שאותו
KDC אומר ל-B את אפשרות השיחה K_{AB} באופן מאובטח.
KDC נותנת הסבלות של A ושולחו אפשרות שיחה, K_{AB}, עם B.
באופן מאובטח
A פונה ל-KDC בבקשה שיתאם לו שיחה מאובטחת מול B.
לאחר ובא שלב ה-KDC:

ענות ש- B ו-A מתוכרים ל-KDC. אין מיידים
איך התשובות מאובטחת ישיר אבל הם בטח כאלו רוצים
לשמר. אך מה הם עושים? מוסכמו הפתח ההתחלה עשויה
לאלו ובא שלב ה-KDC:
1) A פונה ל-KDC בבקשה שיתאם לו שיחה מאובטחת מול B.
2) KDC נותנת הסבלות של A ושולחו אפשרות שיחה, K_{AB}, עם B.
באופן מאובטח
3) KDC אומר ל-B את אפשרות השיחה K_{AB} באופן מאובטח.
4) A ו-B יכולים לשותת באופן מאובטח בדרכים K_{AB} - לה האפשרות שאותו
הם יכולים לשתמש גם יכולים לוותר אפשרות שיחה אשרים. בפיל הנעשה
כיצד ובא שלב ה-KDC. הוא לכן את יתרון. הוא לכן אלגוריתם אלגוריתם
פשוט. פתרון הבעיה הוא שיהיו מאוד אפשר לפחות את ההנחה הלוגיסטית של ה-KDC. אין צורך לפחות עם
אלא אחד, פרט לצורך ההנחה שמעיים משתמש אחד את כל יתר שאר
מפה כ- KDC הוא כ- Yellow Pages.
ההסבר הדרוש הוא ש- KDC הוא בעצם מקומון ומשמעותו רשם ותיק.
הוא אם צריך לדעת Online ם מכאן. אלא שבדרך כלל אנשים מעשים כאלה
KDC-ם כפי שהם לא יהיו בשאר מקומות מתקשרות. מול אלה מיניים
קיימים הרבה יותר מתחילים Security. אלו יהיו על ה- KDC ניה הבאה
יותר יקר אבל ארוכים ם תחנה שם משתמש הוא יותר גדולה.

תרחיש Intranet

מה ההגדרות של פרוטוקול מוסרי לדעתך?

- בדיון שהתנהל בין A ל- B כיצד לא ידעו השניים שעתי, אחד הם ובעלי זכות ואת לא.
- נניח כי B לא מאין (נניח ל- KDC) והתשובה היחיד שלו הוא לשלוח אפשרות זמני בדיק לשרות אלפי אפשרות. A פנה אליו כך והנחה ש- A מאין. אבל B במקרה לא מבוטל מאין וכך במקרה לא יתקבל שה KDC לתקן אחרין. כך יוכל שמשתמש את KDC ויתר אדמיניסטרטיבי הפרוטוקול. ב- Intranet ולמעשה הוא שתאים ע התנהלות מיוחד את אשר ישתמש בפרוטוקול הלה.
- מטרות הפרוטוקול הן מתאימה ע"י סמכות שרתי לקח. ושנית, אם B אפשרי אל A לא רוצה לתכנן לך שאפשר לפחות אם B

תרחיש Internet

מכאן כדיוס. ה- KDC נותן ל- A כרטיס כניסה ל- B. כש- A יתקוף שהוא לו ויכל לוודא שמדובר בו. הזהות הוא ש- KDC נותן שרק אפשר ווידוא מהתאים. לה B מוכר מדוע הקשר שלו את לה.

כאן הבדיל הוא שהפעם הצופן הוא E_{K_A} ולא $E_{K_{AB}}$.

2) A שולח ל- KDC: A, B, N_1 (מספר הפעלה חדשה) $A \rightarrow KDC$

3) KDC שולח ל- A: $A, B, E_{K_A}(A, B, K_{AB}, N_1)$ $KDC \rightarrow A$

תרגום סימטרי

nonce

כאן יש הבדל מהותי בין הצופן הסימטרי לבין הצופן האסימטרי. בצופן הסימטרי אנחנו מכירים את המפתח ולכן יכולים לייצר את המפתח הסשן בעצמנו. לא כך בצופן האסימטרי. יש שם delay מסויים בין הקלט שנותן לנו ל- B ועד שהוא מגיע ל- KDC. יש שם אפשרות לבדוק שהוא שלח את הקלט ל- KDC. לעומת זאת בצופן הסימטרי אין אפשרות לבדוק שהוא שלח את הקלט ל- KDC. כלומר אם הקלט מגיע ל- KDC שאין לו מושג מי זה B, הוא לא יודע מה לעשות עם הקלט הזה. לעומת זאת בצופן הסימטרי אין לנו מושג מי שלח את הקלט ל- KDC. כלומר אם הקלט מגיע ל- KDC שאין לו מושג מי זה B, הוא לא יודע מה לעשות עם הקלט הזה.

תרגום Extranet

אנחנו מניחים כאן ש- A מקבל את K_{AB} מוצפן עבור K_B, מייצר אותו בעצמו ושולח ל- B. הבעיה היחידה היא שאנחנו לא יודעים מי B. כלומר B יכול להיות כל אחד ברשת. אנחנו מניחים שהמפתחות K_A ו- K_B הם כבר משותפים בין A ל- KDC ו- B ל- KDC. כלומר B יכול לבדוק בעזרת TKT_B שהוא אכן מי שהוא אומר שהוא. אבל אז עדיין לא ברור לנו איך A יודע שהוא מדבר עם B. כלומר B יכול להיות כל אחד ברשת. אנחנו מניחים שהמפתחות K_A ו- K_B הם כבר משותפים בין A ל- KDC ו- B ל- KDC. כלומר B יכול לבדוק בעזרת TKT_B שהוא אכן מי שהוא אומר שהוא. אבל אז עדיין לא ברור לנו איך A יודע שהוא מדבר עם B.

מוגדרים כ- plaintext attack.

תוך-מידה, נשים לב שאם A הוא מתחזה את ההודעה לא אומרת לו
דבר כי הוא לא יודע לפענח את ההצפנה עם K_A.
אם A מתחזה KDC לא יוצא את זה אפילו אם מטרה אותנו אחריות
security. הפרוטוקול הוא הזוגי הראשון שלנו.

3) KDC שולח ל-A את TKT_B: $A \leftarrow KDC : E_{K_B}(A, B, K_{AB})$

כלומר K_B שאף אחד לא יודע חוץ מ- B. כל אין מקום להצפין
שוב (תחת ההנחה שלנו אכן KDC שמלא את ההגדרה)

4) A פונה ל- B: $A \rightarrow B : A, B, TKT_B, N_2, E_{K_{AB}}(N_2)$

N_2 אומרת לאחור ש- A הוא אכן A, מאחר שה- TKT_B הוא אכן שלו, כי
רק הם יודעים את K_{AB}

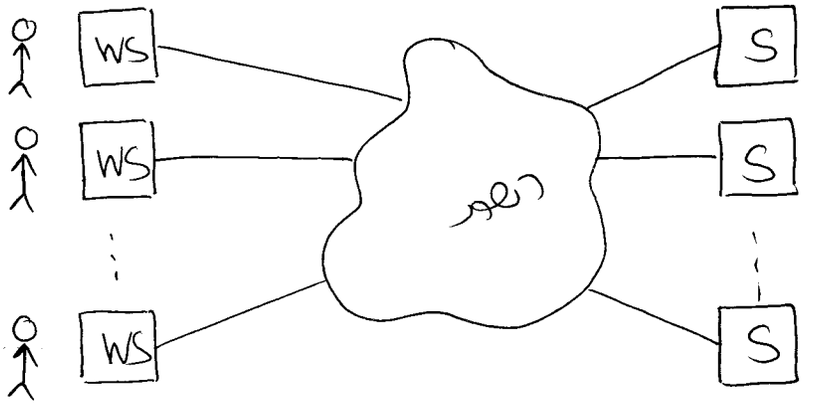
5) B מוכיח את זהותו ל-A: $A \rightarrow B : A, B, N_3, E_{K_{AB}}(N_3)$

נשים לב שבלי N_2 הוא סתם מספר אקראי כלשהו אפשר לעשות
replay attack...

(55) 26.12.04
הלפנה

Kerberos

בשנות ה- 80 התקדמות התחילו לפתח שיטות המופץ על מנת - שלא יהיה צורך עם התשה הפרט, בלא אפשר לפרוץ תוכנים על שרתים מרוחקים. משתמש ירצה להיות עמדה ומכה עצמה עם שם.



אבל איך עושים פה מופע שם אימות לקוח/שרת? איך עושים אימות זה לפרוץ אופן אוטומטי בין משתמש לשרת מרוחק? מכשף צריכה לאפשר גישה בצורה אוטומטית.

המטרה היתה שיהיו 3 איזונים: - Authentication

- Authorization

- Auditing

ב- kerberos (פה תא-ראש מה מיתולוגיה היוונית) המקורת היה לך אימות.

אף משתמש שרוצים לא single sign-on. פירוש הדבר שמרצף שעושים פעם אחת login לתשה איך מקבלים את כל השמורים האפשריים. זה שונה מהאינטרנט - שם כשאנחנו מתחברים אתרים לא איך מתחברים עם שונה מהאינטרנט - שם כשאנחנו מתחברים אתרים לא איך מתחברים עם הסנקים כאלה... אבל במרכז עשתה היינו לקבל את כל השמורים בפני אות. לזה נעשה ע"י שרת KRB.

אל מקום שלנו יש אחלה צמוד - משתמש A, שרת קלה S, ותוך שמורה כ שם: ! - KRB.

TGS - Ticket Granting Server
מה מקשר TKT וקפאו.
כן בדרך שוחים לוותתרו לשת מתבדים שלום 8-5 . 5 ! כן
לעשות login לרשת שעשה משתמש V-1.
AS - Authentication Server - תפקידו רות לאשים אפשרות

מה שניה. אנחנו נתאר כעת את V4:
כיום יש לקרברוס שתי גרסאות מסחריות - V4 ו-V5 אשר שונות את
ההפעלה.
עיין ההפ שיפרת לעבוד באופן קרברוס. היום לה הכי אופנה ב-מערכ
שיפרת לעבוד בשיטה הזאת. לה נקרא קרברוס בלבד - תוספת תוכנה
תוכנה שיפרת לעשות את מה שרוצים. זה נשמע בהיות תוכנה
המשתמשים וקשה מאוד לגרום את לה. מה שיש בתוכנה עבודה לה
תכיר את המשתמש כי כך ע תהנה הלקוחה בהיות להכיר את ע
הלקוחה אין שום מידע של security. כי לא היינו שהתוכנה הזאת
למשתמש A יש סמאו PWA ונשתה S יש אפתח K_S . בהתחת

נפרט את התהליך:

① A משתדק עם SM

- A מקיש את שמו ו- SM
- מאתרשיו (אחר כך SM+A מוחק

② SM+A פונים ל- SA

$< A, WS, TGS, Time\ Stamp_1 (TS_1) >$ אין פה זיהוי !

↓ הזיהוי שיהיו הוא רואה רק יהיה את הזהות.

③ SA עונה ל- SM+A עם NC_{A+WS} (Network Credentials)

$NC_{A+WS} = E_{K_A}(< K_{A,TGS}, TGS, TS_2, LT_2, TKT_{TGS} >)$

↓ הוא למען אותנו מרשים ל-NC להיות מוצפן

כאשר

$TKT_{TGS} = E_{K_{KRB}}(< K_{A,TGS}, WS, TGS, TS_2, LT_2 >)$

↓ מפתח שקלברוס הוא יודע לפשיפה הבלוי LT_2 הוא טוב למען

↓ מפתח סודי בין-תלין שמוכר לקלברוס

④ SM+A מפענחים ושומרים את NC_{A+WS}

- A מקיש סיסמה PW_A
- SM מחשב $K_A = h(PW_A)$ ומוחק את PW_A מהמידכון (הסיבה לכך היא שאנו רוצים שה סיסמה תהיה מסיכון - הסיסמה הרבה יותר רישה מ- K_A) בדרך מואת
- SM מפענח את NC_{A+WS} ושומר את השדות הרלוונטיים (הוא יודע שהפיענוח עשה כיון את השדות ויודאו בכך - משמעות שכן למעלה) ומוחק את K_A מהמידכון.

