**הגדרת הבעיה של הפרוייקט (רעיון הפרוייקט):**

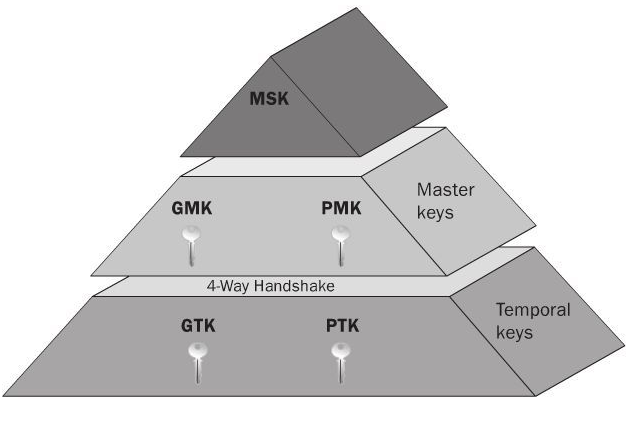
בניית מערכת אבטחה לרשתות ביתיות שמבססות WIFI למניעת מתקפות מסוג Brute Force.

**הבעיה:**

* קל מאוד לדגום את המידע ע"י האזנה לתדרים של רשתות שמבוססות WIFI.(Aircrack-ng)
* היום, מתקפות מסוג Brute Force על רשתות שמבוססות WIFI מתבצעות בשכיחות גבוהה. אחת מהדרכים לבצע מתקפה שכזו, היא בדרך הזו: ה supplicant (client) שולח בקשה להתחברות ל Authenticator (router) ולכן מתחיל תהליך אימות שמתרחש ברשת. כיו"ב האימות אינו מתבצע בנתב ולכן ניתן להאזין לתהליך האימות דרך הרשת. לדגום את המידע של האימות (לדגום את המפתח של הPreshared Key של הרשת WIFI ולהיכנס לתוך הרשת, לצפות במחשבים נוספים, האזנה MITM וסוגים שונים של מתקפות)

תהליך האימות בין הנתב למשתמש מונה 4-שלבים והתהליך כולו נקרא 4 way Hand Shake.

**מושגים וקיצורים של תיאור התהליך: (התהליך מתואר בהמשך הדף)**

**MSK – Master Session** **Key** – מפתח זה נועד בשביל לקיים משא ומתן במשותף בין ה Supplicant לבין הAuthenticator Server.

מפתח זה מועבר באמצעות ערוץ מאובטח מה Authenticator Server לבין הSupplicant.

**PMK- Pairwise Master Key** – הPMK יכול להיגזר מפרוטוקול EAP\* או ישר להיגזר ע"י הPSK (Preshared Key).

הPMK הוא ה256 ביטים הראשונים של הMSK.

EAP\* - EAP is an authentication framework for providing the transport and usage of keying material and parameters generated by EAP methods

**GMK – Group Master Key** – הינו רנדומלי ונוצר ע"י הAuthenticator.

**PTK – Pairwise Transient Key** – הינו ערך שנגזר מהPMK ,Authenticator nonce- Supplicant\_nonce, Authenticator Address, Supplicant Address.

זה למעשה משמש אותנו בשביל להצפין את כל השידורים ליעד בודד בין הclient לAP.

PTK – מכיל 5 מפתחות שונים.

* **KCK – Key Confirmation Key** – משתמשים כדי לספק תקינות נתונים עבור 4 Way Handshake וGroup Key Handshake.
* **KEK – Key Encryption Key** - משומש ע"י הEAPOL Frames בשביל לספק מידע פרטי אודות ה4 Way Handshake וGroup Key Handshake.
* **Temporal Key** – נשתמש בו בשביל להצפין ולפצח את הMSDU של המידע של 802.11 בין הSupplicant לבין הAuthenticator.
* **Temporal MIC**-1
* **Temporal MIC**-2

**GTK – Group Temporal Key –** GTK משמש אותנו להצפין את כל המידע (Broadcast/Multicast) בין הAP לclients.

GTK נגזר מהAuthenticator ונשלח לSupplicant, כאשר הSupplicant מקבל את הGTK הוא שולח תגובה לAP ואומר לו שהוא קיבל את הGTK ויישם אותם ומוכן להקמת תקשורת מוצפנת.

