Where is my AirTag

מאת גפן אלטשולר

הקדמה

תגיות האיתור הגיחו לחיינו בסערה יחסית לאחרונה. רובנו התחלנו להכיר אותן עם שחרור תגיות ה-AirTag של Apple באמצע שנת 2021 - בין אם זה על ידי רכישת אחת התגיות הללו וצירופה למפתחות\מזוודה\חיית המחמד שלנו, או על ידי כתבות שקפצו לנו באתרי החדשות על שימוש זדוני שנעשה בהן. במהלך המאמר נסביר את הקונספט שעומד מאחורי תגיות האיתור הללו - "Offline Finding", כיצד הוא פועל ומה הופך אותו לכל כך מיוחד וחדשני בתחום (רמז: הוא Offline). בנוסף, נכיר מספר תגיות ומכשירים שכבר עושים שימוש ברשתות מהסוג הזה, כאשר נתמקד בתגית החיפוש הפופולארית מכולן כיום AirTag.

משחק מחשבה

נניח והייתם מתבקשים לעצב תגית איתור משלכם. על התגית שלכם לשדר את המיקום שלה, להיות קטנה וקומפקטית, ובעלת סוללה שתספיק לחודשים רבים.

הפתרון הראשון שיעלה לרובנו יהיה תגית שתכיל את הרכיבים הבאים:

- חיישן GPS
- בעלת כרטיס סים או חיישן WIFI

ואכן, החיישנים האלה באמת יעשו את העבודה בשבילינו – באמצעות חיישן ה-GPS התגית תחשב את המיקום שלה, ובאמצעות אותה קישוריות לאינטרנט נשדר את המיקום לבעלי התגית.

ואכן, עד לאחרונה כך נראו מכשירי מעקב קלאסיים, אבל בטח שאי אפשר היה לקרוא להם תגיות. הם היו יותר מכשירים גדולים ומסורבלים לתפעול, בעלי בטרייה גדולה - בכל זאת, חיישנים בזבזניים, ויותר מכל יקרים ולא נגישים לקהל הרחב (אלא אם כן אתם חברת בילוש).

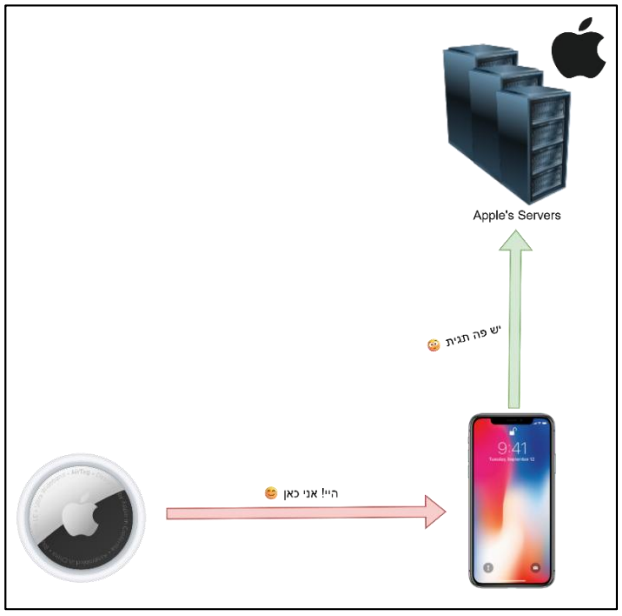
נשמע שלא ממש הצלחנו לעצב **תגית** איתור, בדרכים המקובלות... אם כך מה נעשה?

Offline Finding

אז הבנו שהפתרונות הקלאסיים שנוגעים למיקום וקישוריות לאינטרנט לא ממש יעבדו אם נרצה לעצב תגית אלגנטית, כזאת שתהיה חסכונית בסוללה וגם זולה.

ופה נכנס החידוש הגדול של רשתות offline finding. במקום **שהמכשיר יחשב וישדר את המיקום שלו בעצמו**, **מכשירים אחרים בקרבתו יעשו זאת בשבילו**. בכך אנחנו מייתרים את הצורך של שימוש בחיישנים לאיתור המיקום, ובקישוריות לאינטרנט. וברגע שהורדנו את שני הדברים האלה מהמשוואה – כמות הכוח שנצטרך לספק היא משמעותית יותר קטנה. זה יאפשר לנו להקטין את הבטרייה בצורה דרסטית ולהוריד חיישנים מיותרים, וכתוצאה מכך את המכשיר כולו.