⑤ SM+A פונים ל- TGS מבקשים שירות מנ"ס

$< S, TKT_{TGS}, auth_3 >$

כאשר $auth_3 = E_{K_{A,TGS}}(< A, WS, TS_3 >)$

⑥ TGS מקבל מ- A+WS את $SC_{A+WS,S}$ (Service Credentials)

$$SC_{A+WS,S} = E_{K_{A,TGS}}(< K_{A+WS,S}, S, TS_4, LT_4, TKT_S >)$$

$$TKT_S = E_{K_S}(< K_{A+WS,S}, A, WS, TS_4, LT_4 >)$$

WS מפענח את ההצפנות

⑦ A+WS שולח ל-S

$$< TKT_S, auth_S >$$

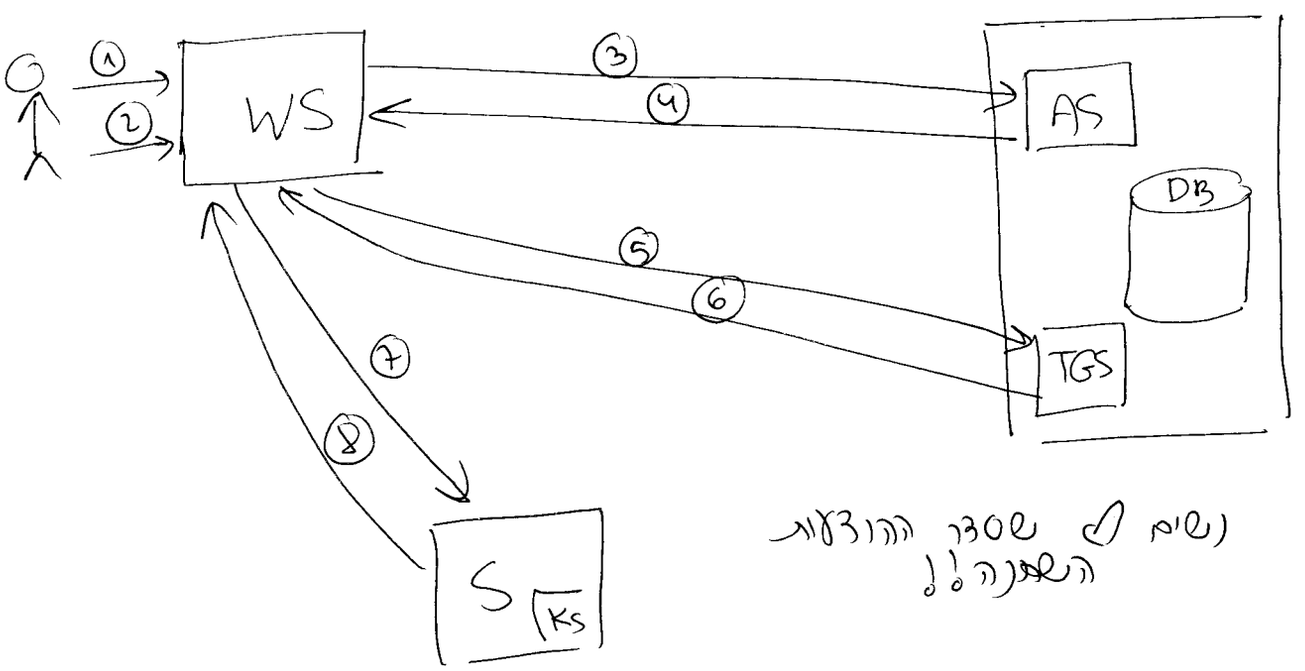
כאשר $auth_S = E_{K_{A+WS,S}}(< A, WS, TS_5 >)$

⑧ S מקבל מ- A+WS

- S מפענח את TKT_S ומקבל

$$auth_6 = E_{K_{A+WS,S}}(< S, TS_6 >)$$

KRB V5



נשים ♥ שסדר ההודעות השתנה?!

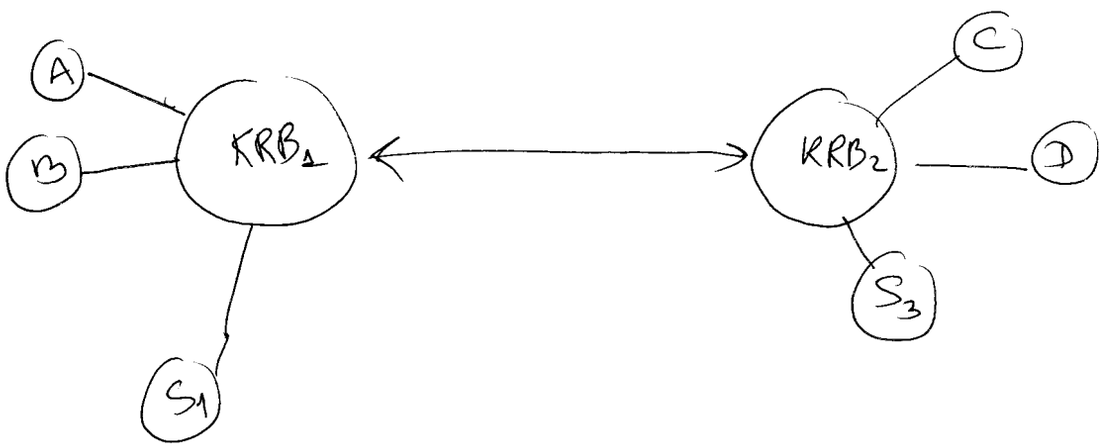
שינויים בגרסה V5

- אוירה - השתמשו אנשים שם 1000 פעמים ביום, ולכן הלקוח מבקש את K_A ושולח תקופת אוירה ארוכה (מעל TS1) כדי לא להתאים מ AS על סתם מתי לה.
- אין הצפנה כפולה של TKT - לכן מאוד קל להפיץ אותו פתאום כי מתאו הכי בפרם ההראשונה היא מוצפן ומיכן ע"י K_{KRB} אז אין צורך שהבנב למן ומשהים כדי להפיץ אותו.

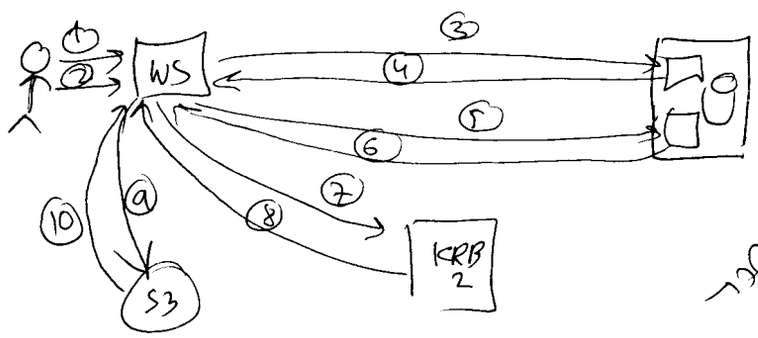
- Postdate , Proxy , Forward

נשים לב שמשתמש בקרברוס רואה שרת כאוסף פונקציות מסוים.

• Forward - למשל כאשר שימוש מתחנה לתחנה / משתמש משתמש אחרי
• proxy - לצורך מי תחנה שמאפשרת אותו בשם תחנה אחרת
• postdate - לצורך במשך הרבה זמן (שבועות למשל) אף אחד לא מחובר. הרעיון הוא לאפשר הגדרה כרטיס לתאריך עתידי או כרטיס שניתן לחדש.
- Realm - יקומים. קרברוס עובד ברמה הארגונית. ניתן להגדיר הסמכות / אוניברסיטה ת"א אחריות קרברוס. אנחנו הן כדורות לדבר אחת עם השנייה!



KRB_1 הוא לקוח של KRB_2 ומספק.
נניח ש- A רוצה לגשת ל- S_3. הוא עושה login מול KRB_1. מיושה ל- TGS אומרים שרוצה לדבר עם S_3 שנמצא ב- Realm2. את KRB_1 פונה ל- TGS של KRB_2 ומקבל שרות. את KRB_1 נותן ל- A כרטיס לגישה ל- KRB_2 וכך הוא נותן כרטיס ל- A לגישה ל- S_3.



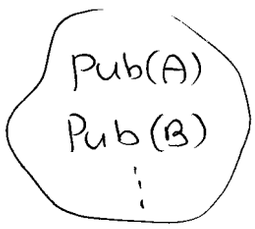
אמתיות A KRB2 הוא שרת שמנפיק כרטיסים לשרתים אחרים.

DCE - הדרכה לאימות הרשאות

PS - Privilage Server - נותן הרשאה למשתמש

לשרת שהוא רוצה מקבל כרטיס.

ניהול מפתחות עם מפתחות אסימטריים



איך נדע
שהמפתח
הציבורי
של מישהו
יודעים
?

יש כמה נושאים שיש לדון בהם:

- מנהל מפתחות
- הפצת מפתחות
- מיצוי מפתחות

מנהל מפתחות

- A בונה מפתח פרטי / ציבורי, שומר את הפרטי לעצמו ומקים אתר ואופיס את הציבורי לכולם (לפחות בהתחלה)
- A לא תמיד מעוניין שכולם יכירו שלו (לפחות בעצמו) (כגון מתנאים שבהם A סומך רק על עצמו)
- כדי שמישהו יוכל לשלוח מהודעה שרוצים שכולם מפתח פרטי/ציבורי. שרת נותן ל-A את המפתח הפרטי, ושומר אותו.
- יש שרת אחד (TTP) שכל הוא בונה מפתחות פרטיים/ציבוריים ונותן מפתח פרטי ל-A (ומוחק שום אותו)

הפצת מפתחות

- A או שרת הפצה הם האפיקים
- התקנה פיסית אצל מקבלים (כמו בטנה אבל קשה למימוש)
- התקנה באתרי אינטרנט מאובטחים (כמו להב וכו')
- שליחת המפתח עם פורמט כתוסף ל- email, doc וכו'. כאן קשה להבטיח מטומים שלא אחר המפתח ובטוח אבל אם אותו מפתח מתקבל מהרבה אנשים שונים אותנו מתחילים להשתכנע. אם אחד משנה את המפתח שלו אז מהר שהוא שינה, אפשר לזהות אותו ולא לקבל ממנו

שאלה בכלל צריך? - אבל האמת הפרטיזציה צריך לה ודיא עלים שאנו
יכפיונו יותר מעלות האמת הלימודי האתרים
- מה מישהו פיזית את האמת הפרט?
- איך האמת פה-תיק? (למשל את עצמים
את האבטחה)
- A יכול להפיל את האינפורמציה
- שרת יכול לזהות את הגיזע
משתמש שרוצה לשלוח את A צריך לפתוח את התקפות של Pub(A) !!

תעודת/אשור - certificates

עכשיו משה רוצה לשלח ואומר "שלום, שמי משה כהן
רק רק, מאוד מיהרתי רק רק, פסיכומטרי רק רק
ואני רוצה להתקדם מהר לה".
אבל איך נדע שמשה שאנו אומר לה שזה צריך להיות
תעודת מהרות:

ראשון ניתן לציין שם, שום בעיה תעודת מהרות - אלו התחייבה כן
הגיה יומם בדיב אותנו אנחנו אנחנו שאם ותעודה ואלו-מסר אומתנו
סומכים עליו (עלא מתקשרים עמהם ומספר עוזבא).
אותו דבר יש לאמתות ציבורי"ם:

עניין שבא מתוך מיומנות אם תזדקק יפה עם פקידים
ומנהלים ותתנהג. אך הסתם לא נוכל להשתמש כדי
אין לנו שום מושג איך נראה תקופת ביניים של מיומנות.
אז מה נעשה?