ניתן לראות את התהליך את ה4-Way Handshake בסרטון הבא :

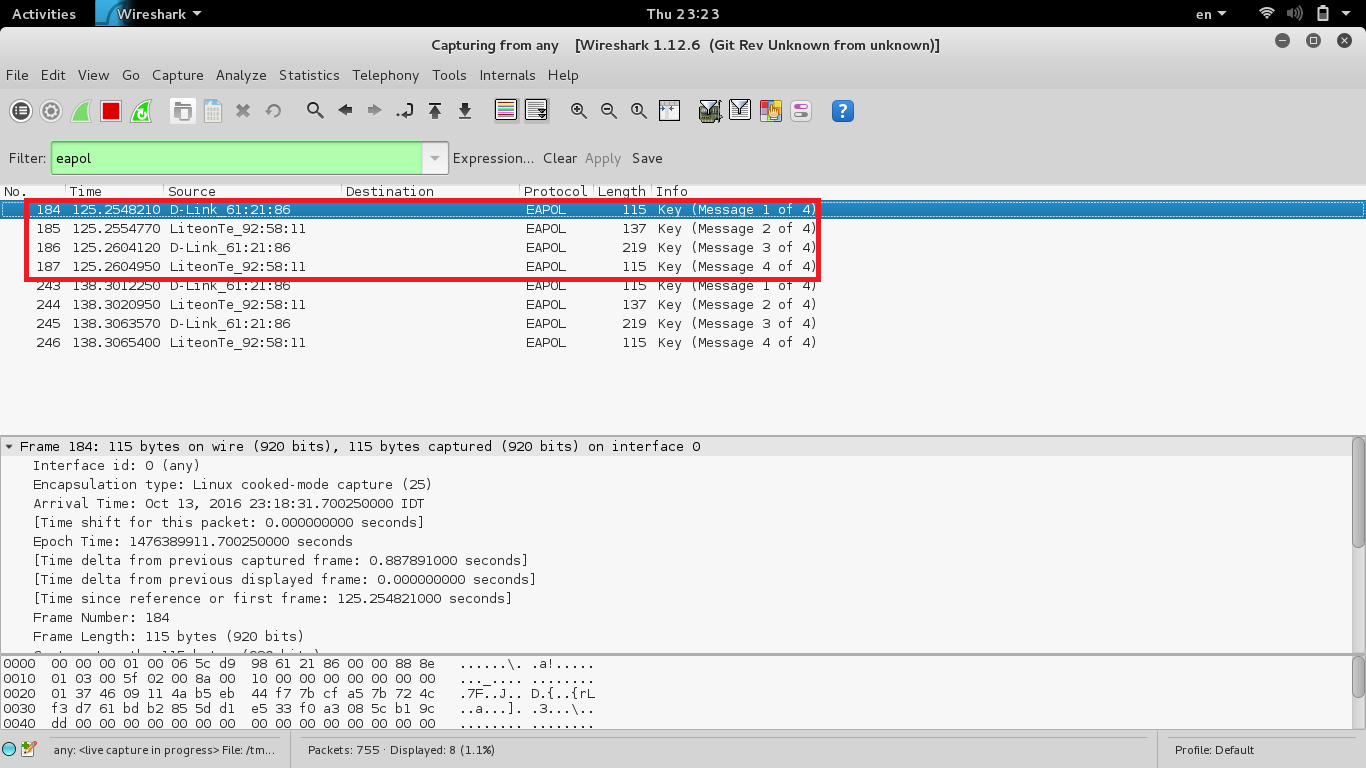
[www.youtube.com/watch?v=9M8kVYFhMDw](http://www.youtube.com/watch?v=9M8kVYFhMDw)