אפשר גם להגדיר את זה בצורה כזאת - הפכנו את תגית האיתור ל-"טיפשה", בכך שהורדנו ממנה את כל החיישנים שצורכים את רוב האנרגיה של הסוללה, והשארנו אותה רק עם התכונה המאוד פרימיטיבית: "להכריז", שהיא נוכחת, כך שכל המכשירים שנמצאים בקרבתה יידעו שהיא שם, ויבצעו איזושהי לוגיקה שתגרום לבעלי התגית לראות את המיקום שלה, לדוגמה – לחשב את המיקום הנוכחי של המכשיר, ולשלוח אותו לשרת.



אך מה זה אומר בדיוק? איך נגרום לאותם מכשירים להריץ את הלוגיקה הזאת בשבילינו? ולמה שהם יעשו את זה בכלל? הרי זאת פעולה שתבזבז להם מהסוללה ומחבילת הגלישה - והם לא חייבים לי שום דבר, בעלי המכשיר כנראה אפילו לא מכירים אותי.

אז התשובה הקצרה: אין להם ברירה. מדובר בתכונה שמגיעה כחלק ממערכת ההפעלה (IOS\Android), וזה מה שמעניק את הכוח ליצרניות הגדולות לייצר רשתות גדולות, בכך שהן "מכריחות" את המשתמשים שלהם להפוך למאתרים בשביל משתמשים אחרים שלהם, בעלי התגית.

אותה תגית איתור עושה שימוש ב**רשת איתור**. אותה רשת איתור אמונה על שליחת מיקומה הנוכחי של התגית, כאשר העבודה מהצד של התגית היא מאוד פשוטה - לשדר שהיא נוכחת. נשמע פשוט לא? אך בשביל לקיים את הפשטות הזאת מהצד של התגית, נצטרך לבנות מנגנון מתוחכם בצד של הרשת, עליו נרחיב בהמשך.

שוק תגיות החכמות

אז כמו שכבר הבנו, תגית האיתור שלנו צריכה להשתמש ברשת איתור מסוימת. היא יכולה להשתמש בזאת של Apple (Find My), שנכון להיום כוללת מעל מיליארד מכשירים, זאת של סמסונג בה מכשירי ה-SmartTags עושים שימוש: רשת ה-SmartThings, עם מעל 300 מיליון מכשירים, או זאת של חברת Tile אשר השיקה תגיות איתור כבר בעשור הקודם (Tile Network) ועל פי החברה מונה מעל ל-40 מיליון מכשירים, או זאת של גוגל שעדיין לא הוציאה תגיות איתור משלה, בה עקרונית כל מכשיר אנדרואיד עם גרסה תומכת אמור להשתתף ברשת, מה שמציב אותה בתור בעלת הפוטנציאל להיות הגדולה ביותר. ישנן אפילו תגיות של חברת Chipolo שעושות שימוש בשתי רשתות איתור שונות: גם זאת של Apple וגם של Samsung!

בחרתי להתמקד ב-AirTag במאמר כי נכון להיום מדובר בתגית האיתור הכי נפוצה, וברשת האיתור הגדולה ביותר. אומנם היא לא הייתה הראשונה שיישמה את הקונספט של Offline Finding, אבל היא בהחלט זאת שהביאה את "הבשורה" של תגיות האיתור.

ניתן גם להסיק לבד שבשביל שתגיות האיתור שלי יוכלו לתקשר בצורה אמינה ורצופה הביתה, אני צריך רשת איתור בעלת פריסה משמעותית כבר מהיום הראשון של יציאת התגית לשוק, ולכן האופציה לייצר תגיות כאלו בעיקר שמורה ליצרניות הגדולות. ככל שרשת האיתור גדולה יותר – כך דיוק ומהירות עדכון המיקום תהיה גבוהה יותר. כלומר, מה שהופך את תגית האיתור ל-"חזקה" הן לא התכונות הפיזיות שלה, אלא פריסת הרשת אליה היא מחוברת.