(A) $Prv(A)$

CA — certificate authority — רשות התעודה

(B) $Prv(B)$

A: cert(A)
B: cert(B)
C: cert(C)

(C) $Prv(C)$

CA מנפיק את $cert(A)$ כאשר

$$cert(A) = \underbrace{\langle A, Pub(A) \rangle}_{\text{תתום ע"י CA משתמש ב-} Prv(CA)}$$

כדי לוודא שמסמך הוא אכן שעתה האינפורמציה בו (לא מצוין כאן שעתה האינפורמציה בו) צריך לדעת שעתה ב-$Pub(CA)$.
אבל עדיין יש לנו מלא מחלוקת. מהיכן יהיה לכולם את $Pub(CA)$ בלי
את"ל צריך לתקין פיסות מידע הכלכלתיות לאריאון! יהיה ראוי
ביאו כאומן שה- CA הזה אמין! ל- CA יש מפ"ב אפתחות
מחוץ מאוד מאוד גלקים (אפשר ב- 2000-4000 בוליס) שהרי
(ו) מחוץ רציני.

מניין אפתחות
שקרים טאפיה של CA לאנשות להתקפה כזו שאר שהפרוטוקול משם.
נשים לב שהפרוטוקול כזה (כמו ב- man-in-the-middle משם בניין האפתחות - כזה

- A בונה אפתח פרטי/ציבורי
- שרת מוציא אפתח פרטי/ציבורי. אותנו מעניין שקשור ל- CA יעניה אפתח פרטי/ציבורי. CA אושר לך
לשמש את הפעולות של שותפים רגישים.
תקיפת MITM - שם אחד אותנו רוצים תעודה
אבל אנחנו מבין ניגש לשרת אתוך אשר יאמר את שהוא לך את"כ
יש ל- CA שקשור את תעודה הנכונה.

בניית cert

A פונה ל- CA (נאמר לשם שינוי ל- CA) ומבקש cert.
ישנם מיני רמות של אחריות והאישור שהוא נותן
מהווה הגישה הפשוטה. למשל, אפשר לקנות מ-VeriSign תעודה
בעשרת ש"ח פנייה מאיימל אלמלחין אמצ כמותן רמת אימות
נמוכה כאן מספיק ואת התעודה לכו שווה הרבה. אבל מצד
שני אפשר להשתמש בתעודות כאלה למשל משהו מפיקי
אתרים או משהו עסקאות שלא כרוכות בהפסד יקר...

הפקד cert

- בדרך כלל ה- CA עושה זה.
- שרת שמותקן ל- CA
- A מזמין
- שירת הפקד cert.

ביטול cert

זה נותן מאוחר סיבות כמו מיטול אפשרות.
אם A מפוץ הצדדה לעיתונות ואומר שהתעודה שלו לכא מתוקף
יתכן לה לכא טוב כי צ אחד יכול לעשות את זה. אבל
אם A חותם על ההצדקה אם האפשרת הפרטי שלו אז אותו פרטים
שהאמין עבור - כי או ש-A מצוות התמון לכה או שיגנבו לו
את האפשרת והפיקו את ההצדקה ואז כמובן לא נוכל שהשתמש בכה.
אופציה יותר נורמלית היא ש-A פונה ל- CA ו- CA יפיק את
ומיד כלה אם התנאים שלו.
אם CA יודע ש-A רמאי או משהו כזה את CA יכול לבטל
מצמו מיטול של התעודה של A והמיטול הלה כמובן יש לפרסם
ברשימה של CA, אותה נה מסר שכן.

הפרת שניים / מיטלים - CRL (Certificate Revocation List)

CRL (Certificate Revocation List) הוא מאגר שבו מפרסמים תעודות שניים. הוא לא נשלח באופן פעיל לכל משתמש, אלא נשמר במקום ציבורי שבו כל אחד יכול לבדוק אותו. כלומר, אם משתמש B מקבל תעודה מ-A הוא לא יודע מראש אם A עדיין מורשה. הוא חייב לבדוק את ה-CRL כדי לוודא. כמעט כל מערכת PKI מפרסמת CRL תקופתי. אם רוצים להיות מהירים יותר, אפשר לפרסם CRL חלקי שמכיל רק את השינויים מאז הפרסום הקודם. ה-CRL מכיל רשימה של מספרים סידוריים של תעודות שבוטלו, יחד עם תאריך הביטול וסיבת הביטול (אם ידועה). כל תעודה שמופיעה ברשימה הזו נחשבת לא תקפה, גם אם תאריך התפוגה שלה עדיין לא הגיע.

פורמט X509 סוג של תעודה דיגיטלית

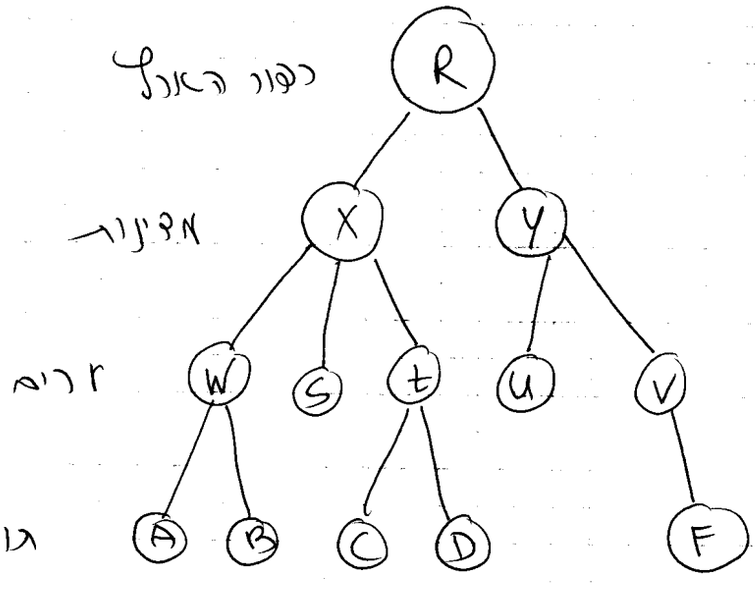
מספר הגרסה של התעודה	version number
מספר סידורי של התעודה	serial number
אלגוריתם ההצפנה בו השתמשו, מפתח ציבורי של המנפיק	signature parameters
שם המנפיק	Certificate Issuer
תקופת התוקף של התעודה	Not Before
	Not After
שם הנושא (פרטים אודותיו)	Subject Details
המפתח הציבורי של הנושא	Subject-Public-Key
שדות נוספים - נתונים אודות שלב, מאפיין נוסף, הגבלות שימוש של התעודה?	Extensions
חתימה דיגיטלית של המנפיק	Signature

המשך ההרצאה נמצא במסמך נוסף.

ניהול אמון מפתח ציבורי - Public-Key Trust Management

המצב האידאלי הוא שבו ניתן לסמוך על CA אחד בעולם. יש המון אינטרסים ואינטרסים שונים. ולכן כולם מוכנים לעבוד עם כולם. מצד אחד, אפשר שכולם היו מעדיפים שכולם יסמכו על CA אחד ויעשו את כולם.
אבל יש CA-ים שונים עם מטרות שונים - למשל מישהו יכול להיות CA שמרוויחים מספק שלו.
הניהול של ה"אנין הזה נקרא Trust Management
אנחנו נתן ארבעה מופעים.

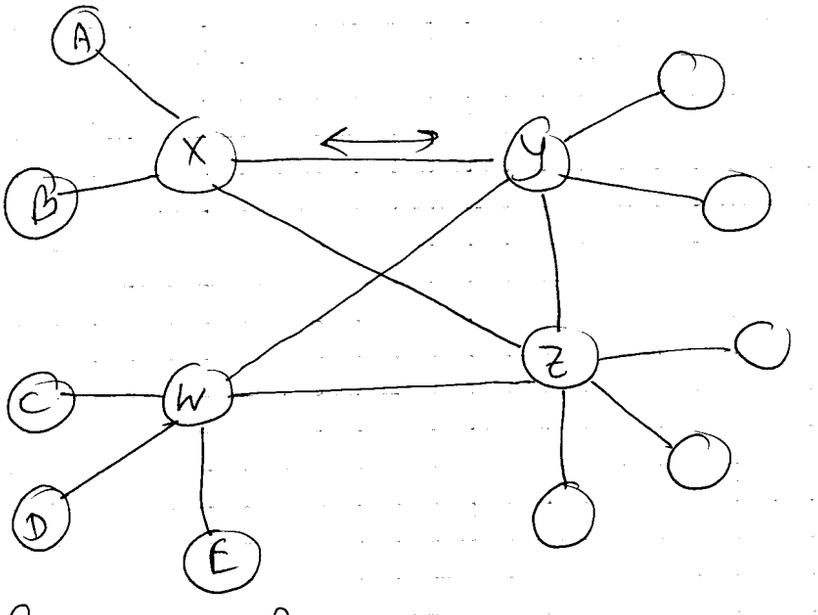
① מופע הירכי



* לכל עלה יהיה אפשרות לדמות של R ושם אמא שלו.
* ל CA מופיע תעודות לבנים שלו.
אם A רוצה את ה אק של F נאמן α מוצא α <<β>> - <<F>>V , <<V>>Y , <<Y>>R
תעודה אחרון β. אל מה שנדרש הסו

זה נקרא אספלן Path של אזורות.
ועוד של F לו תוכן מטרתו. אחרת ניתן שיהיה לאדם שלה מאוניברסלי ובקרוב
ששם שאומר <<F>> V. כזה המחשה ולכן ויבקש <<V>> y
והוא מתוכו מפקח <<y>> R. לזה נראה פשוט אבל זה פו
מורכב ראוי שזה. זה נקרא תשלית האזורות.
המוצג הלה ראוכן לא מדויאות. לזה ישנם בתוך אריאנים אחרים.
לאחרונה ל- IBM יש אינלוזיה ויכול להתפיק את אזורות
עסקיים שהם ומסיפים עובדים. לזה מתאים למבנה שיש לו
אריאן היררכי - כמו בצד ראשון.

ב) מופע אריאני



יש אריאים שיש ביניהם יחסי אמון. ולאריאים יש לקוחות
(לאחרונה בנקים). אלו יחסי הלקוחות הוא כאמון לא מלא.
* בי אריאן מפקיד תעודות על לקוחותיו.
* בין אריאים יש cross-certificates - הם נותנים
אחזורות אחד לשני. אם A מעוניין להיות ל- D:
$x <<y>>$, $y <<w>>$, $w <<D>>$ או
$x <<z>>$ $z <<w>>$ $w <<D>>$
פה יש מה לחפש על path exploration - אלגו-
ריתמים. זה לא ברור כאן כמו במודל ההיררכי. יש אם
יש כמה אספלים איך בוחרים? (אורך, מחיר, אלוק ...)

⑤ מודל שכונתי

* ל CA אפשרי ברשותו עשרות לקוחות
* שאלת הרשאה של CA-ה שאין סומכים עליהם.

כלומר שלקוח ה- browser במחשב. במחשב יש רשימה של CA-ים שסומכים עליהם ולקוחות שהם הם אותרי האוניברסיטה. מי שם את זה במחשב? מיקרוסופט. למה? כי הם שילמו להם. למה? כאן נכנס מושג שנקרא עסקים. אלגו שנקבע הלכה מאוד קלקולית. אפשר לשנות שם CA-ים עבור מאות"ם.

⑥ מודל של Web-of-Trust

אין מאוד כעולם
לו יש תמרים.
עבדים של תמרים וכן
הלאה.

נדבר פה מודל שרכות של אמון. לתמרים שלו יש פרטי אמון מאוחה מיותר. לתמרים עבדים אני יכול עבד תוכנה לעת אמון מדויקת הרבותה ביותר. לזה כמו LinkedIn כלי.
PGP ומזה כמה.

תשתיות PK

תשתית - מה זה?

- שימושי מאוכן ניתב
- ניהול
- פשוט
- אמין
- סטנדרטי
- שקוף למשתמש

הרעיון של תשתית PK בעצם כמו שיש הרבה PK
כך רעיון נתמך, אפשר לשים אותם?