**תיאור התהליך של 4-Way Handshake:**

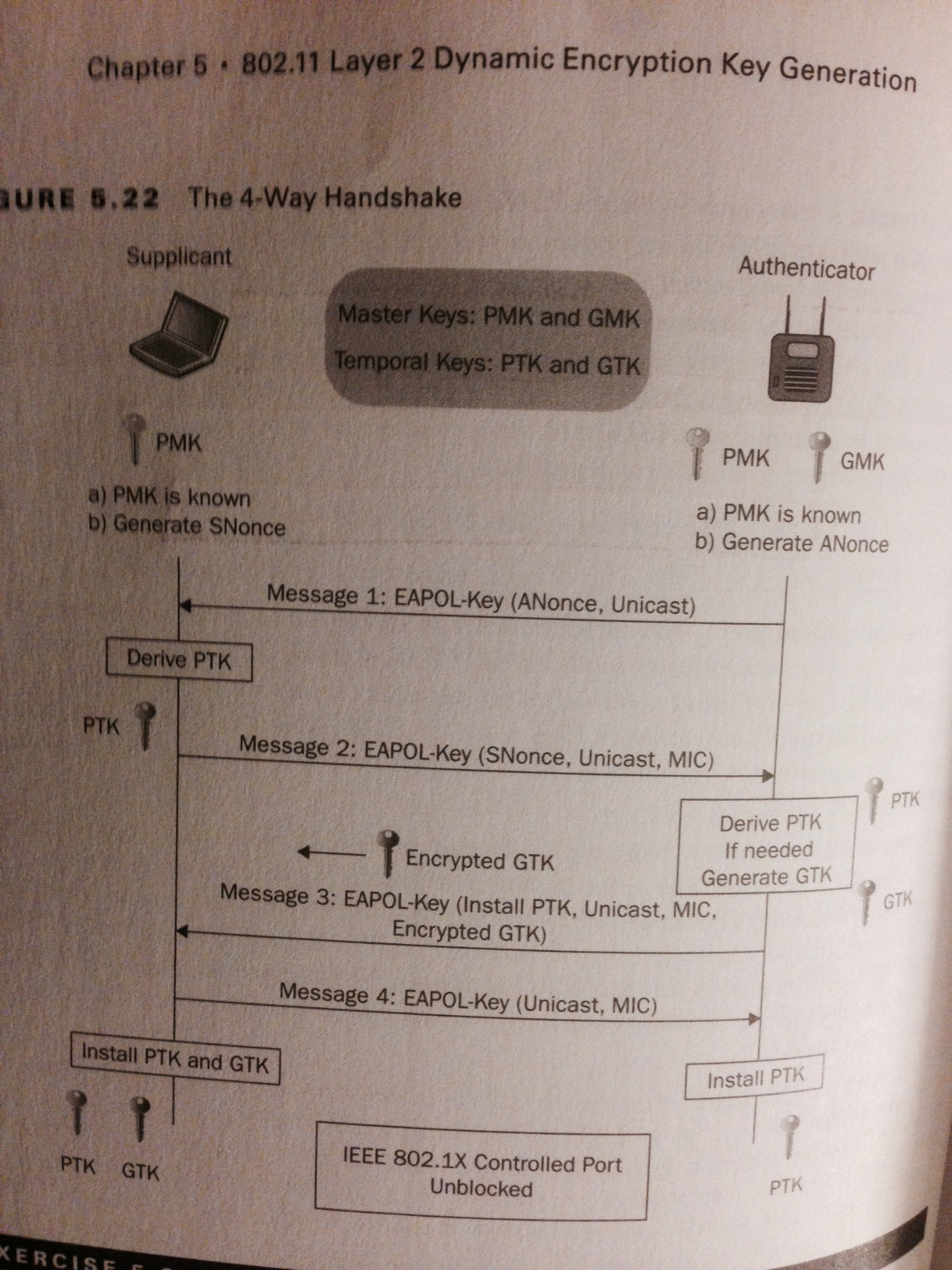
ה 4-Way Handshake משתמש ב EAPOL-Key frames מאותחלים ע"י הAuthenticator server (הנתב) בשביל:

1. לאשר שה (client) peer מחזיק בPMK.
2. לאשר שהPMK הוא נכון.
3. לגזור PTK חדש מהPMK.
4. להתקין את pairwise encryption.
5. להכניס את המפתחות לתוך 802.11.
6. לדאוג לתעבורת המידע של הGTK בין הSupplicant לבין הAP.