נכון לשנה זו דבר אחד בטוח – אם קניתם AirTag, אתם כנראה תצליחו לאתר אותו בכל מקום בעולם, והוא בטוח לא יהיה מוגבל רק לעיר שלכם. זאת גם אחת הסיבות שהוא כל כך יעיל בטיולים לחו"ל: חוששים שהמזוודה שלכם תלך לאיבוד? פשוט תזרקו את אחת התגיות פנימה.

נקודה מעניינת נוספת: בשנים האחרונות יכולת איתור לא מקוון נוספה גם למכשירי Apple נוספים כמנגנון הגנה מפני איבוד\גניבה, כדוגמת Iphone\Ipad ואפילו AirPods. זאת אומרת, גם כאשר מכשיר ה-Apple שלכם מכובה, עדיין ניתן יהיה לאתר אותו באמצעות רשת FindMy!

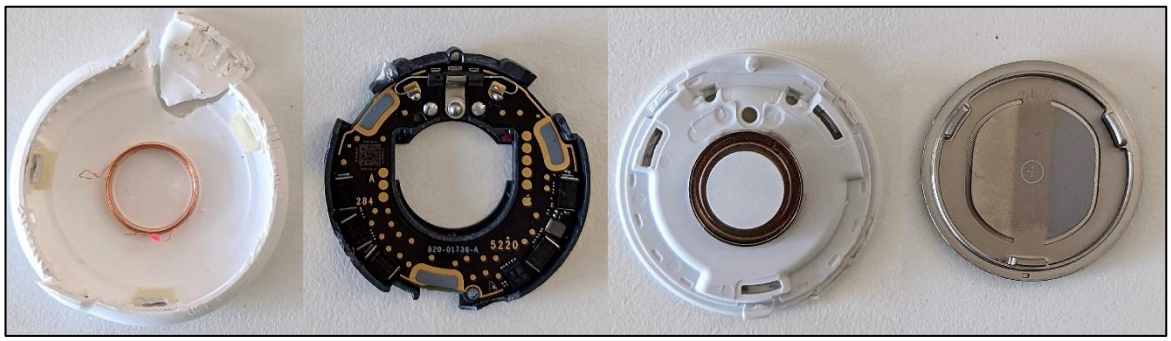
מאחורי הקלעים מדובר באותו פרוטוקול כמו זה של ה-AirTag. לכן, כל מה שנדבר עליו ונדגים במהלך המאמר יהיה תקף לא רק ל-AirTag, אלא גם לשאר מכשירי Apple שעושים שימוש בפרוטוקול הזה.

מבנה ה-AirTag

כעת נסקור את המבנה הפיזי של תגית ה-AirTag. מתחת למעטה הפלסטיק הלבן מסתתרים חיישנים שמסודרים בצורה אלגנטית, כאשר הבטרייה תופסת את רוב נפח ה-AirTag. הסקירה תהיה נכונה גם כלפי רוב תגיות האיתור הנוספות, רק שבמקרה שלהן כנראה יהיה מדובר ברכיבים מיצרנים אחרים ובעיצוב שונה.

תגית ה-AirTag בעלת החיישנים הבאים:

- חיישן BLE – כל תקשורת ה-AirTag נעשית על גבי Bluetooth Low Energy, תקן תקשורת אלחוטי המבוסס על בלוטות', אבל שנועד לצרוך הרבה פחות אנרגיה . הוא עושה זאת באמצעות העברת כמויות קטנות של מידע בלבד, ובקצב נמוך.
- חיישן UWB מסוג U1 של Apple – ראשי תיבות של "Ultra Wideband", חיישן המספק ניווט ברזולוציה גבוהה עבור מכשירים תומכים (Iphone 11 ומעלה). מדובר בצ'יפ שמסוגל לשדר רדיו בתדר ובפולסים גבוהים, לטווח קצר ובתזמון מדויק. בכך הוא מאפשר למכשיר המאתר את התגית לחשב את המרחק והכיוון אל התגית באמצעות מדידת זמן מעבר האות (Time of Flight, המכשיר מחשב את כמות הזמן שלקחה לאות להגיע, ומכיוון שהמהירות בה הגל האלקטרומגנטי קבועה – מהירות האור, ניתן לקבוע את המרחק המדויק של התגית)
- חיישן NFC, של חברת NXP – מאפשר את סריקת התגית. במקרה של אובדן התגית, סריקה תוביל להודעה שהבעלים יכול להשאיר.
- Accelerometer - חיישן תאוצה באמצעותו תגית ה-AirTag יודעת שהיא בתנועה, וכך תוכל לשנות את המצב בו היא נמצא (נעשה שימוש בחיישן הזה באחד ממנגנוני הפרטיות ש-Apple יישמה, על כך בהמשך).
- רמקול – ניתן להשמיע צליל בתגית על מנת לאתר אותה ביתר דיוק. אפשר לעשות זאת עבור תגית שבבעלותנו או כזאת שלא, אם היא חשודה כעוקבת אחרינו (על כך בחלק "פרטיות ב-AirTag")
- בטרייה – מסוג CR2032, בטריית ליתיום "כפתור" סטנדרטית, הניתנת להחלפה על ידי המשתמש. הבטרייה יכולה לשמש את ה-AirTag זמן של בערך שנה, עבור שימוש ממוצע.



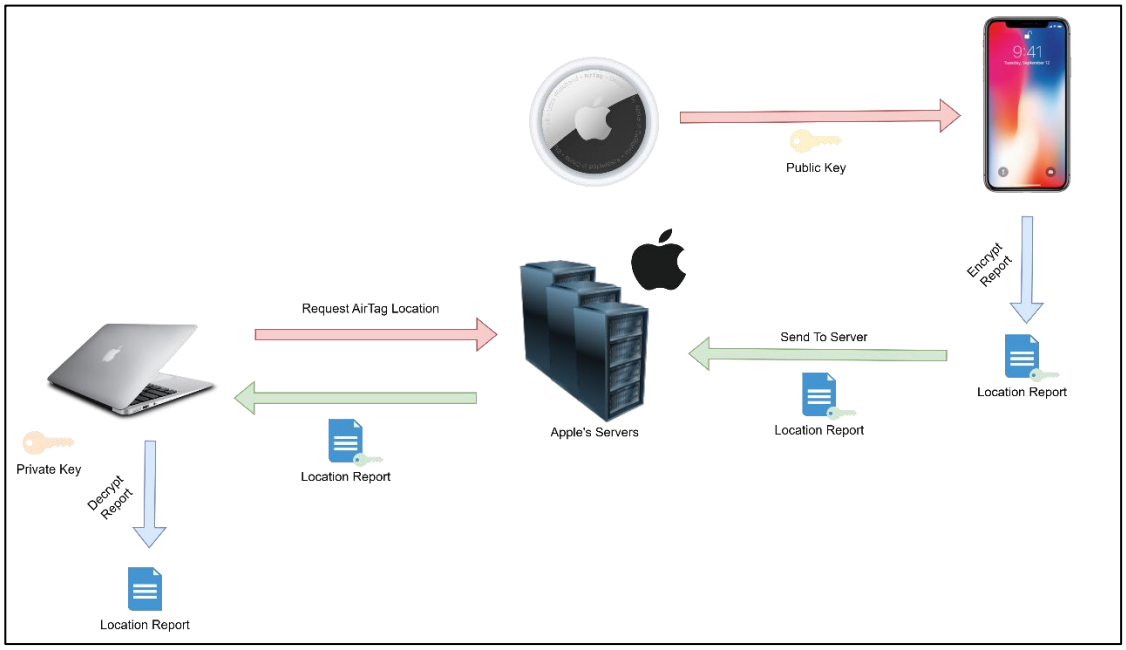
[קרדיט: Apple AirTag Reverse Engineering]

לאחר שהכרנו כיצד ה-AirTag בנוי פיזית, נדבר על הקסם האמיתי מאחורי הקלעים, שמאפשר לבנות אותו בצורה כה מינימלית ולייצר אותו בתפוצה כה רחבה, אך עם זאת לאפשר גם כיסוי נרחב ומעקב בצורה מדויקת וזמינה, והוא הפרוטוקול שלו - הנקרא "**Offline Finding**".