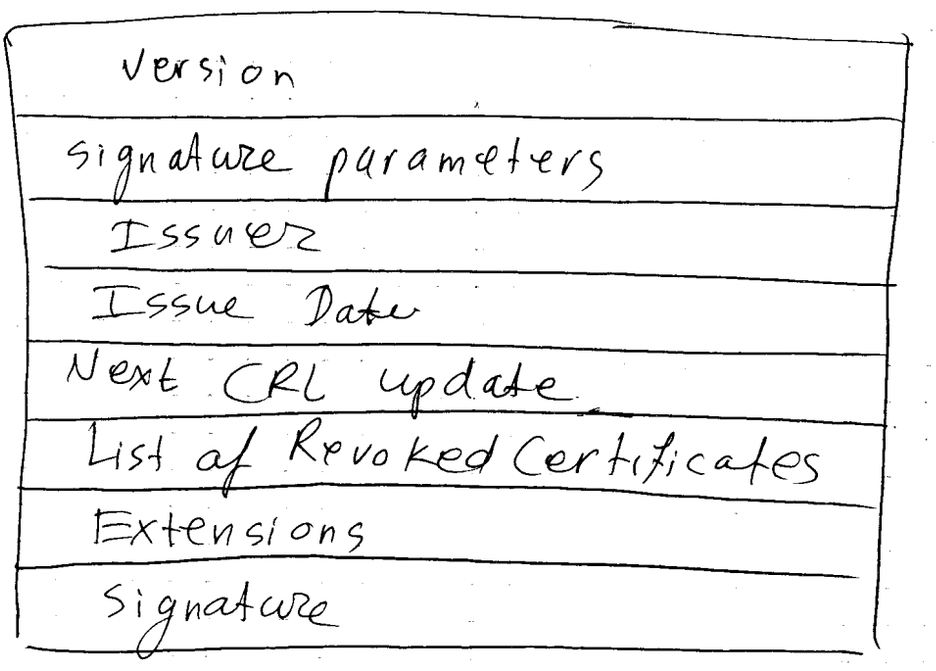
שירותים של PKI:

- רישום המנויות - לדאוג שקיבלו ביטחה מנויות, מה מתאר ומה לא טוב. צריך להיות מסמך מנויות. מה אפשר לעשות עם התעודה וכו'. (בדרך כלל)
- רישום certificates - זה ה- CA. הפעולה היחידה והחשובה ביותר שהיא עושה היא ביצוע התעודה והחתימה עליהן. ומאחת הפרטים שלו מתוך ומיוש.
- רישום ביטול - המשתמש מבקש שאנו נבטל, מצדדים ומביאים את הבקשה. ומהו כך שפועל ה-CA נבדוק שהמבקש אכן הוא מי שנכתב בתעודה שהוא רוצה לבטל.
- אחזור certificates
- אחזור מפתחות - צריך לבדוק שהמפתחות פרטיים, למשל למקרה שהאחר שלם. בדרך היחידה היא צריך שיהיה עבודה ומאין אחוד, כך שהמאחר לא יוכל לשנות. צריך לשמור את מפתחות פרטיים - זה תהליך הדורש מאוד מהירות. צריך שהמשתמש לא מפקיד צדדי ואין לו שום את זה בדרך מאוד מאוד.
- לך כאן מפתחות פרטיים - הרי מפתח סימן יכול מאוד. אין זה קשה בלי שהמבקש את המשתמש? איך המפתח את עצמו אוטומטי?
- אחריות / ביטוח - מפתחות משתמש שמא הוא מפקיד, צריך לשמור/לוודא כי אותו נוכלים שפתח אותם ולוודא אותם זה את מפקיד. כל שנים וכו' מהיום.
- ניהול התעודות - שינוי, הפקה, ביטול וכו'.

ביטול / התיאה של תעודות

יש שתי גישות. הראשונה אומרת שפרוטוקול מיהן התעודות שאתנוטלות (CRL) והשנייה אומרת שפרוטוקול מיהן התעודות שמתוקף (CAL). ההחירה באופן תלויה בשלב ובמורכבות. באינטרנט בדרך כלל משתמשים ב- CRL. הרי מדובר במיליארדי תעודות. ובאופן יש עניבה לפקידה אבל זה הרבה פחות.

CRL
X509



כפי שניתן לראות מהתמונה, ה-CRL מכיל את המידע הבא:

- **Version** – מספר הגרסה של ה-CRL.
- **Signature parameters** – פרמטרים של האלגוריתם שבו נחתם ה-CRL.
- **Issuer** – שם הגוף שהנפיק את ה-CRL.
- **Issue Date** – תאריך ושעה שבהם ה-CRL הונפק.
- **Next CRL update** – תאריך ושעה שבהם יונפק ה-CRL הבא.
- **List of Revoked Certificates** – רשימת כל התעודות שבוטלו, יחד עם תאריך הביטול של כל אחת מהן.
- **Extensions** – שדות נוספים שמוסיפים מידע נוסף על ה-CRL או על התעודות שבו.
- **Signature** – חתימה דיגיטלית של ה-CA על כל המידע שבמסמך, כדי להבטיח שלא שונה.

כל פעם כאשר צריך להיות תקפה ע"י ה- CA.
מוריפים את הפלט מאותאריך (מהונתה שהפלטות קיבלו)

CRL Delta – תושים רשימה פעם מחודש וכן יום
שנים. בדיק מרשימה שלמה כותבים אלפה אפשר לבקש
שהיא תהיה מעודכנת. כדי לאפשר מקל את הבקשה.

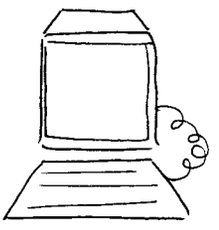
Distribution Points – מפנים אל ה- CRL של שרתים
CRL complete – ם הקלף מאשאו. ויכל להיות ששרת אחרים
יכיל את ה CRL הנשלח ם יום אחד תכיל.

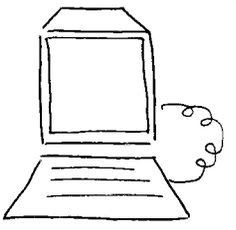
כל פעם כאשר צריך להיות תקפה ע"י ה- CA.

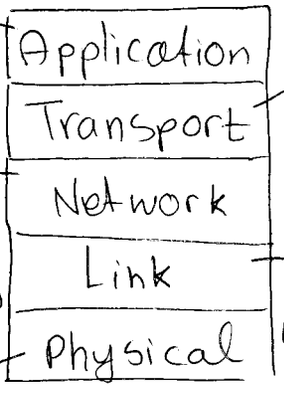
אבטחת מידע ברמת הרשת (IPSec)

היום נדבר על אבטחת מידע ברמת הרשת. סטנדרטים ומוצרים שונים של פרוטוקולים.

איך נראה מצב תקשורת ב
יש הידרדרות של
פרוטוקולים.







שנותים כמו http, ftp, smtp. כאן מדובר בדף הסופקציפי של "שומים.

כאן קבוצה ניהול השרה מתעלמים מהמיקום הפיסי כאן יש פרוטוקולים כמו TCP או UDP.

כאן הכתובות קבועים לכל ציוד מחוברים ישירות ובדרך כלל שלה תחנות-קצה. לה יש מזהה כתובת IP (V4/V6) כל תחנה של הנפסי תשתף.

כמה שמטפלת משרות בין של תיבור ספציפי אל יבי Ethernet, באופן מיוחד, יש כתובת MAC.

מפרט IEEE מטפלים בלה.
יש פרוטוקולים לאבטחה בתוך הרשת, אפשר לבנות על לה.

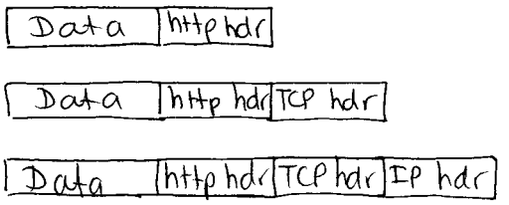
פרוטוקול אבטחת מידע ברמת אנלוגית.

איפה בדיק לתת את האבטחה? ברשתות הכי נמוכות ותלוי מה משתמשים. ראשון, אם עושים רק אבטחה פיסית אז לכאורה לה מדברים. לה שכבות מאוד מזומנת. אולי נדבר אולי לאחת יש ישויות עליונות אלו ותכונות פיסיות ולה מלת אפשרי ברמה הפיסית.

ברמה האפליקציה אפשר לתת הכמה יתר אמינות לאבטחה. מצד שני, התוכן ובדרך כלל יש צורך שאנשים בדיק לאמת אותם לה אחד ולה לצד שלם.

כשמדברים מידע ברשת יצרים הכוונה מקוללת של מידע לפי ההבדלים בתכונות הלה נקראות packets.

לאשר, נניח שמדברים ב- http:



מה מוצא ב- headers האלה?

תכונות פרוטוקול IP

מאלו האפשרויות לא היתה שייכת לפרוטוקול IP? שים פתור כלי פתח
לפני 30-40 שנה מהרב רבי ליזר תקשורת בין אוניברסיטאות וכו'.
צ'ט פיתרו על אופן כמה מחשבים קטנים והמטרה היתה מוקרית שלהם.
ההקלטה היתה כך שלפני שלח ציפן ומרשו יש לו אינטרנט.
מי שמאזין יכול למשש לחשוב בשלבים.

ההבדלים:

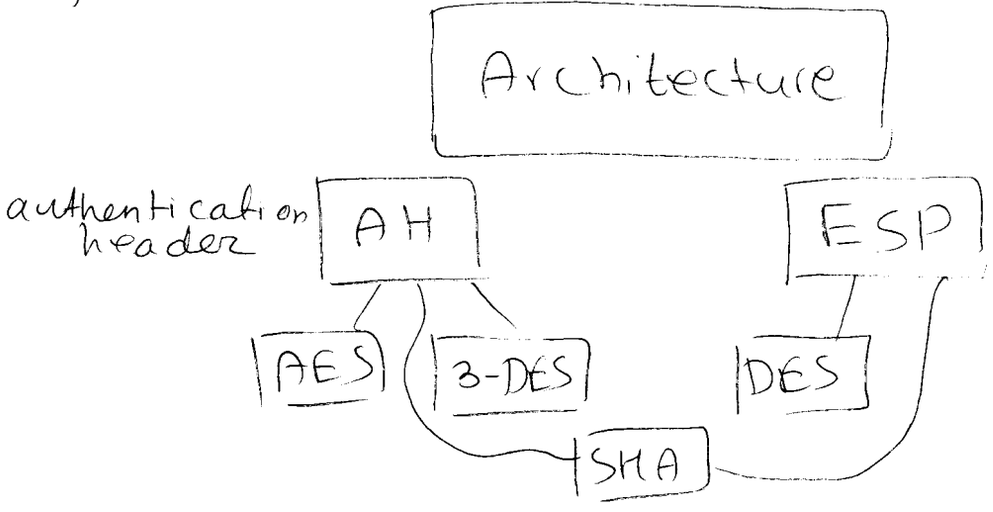
- תשב כיווני - השמות כך שלח זה מקבלים לך מקבלים. אבל שלח רוצה
  לקבל זה מקבלים לשלוח כמה בכיבים לפחות עוזר נוסף
- לא אמין - אין מפיקים הלכה Best effort לשלח אין
  אינפורמציה אם ההבלה הגיעה בהוד אי שיון למקבלים אינפורמציה
  אם תמונה לא הגיעה. מכאן ה- Trasport שמשב שם לפחות
  feedback שבין זה לאחלק מה- IP
- בתמורת IP (ב- V4 יש 32 מיט וכן מלמעלה כי מה לא) מספיק שם
  המשתמש בעולם). לפעמים הכתובות מסורת עשויות כי למשל מספיקות
  בקשורה נוחות הכתובות IP פיזאולוגי למכונים שונים אחת רחוב-
  יכולה להיות אשייכת למשתמשים שונים.
- סופר הגדה של תכונות לא מופטא (אמהן מיצוי מכלל)
- לן לאוצין/לשתנן/לניתן תכונות IP. אין שום פרוטוקול הגנה נגד
  שמה לא עשו את זה? כי לא היה פרוטוקול מתקדם שאת תשבו
  שיהיה לו שימוש בפועל ת'חת אם לא הוא על פרקים מאז שהיו.