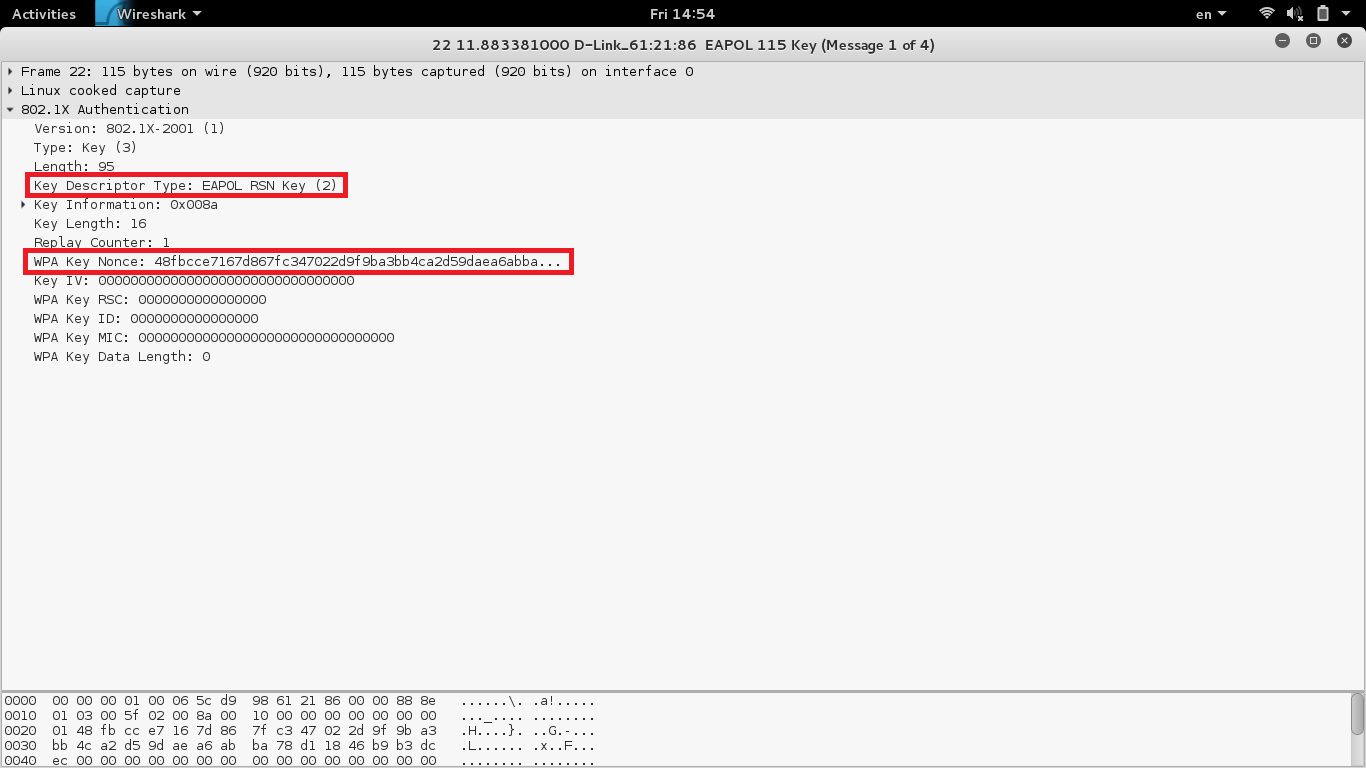
ניתן לראות את תהליך ה4-Way Handshake בWireshark בתמונה הבאה:

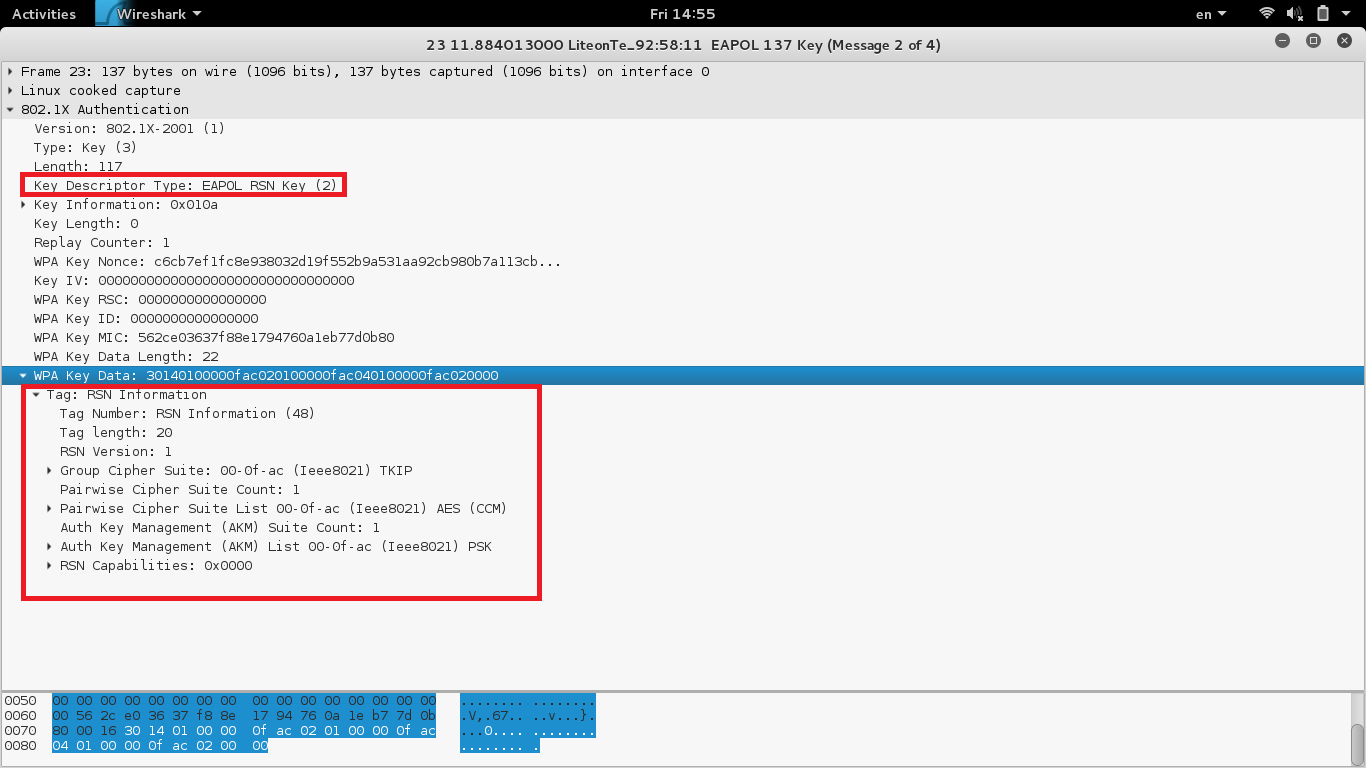
****

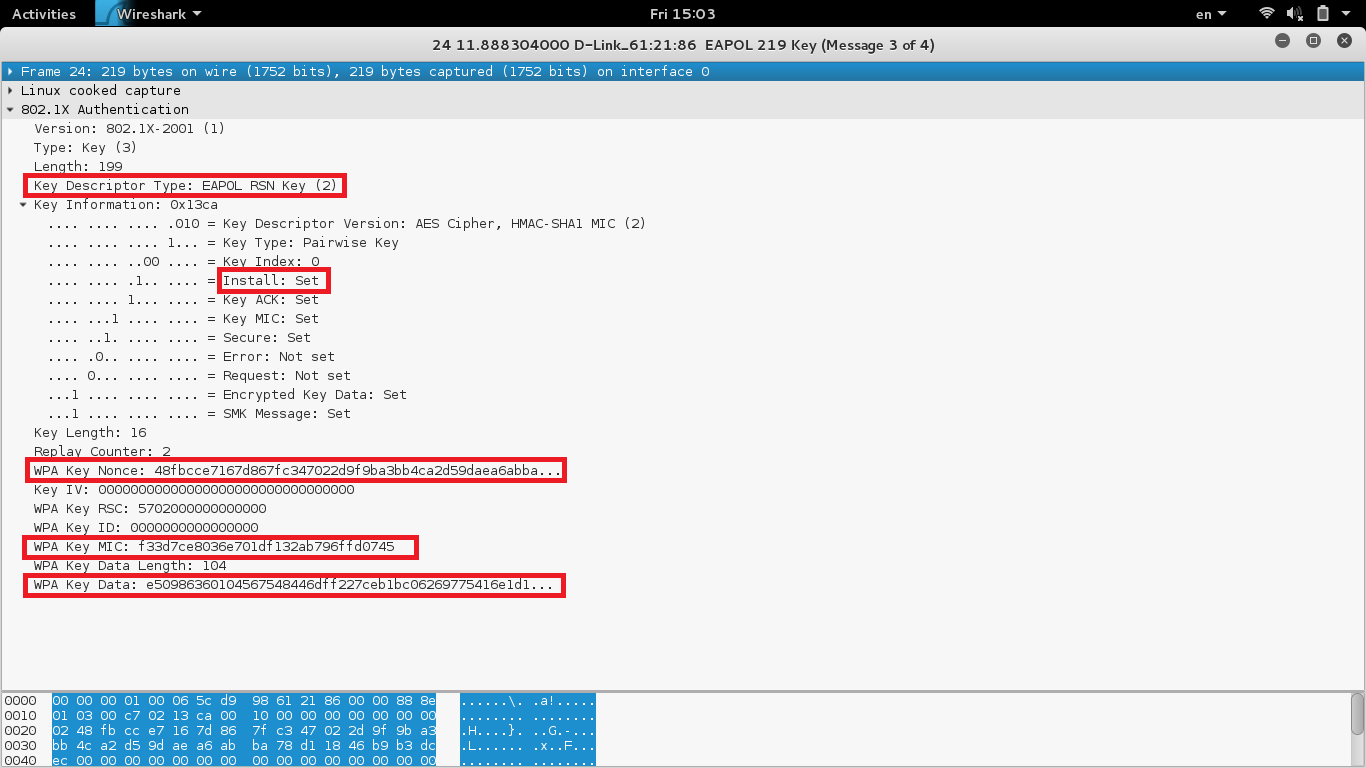
**בתמונה הזאת מתואר כל צעד בתהליך:[[1]](#footnote-1)**