Offline Finding Encryption

מנגנון הצפנה שילווה אותנו הרבה בתיאור הפרוטוקול הוא הצפנה א-סימטרית.

נעבור על הסכמה הבאה שתייצג לנו בצורה מופשטת את צורת תקשורת ה-AirTag:



כעת נסביר בפשטות, ובהמשך ניכנס לפרטים הקטנים: תגית ה-AirTag מפרסמת את המפתח הפומבי של המשתמש כל interval זמן מסוים. המפתח הפומבי משודר על גבי פקטות BLE, אותן קולטים המכשירים המשתתפים ברשת ה-FindMy, כדוגמת מכשירי Iphone ו-MacBook שנמצאים בסביבה.

המכשיר שמשתתף ברשת (מעתה נקרא לו המכשיר "המאתר") לוקח את המפתח הפומבי שקיבל מתגית ה-AirTag, לאחר מכן מצפין את **המיקום הנוכחי שלו** (לא זה של התגית, וזאת הסיבה שמדובר במיקום גס) בנוסף ל-Timestamp הנוכחי.

לאחר מכן, המכשיר המאתר שולח את התוכן המוצפן לשרתים של Apple. אותו התוכן יהיה זמין לאחר מכן לבעלי התגית, שיקבל אותו מהשרתים של Apple, יפענח אותו באמצעות המפתח הפרטי שלו – וכך יוכלו לגלות את מיקום התגית. שימו לב שמיקום התגית מוצפן **מקצה לקצה** (EndToEnd Encrypted) ולא יהיה זמין אפילו ל-Apple במהלך שהותו על השרתים שלהם, וזאת באמצעות הצפנה א-סימטרית.

אך כיצד המפתח הפומבי מגיע בכלל אל תגית ה-AirTag, וכיצד המיקום מוצפן בעזרתו?

Elliptic Curved Encryption

המפתח הפומבי מגיע אל תגית ה-AirTag בתחילת השימוש בו, והוא "תהליך הצימוד" – בו מתבצעת העברת המפתחות בין מכשיר הבעלים של התגית לתגית עצמה באמצעות שבב ה-NFC שנמצא בשני המכשירים. אך לפני שנתאר את תהליך החלפת המפתחות, עלינו להבין את סוג ההצפנה שעומדת מאחורי תגית ה-AirTag: והיא ECC P-224.

ECC הינו קיצור של "Elliptic curve encryption". מדובר בהצפנה א-סימטרית מסוג קצת שונה מזאת שאנחנו מכירים מחיי היומיום (לדוגמה RSA), והיא הצפנה מסוג עקום אליפטי.

"הצפנת עקום אליפטי היא שיטת הצפנה אסימטרית העושה שימוש במבנה האלגברי-גאומטרי הנקרא עקום אליפטי מעל שדה סופי גדול, למימוש מערכת כגון פרוטוקול דיפי-הלמן" ויקיפדיה

לא ניכנס לעומק כיצד הצפנה זו עובדת, ניתן לקרוא יותר בפירוט במאמר הצפנה מבוססת עקומים אליפטיים של גדי אלכסנדרוביץ' שפורסם גם הוא במגזין, אבל נסביר זאת כך בקצרה: אם RSA נשען על כך שקשה מאוד לפרק מספר לשני הגורמים הראשוניים שלו, הצפנת ECC מבוססת על כך שקשה מאוד לחשב את האלגוריתם הדיסקרטי בעקום אליפטי (לא להיבהל, זה יהיה המשפט הכי מסובך שתקראו פה). להסביר את ההצפנה במלואה לא תהיה משימה קלה, ומכיוון שתוכן המאמר אינו מתמטי, לא נפרט על התיאוריה שמאחוריה; אלא נדבר על היתרונות שלה:

- מפתח קטן יותר משמעותית מ-RSA, עם רמת אבטחה כמעט זהה. לדוגמה, מפתח ECC עם 28 בתים (224 ביטים) יהיה שקול למפתח RSA עם 256 בתים בקירוב.
- מהיר משמעותית ביצירת המפתחות מאשר RSA (פי כמה עשרות).
- מצריך פחות כוח חישוב ליצירת המפתחות – חסכוני באנרגיה.