משהוא מ- סף הגיעו להבנה שאי אפשר למשיך עם הפרוטוקול בלא
שום הגנה והקימו לשכה שלמה שאחכה היתה לפתיח פרוטוקול IPsec
(IP Security) עם הדרישות המלו:

- סודיות מלא
- שלמות מידע
- סודיות מידע (את נוכים)

- שמירה על סדר התעלות
- מניעת Replay
- הרשאות גישה Access control
- פרטיות (סטויות) (הצפנת הסטויות כך שאחרים לא יוכלו לשמוע מסוף אחד לשני שאנחנו מתכתבים עם אישהו)
- תאימות מלאה עם IP - (התכוונו עלו על מנת הזדהות ים אם הם מאמינים IPSec, אפשר להשתמש הקלות בדברים שמגדר IPsec.

IPSec יש 2 סוגי מאמרים שמגדירים את הפרוטוקול:



DOI קיבוץ שיטות

IKE internet key exchange

ארכיטקטורה - (מתואם)

- שפת ומנגנון מתקדמים מבוזרת (תנים שקוראים SA (security association) והוא מתקדם בתוכו ע"י דברים כמו למשל אפשרויות הצפנה, מספרים סדוריים של תעלות. מה' בשביל הוא להתקין את כל פיסות.
- IKE אחראי לתיאום מרחוק (לא פיסי) של SA.
- השגרות והמנגנים מתקדמים כמה מאגרים: (אנשים)

SAD - database SA (כדי שנוכל לדעת שהחבר

SPD - Security Policy Database - מדיניות

של איזה תעלות יוכלו, אלו שירותים מתעלמים, להצפין?

ב- IPsec יש שני Modes של הצפנה:

Trasport Mode : Tunnel Mode.

Trasport Mode

מידע	IP Hdr*	מידע מקודד

* IP Hdr

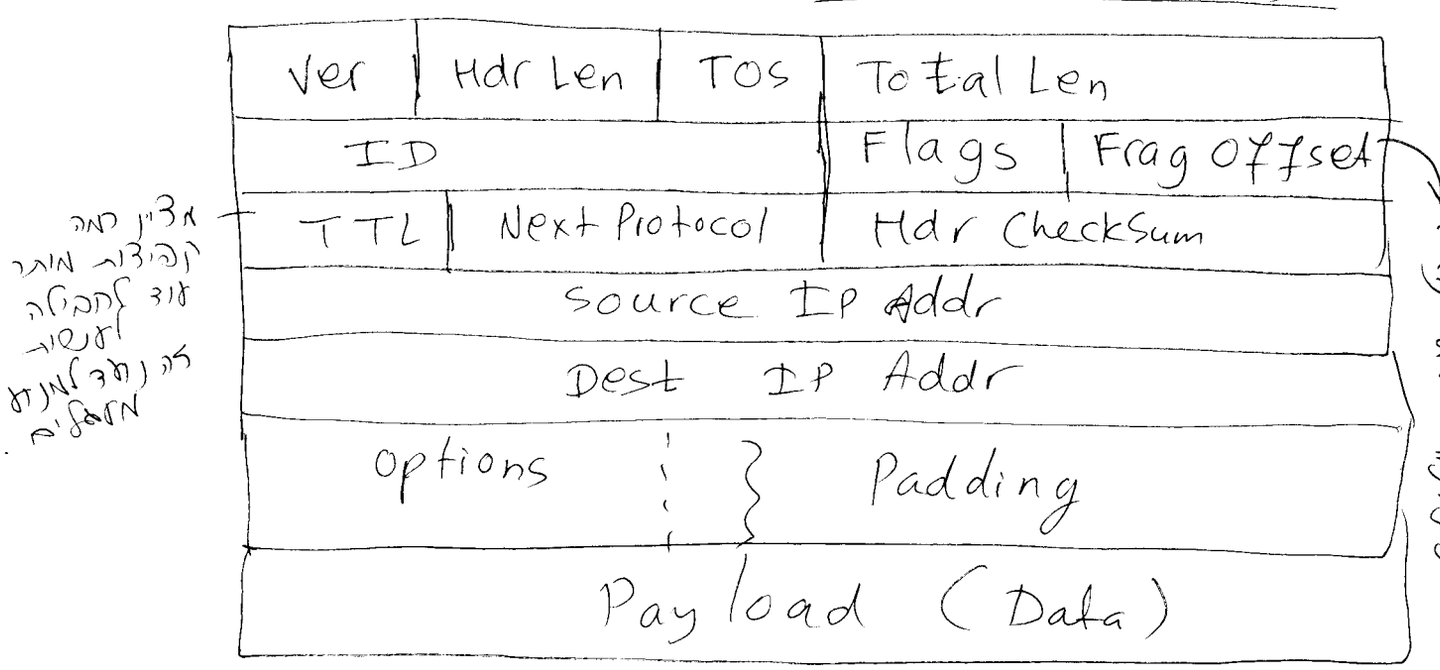
כאן מוסיפים לכאן על ההמולה שמגיעה או יותר כמו שהיא.

Tunnel Mode

מידע	IP Hdr	IP שמשני Hdr

כאן שמים את ההמולה הישנה מתוך תמונה חדשה. עושים זה Source ה- destination של הבית להיות הנתבים, וכמה אפשר עושים פרטיות.

מבנה תמונה IPv4



- AH - פרוטוקול האותות.

לא כאן יש שני אופנים.

מקורי

payload	IP hdr

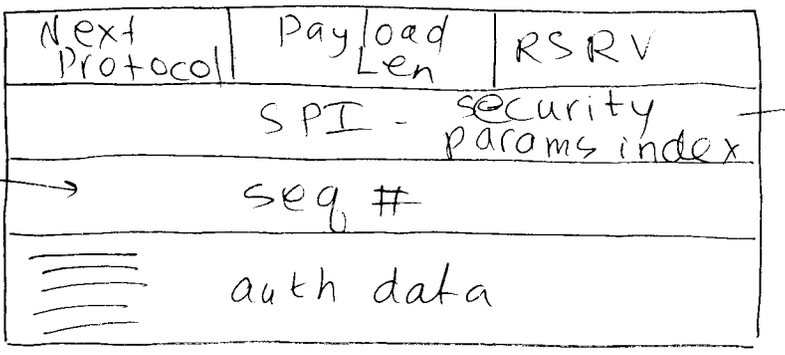
Trans

payload	AH hdr	IP* hdr

Tunnel

payload	IP hdr	AH hdr	new IP hdr

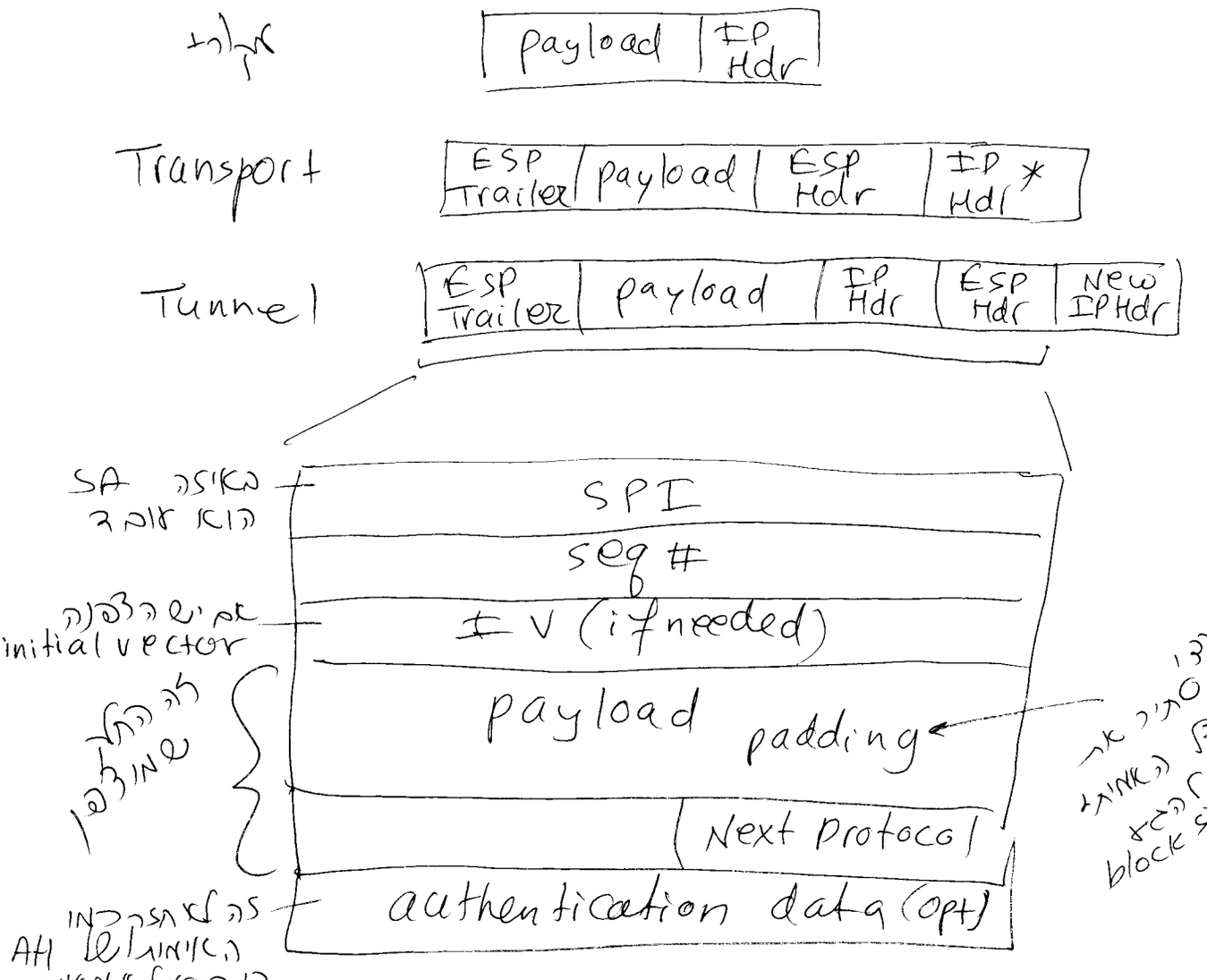
מה יש ב- AH hdr ?



כאן ה- Next protocol הוא בדרך כלל TCP אלא AH. כשיש AH מסווים את פרטי ה- TCP כי עוטפים IP (אם זה tunnel mode).

ה- AH מאמת את ה- payload וגם את חלק מה- IP Hdr. הוא מאמת את כל מה שלא משתנה בדרך.

Encapsulated Security Payload - ESP



יאבי שמוץ הכאו ע"ל קלקה באתר אחרי התרגול ...

אנו מאמינו את לה משקיר שלפני אנחנו IPsec הן תמורות IP
עם פחד, שהרי אחרי הצפנת תאימות אפשר עם IP לאקדה שהתבים
מפתח לכן מפחדים IPsec. הם רק מסתכלים על ה IP header
ומעל לכן מעניין אותם מה יש בפנים.

מהנה הנתונים SA שאותו מתאמים ע"י IKE

- SPI - בפ מיס ניקבע ע"י הנמען - רפי שמוצ יוע לפחות את התמורה)
- כתובת מקור
- כתובת יעד
- פרוטוקול
- תבורת פעולה - Tunnel / Trans
- אלגוריתם + מפתחות הצפנה ו/או אימות
- מספר סידורי בלים הלה - פ פעם שהשולח שולח תמורה הוא מקדם אותו.
- תלון Anti-Replay
- Lifetime - כמן עד התפוגה או
  - כמות תמורות עד התפוגה.

מנגנון מניעת Replay

השולח לכל חושש משום פחד והוא עומד בסדר עפ"י הסדר. מי
שמוצא הוא המקבל.

1	2	3	...	14	15	16	17	...	84	64	08	...

חלון של 64 שמוצא מספר תמורה עדכני

תמורות שהתקבלו.