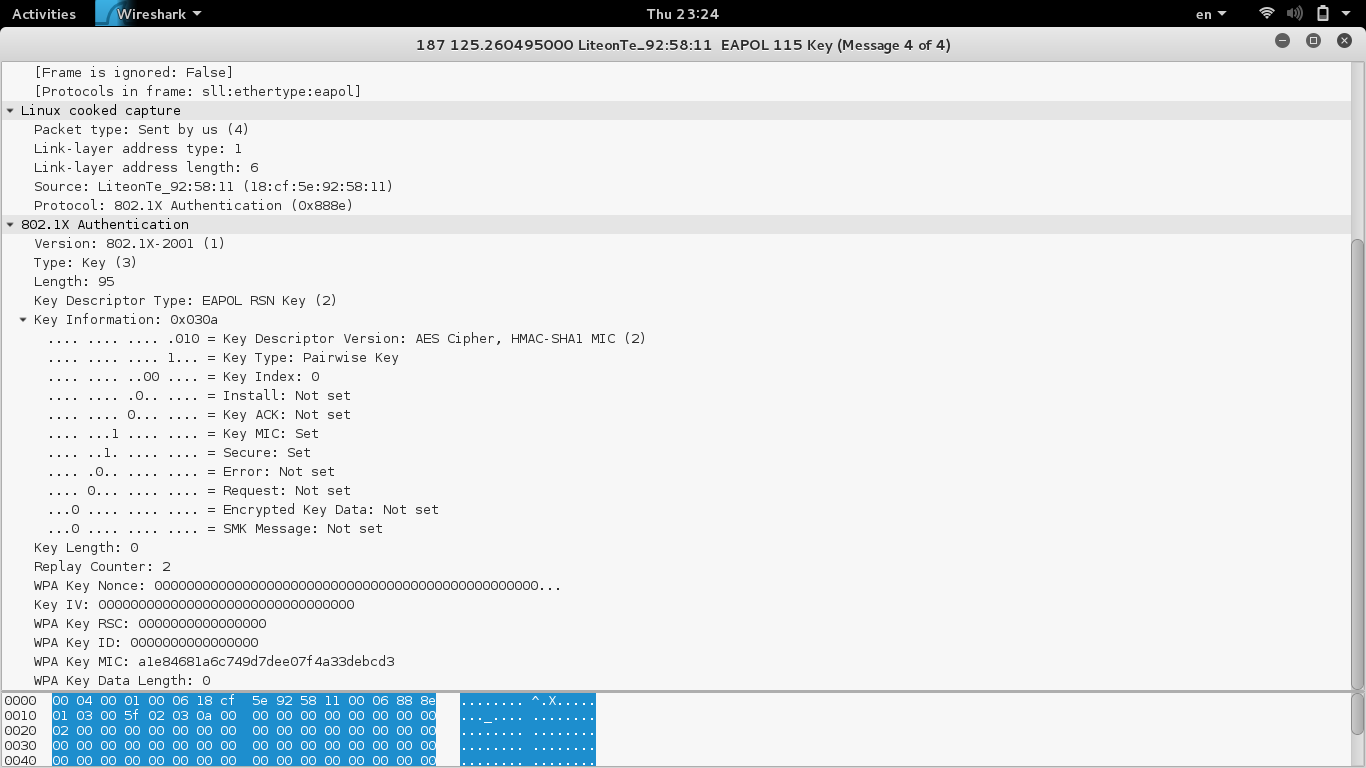
**התהליך של ה4-Way Handshake דרך כניסה לframes דרך הWireshark :**

בהודעה הראשונה ניתן לראות שהAuthenticator שולח את הפריים של הEAPOL KEY שמכיל את הWPA Nonce

בתמונה השנייה ניתן לראות את הSupplicant שולח את הפריים של הEAPOL KEY שמכיל בתוכו את הSNonce לAuthenticator

בהודעה השלישית ניתן לראות שהAuthenticator שולח את הפרייםEAPOL KEY שמכיל בתוכו את ה ANonce, MIC וניתן לראות שהInstall-Set כלומר הוא יישם את הפריים הקודם.