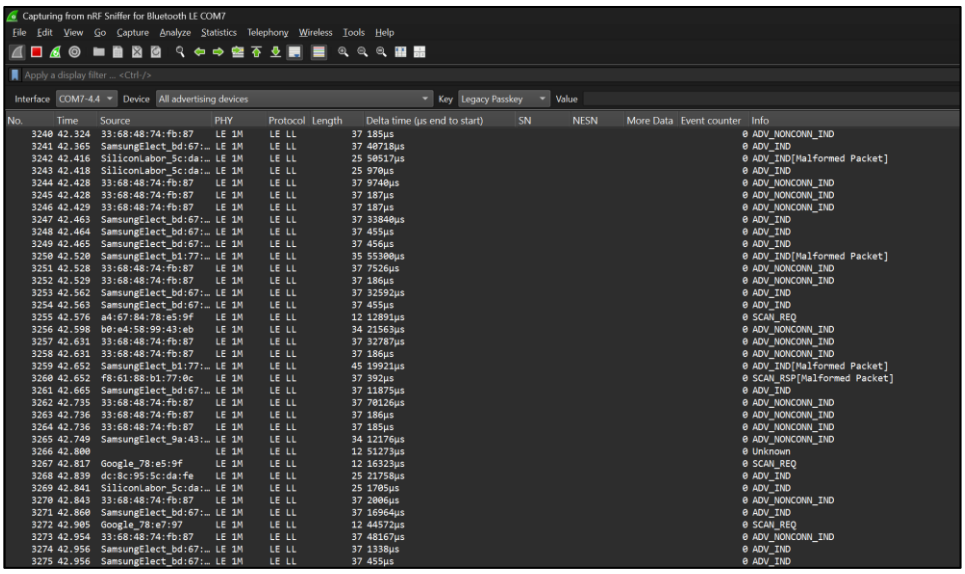
ניתן לראות שהיתרונות הללו מתאימים בדיוק לצורך של רכיבי IOT הדלים בכוח עיבוד וזיכרון, כדוגמת תגית ה-AirTag שלנו. גודל המפתח הקטן מאפשר חיסכון בזיכרון ותעבורה רזה משמעותית שפרוטוקול ה-BLE יתמוך בו. בנוסף, המהירות בה נוכל לחשב מפתחות חדשים תעזור לנו לשימוש יעיל על החומרה החלשה שלרשותנו. שמענו בעיקר על היתרונות של ECC. שאלה שיכולה לעלות בעקבות כך היא: אז למה לא כולם משתמשים בהצפנה בה אלא ב-RSA? נענה על כך בקצרה: RSA ותיק יותר (משנות ה-70) ו-ECC נכנסה לשימוש מסחרי רק מאוחר יותר (תחילת שנות ה-2000) כלומר הוא יחסית חדש, אז RSA הספיק להיות ברירת המחדל בתשתיות רבות. בנוסף לכך, מימוש ECC נתון יותר לשגיאות ביישום בגלל התיאוריה היחסית מורכבת מאחוריו, מה שמוביל לעוד סיבה בה אנשים מעדיפים להישאר ב-RSA: יותר קל להבין אותו. אחרי המקרה של Dual_EC_DRBG, מחולל מספרים אקראיים מבוסס עקום אליפטי גם הוא בו ה-NSA דחפה לשימוש למרות החולשתיות שבו (בשביל שיהיה לה קל יותר בשבירת הצפנות), אמון הציבור בתקופתו פחת באלגוריתמים כאלו, וניכרה ההעדפה להישאר באלגוריתמי הצפנה מוכרים.

תקשורת ב-AirTag

כעת אחרי שסקרנו את המבנה הפיזי של ה-AirTag, הגיע הזמן שנצלול אל הצורה בה הוא מתקשר. נתחיל בדרך של Blackbox, ונסניף את תעבורת ה-AirTag.

בשביל להסניף תעבורת BLE בכרטיס בלוטות' רגיל שנמצא בלפטופ שלנו לא מספיק, ולשם כך ניעזר במודול BLE חיצוני של חברת Nordic מדגם nRF52840 (אותה החברה שמייצרת את חיישן ה-BLE עבור Apple), אך גם מודול תואם של חברה אחרת יעשה את העבודה. אחרי שנתקין את הדרייבר עבור החיישן, יהיה זמין עבורנו כרטיס רשת חדש כשנפתח את Wireshark.