2) אם i מתאים התהליך –
   - אם כבר התקבלה – מתעלמים
   - אם עוד לא התקבלה – מעדכנים את הביטוי AH או ESP ובודקים אם ההודעה תקינה. אם היא תקינה מקדמים את החלון ומאתרים שקיבלנו. אחרת, פותחים את ההודעה (באופן סמלי ב- log שקיבלנו ומשהו היה מלוכלך מה, אפשר לא לכתוב) כמובן בודקים גם שברים מכיוון
3) אם i מימין לחלון – מעדכנים HA / ESP
   - אם לא תקין – פותחים
   - אם תקין – מקדמים ומציגים את החלון ימינה במספר הביטים המינימלי כך שיהיה התקבל ביט משמאל.

לאחר כך יהיה סיכוי Replay ? לא בהכרח – כי ברגע שפתחנו את ההודעה ובדקנו תקינות היא אל תתקבל יותר שוב.

מתוארת ואיך חוקף קבוע לא בהכרח יותר יעיל. לאחר ההתחברות מיוחד קבוע ופתאומי? בהכרח פעמים מאמשים את לא

DAS – כי טעמי של VPN-ים. יש אחד ים רשות ואם למקבל אפשר בכמה לכמות מעט לכם יש אחד רשות ואחד מקבל יכולים לא רשות או פיסי או בלי לרשת IKE. את התאום של ה- SA בדיוק

SPI	SA
1	
7	
⋮	⋮

SPD - טבלה מיוחדת שמשתמשים בתיקי הנתונים.

src	dest	selectors s-port	d-port	---	rule *SA (pointer)

בפועל היא מתפקדת כמו מסוגת כאן firewall.

התוקים האפשריים הם drop, pass או apply.
pass לא אומר שמעבירים את הנתונים כמו שהיו.
drop לא אומר שאסור לעבור אלא אחרת.
apply לא אומר שצריך להפעיל IPSec על התעבורה עם ה SA שיש לנו פונקציה אחת.

רשומת תנועה שקיים לגביה צורכים על ידי הטבלה ורואים אם כתוב מה צריך לעשות - מה היחידות שהיה עליהם ההודעה הבאה. בפ"כ הנתונה הטבלה אומרת אי לעשות עם הודעות שלא אחד מהתוקים הקודמים אלו על פניהן. שאלה כזו יכולה להיות אם בכל המקרים שאין התו צריך לעשות עם הודעות שאינן

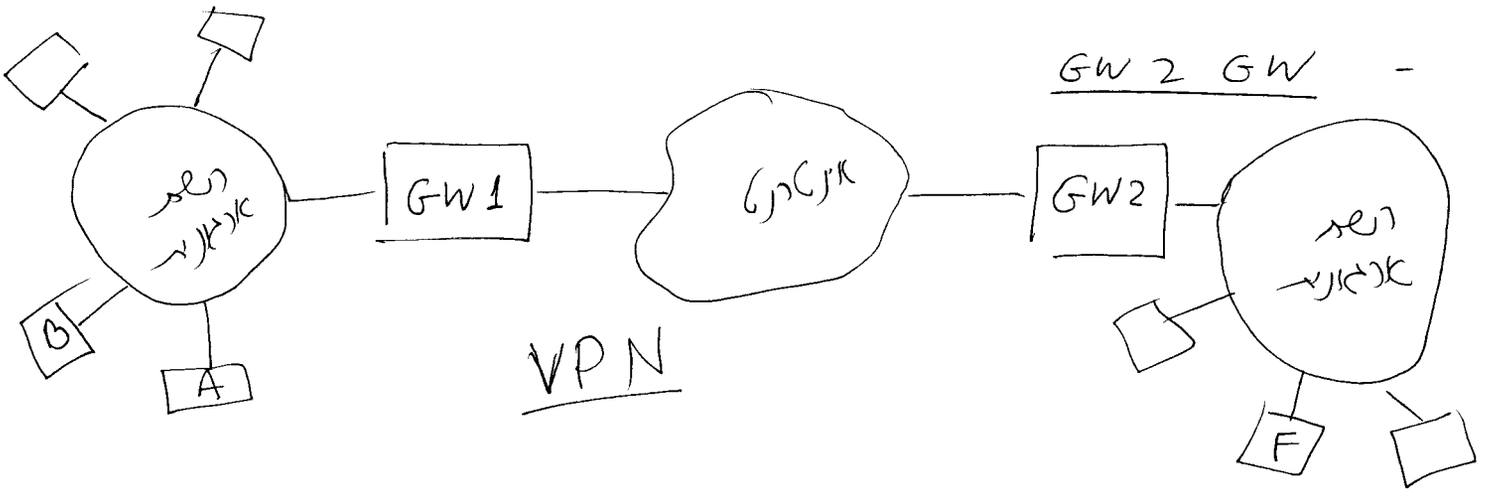
צורות ישום IPSec

P2P -

מקור:

trans mode

זאת יתר אותו הדבר - השרת והמקלט הם כמו המקבילים. נסביר שניים את התעבורה בתוך תעבורה נוספת וכותבים עליה פחות האחרים ניתן ב לעשות tunnel באותה הדרך אבל נאמר, כי



אם A רוצה לשלוח משהו מתוך הרשת הארגונית ל- B ומשם
הוא מצ"ב לכאן אומר לצאת מחוץ לרשת. אין לנו בעיה בהפעלה של IPSec
אותנו מוצאים על הרשת הארגונית.
אולם אם A רוצה לשלוח אי אמון ל- F נרצה לעבור באינטרנט
ובכון מתעוררים תשש, אלו נדרים להפעיל הגנה על התעבורה -
אלימות או הצפנה. אולם אם A ו- F לא מכירים IPSec, מי
ישלח את זה בשבילם? - ה gateway!

data	ip	⇝	IP חדש	esp/AH	data	ip

זה tunnel! ה IP החדש הוא של GW1 והמקלט
הוא GW2. עם התעבורה המקורית. היה מוצאת ע"י הצפנה
או אימות. התעבורה מגיעה מוצאת ל- GW. הוא מפרנת את
המתוכן הפנימי ומכיר כאן שצריך להעביר אותה הלאה. השאלה היא
פרטים מסויימים. יתכן שארגון 1 מאפשר ל- ארגון 2
אולם לא יתכן מי מפיקים מפתח סמ"ת ואם כן אימות או משהו אחר.

אם המדיניות של הארגון היא כזו שכולם מכירים IPSec ועם הזמן היה
צריכה להיות מאומתת. אלא שבתוך הארגון ס הוחלה תלך
מ- mode trans שיש מה רק AH. יהי רש- A שולח ל- F היה
מה שקורה מתחילה. אלא שברגע ש התעבורה מגיעה ל- GW1
תושים לה tunnel ומאמינים ל- GW שאלף את כה ומאז
ל- F. על היה שקוף מבחינת A ו- F.

IKE

פרוטוקול שאחראי על ביצוע אבטחה ב IPsec

phase 1 - יצירת SA של ערוץ - מגדיר את הדרך איך

phase 2 - יצירת SA של המידע

שלב 1

- יצירת SA
- יצירת אסמכתא (דיפי הלמן)
- אלגוריתם הצפנה
- הסכמת לזהות
- הערכת תקפות

שלב 2

- יצירת אסמכתא למידע (סימטרי / אסימטרי)

21.1.08
הפסקה

אבטחה ברמת Transport

סוגי אבטחת האפליקציה:

- אבטחת שיחה (TCP)
- פ - ביוונית (- מניעת) אבטחה ברמת IP!)
- אימות מקורות (?)
- שלמות תוכן
- אימות תוכן
- שאיבה של סדר / אי משלוח חוצה
- שליחה של תעבורת אפליקציה

רוב הפתרונות האלה קיימים ב- IP.

המודל של SSL הוא כזה:

הלקוח מתוניין לבצע רכישה או להעביר כספים. תמיכת Netscape הביעה שיש צורך באבטחה (לא היה אז IPsec). הפניה הראשונה של הלקוח לשרת לא מחייבת אבטחה, אלא רק מאוחר יותר שיש צורך באבטחה.

מבחינה טכנית יש איבשהו תוכן סטטיות - שרתים מקדימים אלפי לקוחות ונותנים אותו מוגנים. לעומת זאת יש בד"כ איבשהו סימאות אלפי השרתים בד"כ שיכים לחברות ענקיות ושם כבר יש מה לפגוע עם certificates. את אימות ההזדהות מבצע בד"כ סימטרי. או ה שרת אפשר לאמת קריפטוגרפית והלקוח לבן אנוכית את הזהות של שהוא אמיתי נכנס לחשמון שלו. אלפי משלם הלוה כבר פחות ערוץ מאומת את הלקוח ירוץ לחברת זאת ה סימאות שלו לפעול תשם.

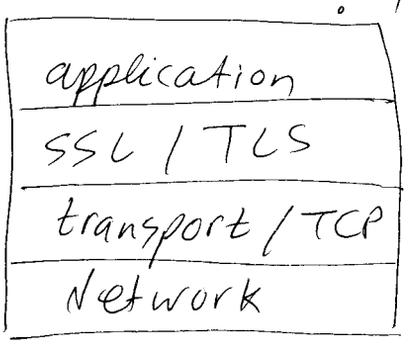
את האימות של השרת תובה משמעות SSL ואולי את הלקוח עבר תיבים לאומת. נניח היתוך כי הלקוח לא רוצה להסתיר את

SSL - Secure Socket Layer (Netscape)

TLS - Transport Layer Security (ietf)

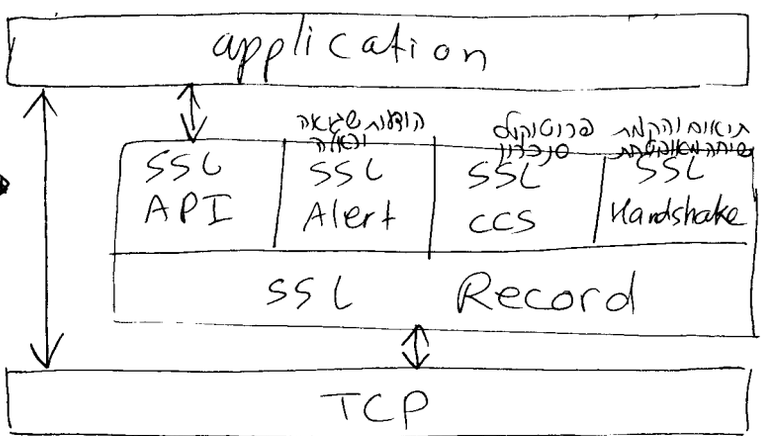
TLS V1.0 הוא מאוד דומה מבחינה ארכיטקטונית, אלא הן שונות מימוש כך אין קומפטיביליות.

אפשר לבנות את SSL או TLS?



כאשר שיכתב מחדש שנותן אוטומטית נוסף ל- sockets

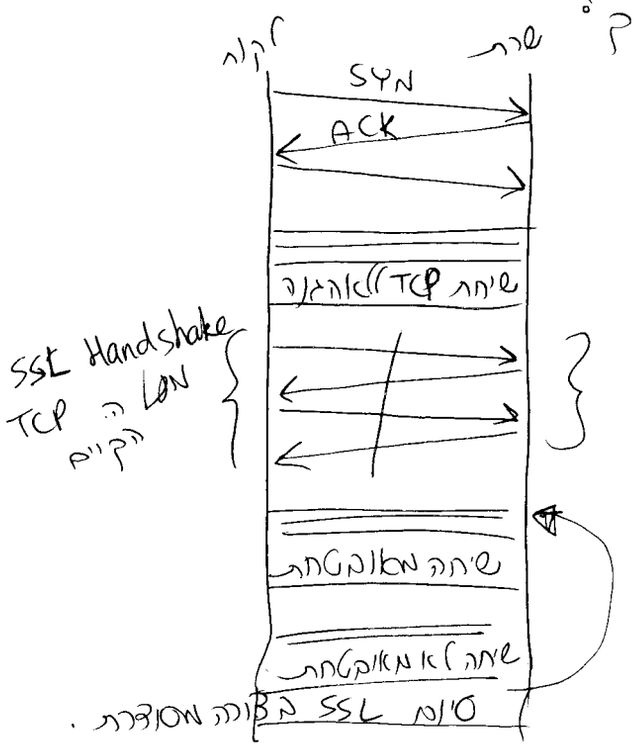
הארכיטקטורה של SSL



לאחר ש תיב לעבוד עם SSL.