בהודעה הרביעית ניתן לראות שהSupplicant שולח את הפריים האחרון של הEAPOL KEY לAuthenticator בשביל לאשר את המפתחות שהותקנו



**תהליך הכניסה לנתב דרך האזנה לרשת הWIFI:**

1. נצטרך כרטיס רשת שתומך בPacket Injection. (Packet Injection זאת פקטת מידע שנוכל להזריק לרשת ע"י צד שלישי שאינו מחובר לרשת).
2. נצטרך להוריד את הכלי aircrack-ng שבתוכו נמצאים הכלים הבאים:
   * airmon-ng – פותח מוניטור שנוכל לבצע האזנה/השתלה של מידע ברשת
   * airodump-ng - זהו כלי שמבצע פעולת Sniffing שסורק את כל הנתבים והמשתמשים.
   * aireplay-ng- זהו כלי שבעזרתו נשתיל פקטות לנתבים (ע"י צד שלישי)
3. נפתח מוניטור לצורך האזנה והשתלה של הפקטות ע"י airmon-ng לכרטיס רשת המתאים.
4. נאזין לרשתות WIFI שעובר אחד מהם בהמשך נבצע פעולה של 4- Way Hand shake
5. ?? מה כתוב פה
6. נשלח פקטת DeAuth ע"י גורם שלישי לצורך דגימת המפתח של תהליך ה4 Way Hand shake.
7. נפעיל את הכלי aircrack-ng לצורך פענוח המפתח ומציאת הסיסמה של רשת הWIFI.[[2]](#endnote-1)

**פתרון לכתיבת הפרוייקט:**

1. המשך הפרויקט בקורס מערכות מבוזרות שנעשה בשנה שעברה ע"י הוספה ושינוי של פרטים בתוכנית שכתבנו (תכנות ברמת הHigh level)

* אפשרות כתיבה בshell script/python/c

1. תכנות מחדש של הEAPOL /כתיבת פרוטוקול מחדש כי האימות אינו צריך להתבצע דרך הרשת כלומר הסיסמא אינה צריכה להתבזר דרך תקן 802.11, הוא צריך להתבצע ע"י הנתב בלבד (תכנות ברמת הLow Level)

* אפשרות לכתיבה באמצעות dd-wrt

1. בניית ממשק GUI (שם משתמש וסיסמא עבור כל משתמש)

* שמירה כתובות MAC
* הגבלת כמות המשתמשים לפי השם משתמש.
  + - דוגמה user:password - 1 יכיל עד 2 משתמשים עם כתובות MAC **דינמיות**.
    - דוגמה 2 - admin:admin יכיל עד 1 משתמשים עם כתובת MAC **קבועה**.
* שליטה ובקרה על יצירת המשתמשים היא באמצעות אפליקציה שמתחברת לנתב באמצעות SSH(20)/Telnet(23)/HTTP(80)

1. CWSP® Certified Wireless Security Professional Official: Study Guide

   <https://www.safaribooksonline.com/library/view/cwsp-certified-wireless/9780470438916/ch05.html> [↑](#footnote-ref-1)
2. <http://www.hsc.fr/ressources/articles/hakin9_wifi/hakin9_wifi_EN.pdf>

   Short answer is, 4-way handshake password "cracking" works by checking MIC in the 4th frame. That is, it only checks that KCK part of the PTK is correct. 4-way handshake doesn't contain data that would allow checking of other parts of the PTK, but that's actually not needed, for two reasons:

   1. MIC verification is how AP checks the validity of PTK (and, consequently, the password);
   2. Chances of a password producing PTK that has valid KCK but invalid other parts are really low: KCK is 128 bits, so probability of incorrect password producing correct KCK is 2^-128.

   Overall, 4-way password "cracking" works like this:

   1. 4-way handshake is parsed to get SP and STA addresses, AP and STA nonces, and EAPOL payload and MIC from 4th frame;
   2. Candidate password is used to compute PMK;
   3. PTK is computed from PMK, AP and STA addresses and nonces;
   4. KCK from computed PTK is used to compute MIC of the EAPOL payload obtained at step 1;
   5. Computed MIC is compared to the MIC obtained at step 1. If they match then candidate password is reported as correct.

   If you'd like to see actual implementation of the attack, one place to start is [coWPAtty](http://wirelessdefence.org/Contents/coWPAttyMain.htm) sources: they're relatively small, self-contained, and easy to read.

   <https://www.youtube.com/watch?v=-Q_WXeEf8Fw> [↑](#endnote-ref-1)