נתחיל להסניף על כרטיס הרשת החדש, וכבר יתגלו מולנו כמויות עצומות של מידע, שרובו כלל לא קשור ל-AirTag:



נראה שאיתור פקטות ה-AirTag שלנו בהסנפה יצריך אותנו לדעת קצת יותר על מבנה ההודעות שלו 🙁 . אך אל דאגה, נרחיב על כך כעת.

ל-BLE ישנם שתי סוגי הודעות המשמשות להעברת מידע: אחת מסוג "Advertising Packet" שהיא למעשה הודעת Broadcast שמטרתה להכריז על נוכחות המכשיר ("Beaconing"), כאשר גודל ההודעה המקסימלי הוא 37 בתים - כאשר 31 בתים מהם יהיו ה-Payload שיכיל את תוכן ההודעה. הסוג השני של ההודעה היא "Data Packet", בה ניתן לעשות שימוש לאחר שהושג חיבור BLE מול מכשיר נוסף, והיא תומכת בהעברת מידע בכמות גדולה יותר.

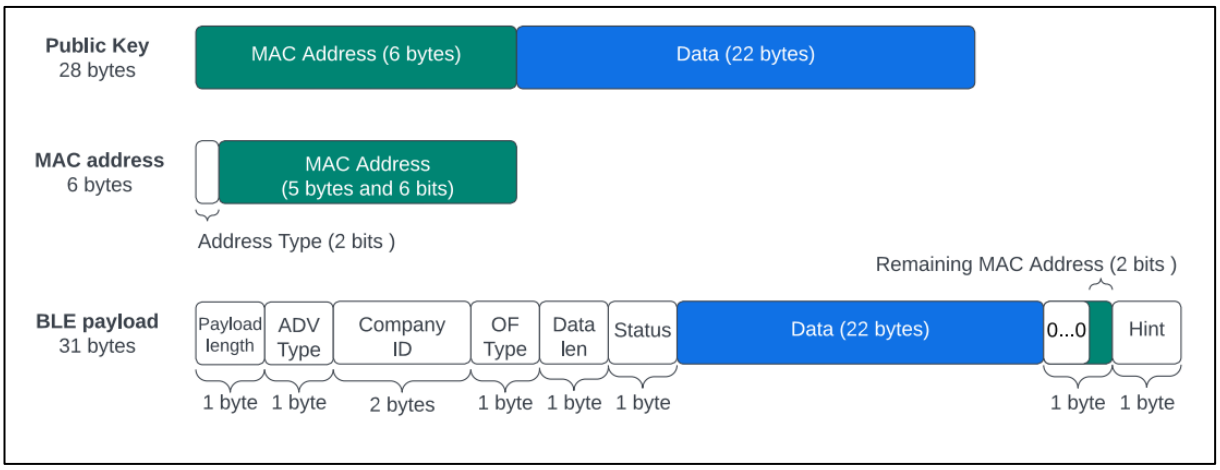
בשידור מפתח פומבי של מכשיר ברשת FindMy, Apple עושה שימוש ב-Advertising Packet.

מתוך 31 הבתים הפנויים עבור ה-Payload, Apple מכניסה **9** בתים של Metadata משלה (הם מכילים אינדקציות לגבי סטטוס המכשיר והיצרן), כלומר כעת נשארנו עם רק **22** בתים פנויים לכתיבת תוכן, ועלינו לפרסם מפתח פומבי בגודל **28** בתים. אבוי! מה נעשה עם 6 הבתים הנותרים?

בשביל לענות על הדרישה, Apple החליטה לדרוס את שדה ה-Mac עם 6 הבתים הראשונים של המפתח הפומבי, ובכך להרוויח עוד מקום.

שימו לב שבשל כך נעלמה יכולת הזיהוי של התגית - לא ניתן יהיה לדעת אם תגית מסוימת נמצאת בחדר בהינתן הסנפת BLE, כי כל המזהים - כתובת ה-Mac של השולח והמפתח הפומבי, משתנים תדיר במהלך היום.