אך ישנן תלכים אך אפשר בדיוק יכולים להשתמש ישירות ב API של TCP.

מה קורה מהרגע שאחר הלקוח שולח בקשה?



SSL Record – פרוטוקול שאורכב מעל TCP

appl | app data

fragment | בית | בית | ... | בית | בית | בית

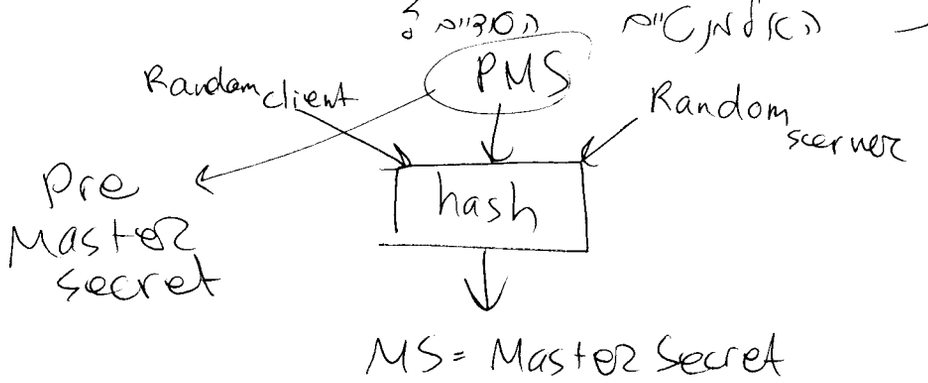
אז בקצרה ההבדל מתקיים לשניים: session ו-connection
session יכול להיות אוסף מתוכו נוצרים connections.

- session
  - session ID (32 Byte) - מזהה ייחודי שהשרת בחר כדי לזהות את ה session
  - peer certificate - במידה והשרת מאמת הצדדה לקוח אחרת לפחות אם הפוך.
  - compression algorithm - כלומר האלגוריתם בו משתמשים בדחיסה
  - cipher spec (enc, mac) - אלגוריתם ההצפנה וה MAC
  - IS_RESUMABLE (flag) - בקשה אם ה session הנוכחי אפשר לחדש או שמוחק מיד ב השרת!
  - MASTER SECRET (48 Byte) - המפתח הסודי שנוצר בהדדיות ושאר האלגוריתם.

- connection
  - session ID (32 byte)
  - server & client seq #'s
  - server & client Random #'s
  - Cryptography keys:
    - ENC_WRITE_KEY
    - ENC_READ_KEY
    - MAC_WRITE_KEY
    - MAC_READ_KEY
    - IV_WRITE
    - IV_READ

יצירת המפתח - session משתמשים בקריפטוגרפיה סימטרית ומשתמש ה connection משתמשים בקריפטוגרפיה סימטרית, כיצד לזה יתכן קל.

איך מייצרים את המפתח הסודי?



(72)

עפ"י Connection יוצרים משפרים אקראיים חדשים.

מפיכך השפותיות Session אל יהיו מה שאחר הפתיחה פותחים
אלא את ה Connection הראשון ושם המשפרים הסימטריים
ישאירו אותם משפרים. אעפ"י אם אותך יוצרים Connection
חדש אל פרים משפרים אקראיים חדשים, שאפשרות יהיו שונים.

איפוה ה SMS? יש רמה אפשרויות:

- שמנוע ע"י DH או משהו שרקוח ניתן ומשאו שתשרת ניתן
- הצפנה מוצפנת ע"י RSA
- שימוש מתומרה משני הצדדים (מעיקר בשימוש לצמח)

(#3) 23.1.08 הפקולטה

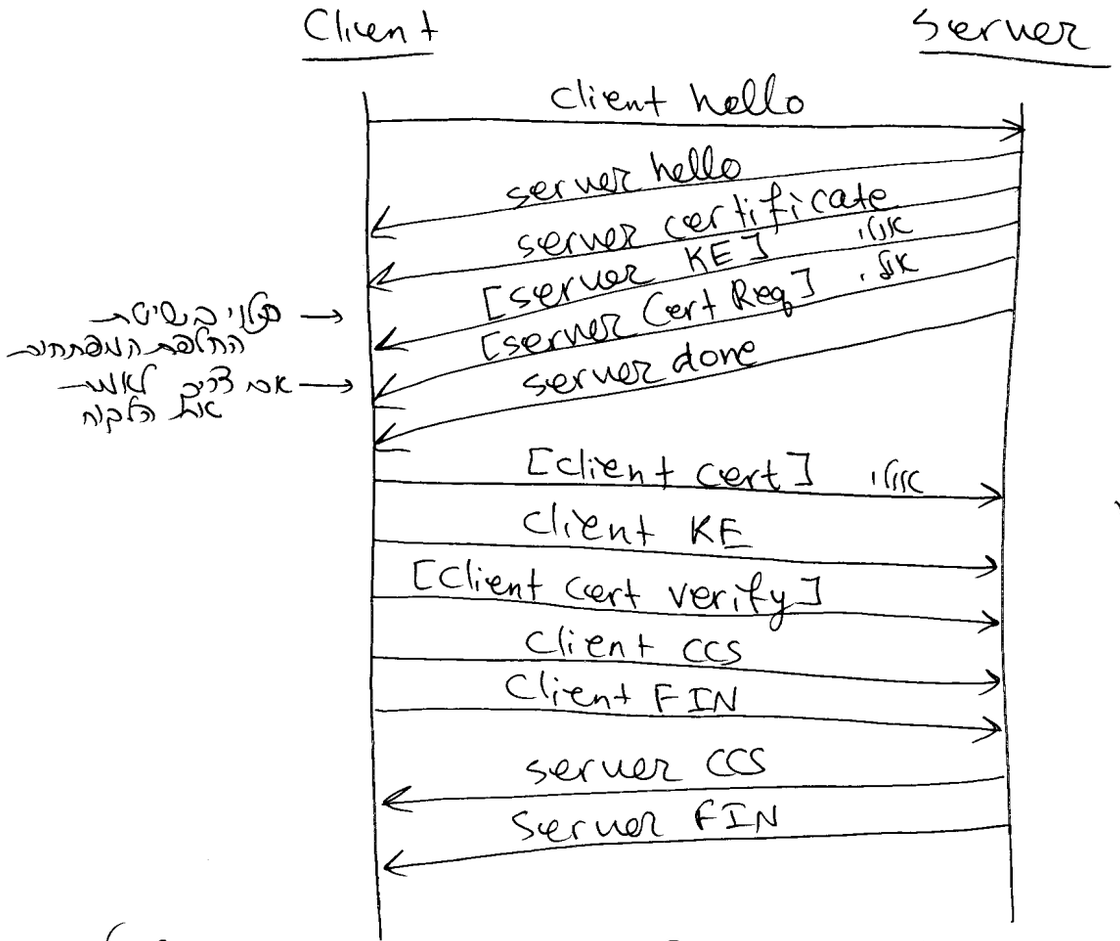
שיעור תרגיל בנושא שני לפני המבחן

SSL Handshake

- מטרות הפרוטוקול:
  - אימות הלקוח - אימות השרת
  - הצפנה - יצירת מפתח שיתוף
  - יצירת session קיים - תיעוד session קיים

- מה מבדיל SH מתקיים של SSL - הכרת אלגוריתמים

הפרוטוקול:



מתבצע יש סדר!

לפרוטוקול יש כמה, אחת אשר לאורך רוב הזמן עוד איתך אולם מה נראה שם זה ההתחלה שבהם.

פירוט:

① Client Hello - שולח R_c מספר רנדומלי
{ ID, version, algorithms, R_c } של ID של session הנוכחי.

② Server Hello - שולח R_s
{ ID.ver, alg(s), R_s }

ניתן לראות מהרשימה הבאה שמניה נשיג הבדים. השרת אומר ללקוח מה יש לו. אם לא נמצא מפתח אלגוריתם ניתן לעשות abort.

אפשר שם הלקוח הנשלח.

④ ClientCert – אופציונלי שבו מקשרת השרת אם יש כאלה.
אם אין השרת רגיל להתעלם ואז שרת – להפסיק או להמשיך.
אם מתקבל נאמן קופסא – השרת לא יוכל להיות בטוח שמדובר
ישנן מתקפות על הצפנה ואומת אלגוריתמים.

⑤ Replay – רשות שלקוח אופציה של השרת.
סוג של – הוא הודעות שכולם רואים ואישהו יכול לעשות
– הוא כי אלה הודעות שכולם רואים ואישהו יכול לעשות
משהו בזה הלקוח לא יכול להיות בטוח שמדובר בשרת
את מי אנחנו לא מתגיד. זה כשמול מסכים.

⑥ ServerHello – יפי, שלקוח יכל שהשרת סיים לשלוח
את כל מה שיש לו להגיד. זה כשמול מסכים.

⑦ ServerCertReq – הישות שלח לקוח רשימה של CA
-ים שהוא מכיר. זה לא קליינטיבילי. אפל אפשר לקנפג
את בכל שהוא הלקוח, בצורך להודיע את להתעלם.

⑧ ServerHelloDone – יפי, שלקוח יכל שהשרת סיים לשלוח
את כל מה שיש לו להגיד. זה כשמול מסכים.

⑨ ClientKeyExchange – הלקוח שולח לשרת את המפתח הסימטרי
שהוא בחר. אם בחרו ב DH אז השרת שולח $g^x \pmod p$ שלו. אם זה RSA אז אין מה לשלוח
וההודעה לא קיימת בכלל.

⑩ ServerKeyExchange – אם לא DH אז השרת שולח
את כל ה CA-ים שהוא מכיר (אלו הלקוח לא מכיר
תלוי ע"י אתר ה- CA שמשתמש בפרטיכן.
לפעמים השרת שולח רמה אחרונה (אלו הלקוח לא מכיר
אופציונלי. לך לרשום אין שום פגם בקיפוליכך.
לה התיאום של ההקשה והמפתח האלגוריתמים + התעם מספרים
אם הוא לא מאמן מול אופציה אין שיהה.
השרת שולח הלקוה טוויטה את מאמן הרשימה של הלקוח שפיא
ולפי סדר הכי גמוהה של הלקוח שהוא לפי מוכן להתאמן בה.
הכי טובה. מתחילת.

אופציה, והתעם מפתחות אסמ) . יכול להיות אם שהלקוח ישלח רמה
(אולי)

⑧ Client KE: אם כבר DH אז הלקוח שולח את $g^y \pmod p$ שלו. ← מספר אקראי מומלץ מיוצר מחדש

אם זה RSA הוא שולח $Encrypt(PMS)$ מוצפן מראש ע"י המפתח הציבורי של השרת שמגיע מהתעודה שהתווה ע"י איזה CA שהוא סומך עליו.

⑨ Client Cert Verify: הלקוח מוכיח את זהותו מול השרת. הוא שולח חתימה על ערכים קודמים (- לאחר אקראיים) מראש המפתח הציבורי שהגיעו למפתח הציבורי מהתעודה שהוא שלח קודם.
- ברשות הלקוח הבית את כאלו או השרת מטעם עם או הוא מפתח.

⑩ CCC Client: ברשות הלקוח רוצה לעבור לסוויטה המוצפרת מהסוכן מראש ל- ID - כי הוא יכול יודע את ה המידע ההצוות, - ע המפתחות ה נתונים.

⑪ FIN Client: שולח (ערכים קודמים) MAC_* כדי שהוא סיים.

⑫ CCC Server: שולח ID - אומר לו שיש איתו

⑬ FIN Server: (ערכים קודמים) MAC_*

הפרוטוקול הזה הוא מידע שאפשר לקלקל. לאחר אתקוף יכול לגרום שהלקוח מעניין כאילו 1.3 אין השרת אתן לזה. אופן האתקוף יכול להתבזר בלקוח 2.5. אם הלקוח מוכן לדווח שיש פה פגם. אז לאחר אחד הערכים הקודמים לא יהיה הגיוני. אחד הדברים שיגרמו סוויטה הצפרנית מוסיפה שפשוט מעוניינים בהם. ע כל נקע בשלב האומת.

מעל הפרוטוקול שני הצפנים יקרוס שהערכים שיש להם הם הערכים הנכונים.

מלוסת האינקריפציה היא מהפשים בדיוק אם RSA ומפ'ג ומלקים האופציונליים. אז השרת בעל יודע שאנו אפשר עם הלקוח השני.

בדרות הפרקים 5H

- RSA
- DH עם אימות
- DH ללא אימות
- חידוש session

RSA - ההודעות הן:

1) Client Hello
2) Server Hello, Server Cert, Server Done
3) Client KE, Client CCS, Client FIN
4) Server CCS, Server FIN

שימו לב שאם הלקוח וגם השרת שלחו לאקראיות - בעלי R_S ו-R_C. הלקוח יצא ונוכל לחשב מספרים אקראיים יחד מהם גזרנו. השרת בודק ונוכל אחר"כ לדעת את כל מבנה מפתחנו לקוחות. כיון "פרוטוקול בינוי" - אפשר לעקוב אחרי המספרים האקראיים" ויש מניעה בלבד.
כאן סיבתו הראשונה של SSS!

DH עם אימות ההודעות הן

1) Client hello
2) Server hello, Server Cert, Server KE (g^x), Server done
3) Client KE (g^y), Client CCS, Client FIN
4) Server CCS, Server FIN

כאן ניתן לראות בבר שהשרת שולח g^x והמפתח הפרטי שלו. זה התנופה שהשרת ישלח כל פעם שיש לו את המפתח הציבורי

שמאיים לשפוך התיזה של השרת, וכך את g^x
הוא יכול לשלוח פעם נוספת אלא עם התיזה.

DH עם אימות ההידקרות הן:

1) Client Hello
2) Server Hello, Server Cert, Server KE (g^x), Server Cert Req, Server Done
3) Client Cert, Client KE (g^y מאומת), Client Cert Verify, Client CCS, client FIN
4) Server CCS, Server FIN

כאן אפשר לעשות Replay בשליחת ה- g^y המאומת אלא שלא יעזור כי מתישהו שני הצדדים יגלו לפחות ש- g^{xy} וכך המתקיף "יתקע". לא יקבל מתישהו קריפטוגרפיה סוף הפרוטוקול.

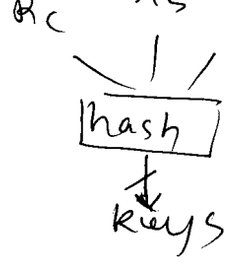
תיבוש session

יש פה פעולות קריפטוגרפיות לכל אחד. אלה מחירים יקרים. נלקחת מלא ורשות מאוד. אנו רוצים לאמץ את הקריפטוגרפיה לאחר יצירת session תיבוש 6 פרק אלא רק לתוסף לו connections.

ההידקרות מאפשרת לה ת"ב:

Client Hello (ID) ⟶
Server Hello (ID)
client CCS
client FIN
Server CCS
Server FIN

לה שהי ID קיים ואם השרת מוצאים אותו משמש הם ב- M_S, R_S, R_C לפיהם מפתחות חדשים



תודה מראש

משתמש יקבע את מהדורת שפת הקוד.

שיעור תכנה - יום שני 2/11 (להגיש תחילת 9!)

יקבעו את מהדורת שפת הקוד (גרסת הפרוטוקולים)
שבה הכתובת יהיו יקבעו את ההרשאות
(התשובה) (מתשובה)

יש שעורי יתר בקורס ושעורי זוגי בקשת

המתן - כתבה בו שעורי ה- 3 שעות.

ה- DB שמשרת התוכן שלו.
מהלקוח יגיע הפרוטוקול אחרי. וה CSS משתמשים
- הלקוח מוכן להשתמש ב session שהשרת שמר ואז הם
- הלקוח לא מוכן לכך והשיחה נגמרת.

יש שני, אפשרויות:

ה- CSS הם משתמשים ב- ID שנתתי. אם השרת לא מוכן
לקבוע ישר ל- CSS.
לקוח בו. את השרת הסיום זאת שלקוח מיתש הכל טוב ויפה ואפשר
שמוצא מעניין מן ההשרת שהוא תלוי את ה session שמוצג מוכן
ואפשר לו עובד מאין זאת כך את. הלקוח שהוא לשרת את ה session
מוצ בדיק להריץ את ה הפרוטוקול אחרים.
אם השרת לא מוכן לתפוש את ה session אז היא שולח ללקוח "0"

11.2.08
הכפיה

2004 מועד ג' שאלה 11

הפלט עם ג' הינו שמדפיסים את השם של תחנת העמדה וכך
ג'אשר נותן כח יפות שאפה ההודעה מיועדה. הבעיה שם
התנה ראשון תמיד אורך אפס המטרה היא לעשות פרוטוקולים
מינימליסטיים אלא לא נדרש להוסיף את A.

2006 מועד ג' שאלה 12

א - אפשר HA מתיק HA וכן הלאה

ב - עקרונית אפשר להשתמש ב- MG-2-MG ב- trasport
אבל אז לא יהיה קצף וה- MG לא משתמש ב-MG

ג - לא מוסיפים לו את ה checksum כי היו לו קורץ
- רק אם יש אימות אפשר לדעת שלא שונה אבל כאן אין
אימות אז אם מצפינים את ה #sequence ההודעה
תתקבל וההודעה תיבדק

ד - יכול להיות אותו SPI כמה פרוטוקולים אפשר לעשות זה מקלות
שנים. (ה- SA נקבע ע"י ה שלוח, והמקלט זה SPI).

ו - סוג ההצפנה מוגדר ב- SA. אלא ב SPD.

2004 מועד ב' שאלה 8

א - לא חושף סוד כי hash הוא one-way

ב - יש לא חושף סוד כי קשה להפוך את הסיסמאות

ג - ה salt לא מוסיף לאמיתות כי כבר ברגע השרת וה מפתח
קדר את אפשר שפשוט חיפוש אחלה

ד - בשרת יש נוכחות מפתח סימטרי בין הלקוחות, לכן למן ואם הצליחת.
את הפרוטוקול נוכחה יכול לעודד סיסמאות.

ו - אם יש רשימה על כל אחד היא נוכחה אם בשרת אחד הפרוטוקול
עכשיו יצא הפ.

2004 מועד ב' שאלה 1

א – לא טוב מאחר ש- a סוד והפיך

ב – לא גם טוב

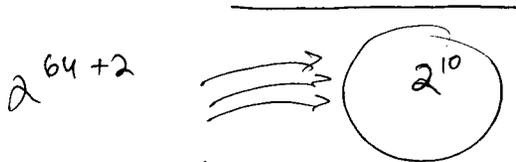
ג – ניתוח אפשר לשלוח את g. ה הפונקציה ידוע בסדר אפס
באיזשהו נקודה כי, אפשר לשלוח את $\frac{x}{2}$ ואת $g+\frac{g-x}{2}$
ולקבל אותה שקול מסדר.

3 – נכון

ה – לא שון

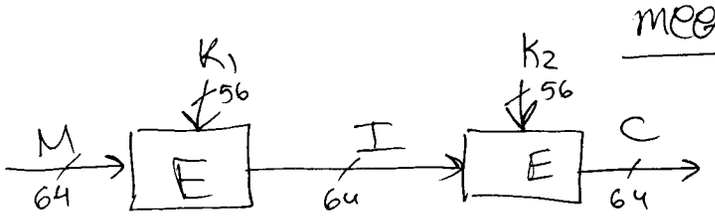
1 – לא שון

2003 מועד ב שאלה 14



$$\frac{2^{56}}{2^{10}} = 2^{46}$$

Meed-in-the-middle



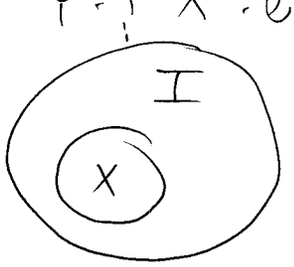
יהיו הנתונים (M, C). נסמנים את ה האפשרויות ל- K_1 ו K_2

$$X_1 = E_{K_1'}(M)$$
$$\vdots$$
$$X_{2^{56}} = E_{K_1^{2^{56}}}(M)$$

ואילו הדבר עבור הצד השני,

$$Y_1 = D_{K_2'}(C)$$
$$\vdots$$
$$Y_{2^{56}} = D_{K_2^{2^{56}}}(C)$$

הטענה היא שמהתאה ש- X ו- Y מופיעים אקראיות יש כ- 2^{48} התנגשויות.



ואם יש ניסיונות לה $|I|$ אם מותרים אקראיות נקודה מפנים $\frac{|I|}{|X|}$.
הסיכוי שתוך מפנים הטו $\frac{|I|}{|X|}$.

2004 מועד ב שאלה 14

א - הצפנה אחרי הקידוד ולפני האות

ב - כנ"ל

ג - שמות הנמענים עבור מאוימתים אלמ אפשר להשמיט אותם

ד - אותו הדבר

ו - מאוימתים את הנמענים ומתעמת עם האפשרות אמלם עבו ההודעה
עדיין על A. למשל אם B מקבל את ההודעה הקודמ הוא
יכול לשמש את ההודעה ש- A מקבל.

2005 מועד א שאלה 9

א - עבור נכון כי שואלים את הקלות מה היחס ביניה

ב - אפשר לזהר שיטה מאולמתת אמלם לה כמאה לא מי שרצינו
לשתת איתו. הרעיה היא שהת גלה יודע את האפתת
הפרטי שמתאים לאפתת הציבורי מאשרה. כעתה לא מקבל
הצנה, שאלו את קלות מה הוא ביקש עת שוב.

ג - עבור נכון כי יש לו את Rc ואת Rs

ד - אם היינו רוצים MD היה לנו לזהר אם Serverke את לה כמאה
נכון

ה - עבור נכון ואם אמרנו את לה מהסבר על הפרוטוקול

ו - ה. MAC מאומת רק את ה- FIN

2006 מועד א שאלה 2

פ = Xor-ב לוקחת 6 ביט ל- 4 ביט. לומר היא מ-
קבוצה בגודל 2^6 לקבוצה בגודל 2^4. יש $(2^4)^{2^6}$ אפשרויות
כזה. אמלם יש $(2^{256})^8 = 2^{2048}$ לפסאות את יוכל
אם יש קפסה את שתאיה באפתה יש 2^{53} אפשרויות מלם האפתת
מטוקם 2^{53}.

2006 מועד א שאלה 1

ג - נכון כי M_1 גדול מ M_2

ב - אפשר לתלק הכל ב-2

ד, ו, ה - לקחת את השורש הפשוט M_1^2

ז + ט - נראה נכון כי השיטות הקודמות פחות יעילות (במקום האומר התופש.)

2004 מועד ב שאלה 10

את מנותח שיש סנכרון שעונים שם אין שימוש בפתן מונע Replay ואם יש שתי תתיאורי פתן יש אימות הדדי.

אם רק כיוון אחד משתמש בפתן אין אותן דך ישתמש את עצמו.

בשיטת אפון עם מספרים סופרים מתואמים.

עם מספרים אקראיים יש מדיניות כי ריק E שמע מתושהו הודעה $(N_A, B, A)E_K(A, B, E_A$ הוא יכול לשלוח אותה ל- B ואז B לא יכול לדעת שזה ולא הוציא בשבר אז פחו והוא לכן באמת יודע עם מי הוכו מדבר.

עם יש מונים יש בעיה כי אפשר לך דור את ההודעה באמצע ושלוח אותך בתוך שעה, אפל פה לכן ממש התקפה...

בקיצור התשובה וגם הנכונה היא עדיפות של פתן וסופרים.