אנקדוטה: ישנו פרויקט הנקרא "Send My" שמטרתו להעביר מידע לבחירת המשתמש באמצעות רשת Find My, ובכך לאפשר יציאה לאינטרנט לכל רכיב בעל חיישן BLE שנמצא בסביבת מכשירי Apple - וזאת על ידי ניצול את המכשירים המאתרים המשתתפים ברשת. הפרויקט עושה שימוש ב-OpenHayStack בשביל להרכיב הודעות Find My זדוניות, ובאמצעות שינוי המפתח הפומבי הפרויקט מקודד מידע לבחירתו.



[מקור: Track You: A Deep Dive into Safety Alerts for Apple AirTags]

נסתכל על הסרטוט הבא שמייצג את מבנה הודעת ה-Advertisement שה-AirTag משדר:

ערך ה-**Company ID** יעיד על החברה שייצרה את המכשיר (0x004C עבור Apple),

ערך ה-**of_type** - קיצור של offline finding, יעיד על סוג השירות המבוקש על ידי המכשיר המשדר (0x12 עבור שירות מסוג FindMy, או 0x07 כאשר התגית עוד לא עברה תהליך צימוד).

שדה ה-**Status** יעיד על סוג המכשיר: Iphone\Ipad\AirPod\מכשיר תואם אחר ועל מצב הבטרייה שלו.

לדוגמה, ערך של 0x10 יעיד על AirTag עם בטרייה מלאה, וערך של 0x50 יעיד על בטרייה במצב בינוני.

פענוח הודעת AirTag

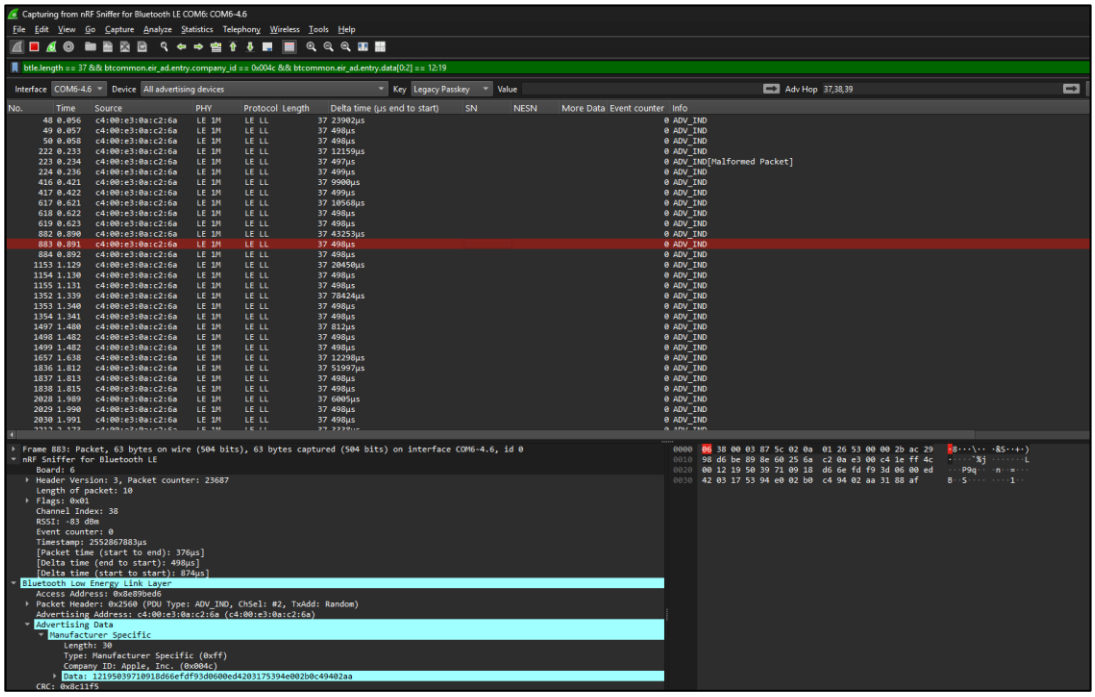
כעת, נחזור ל-Wireshark ונבנה פילטור על מנת למצוא את פקטות ה-AirTag שלנו ולנתח אותן יחדיו. יש לכם כבר רעיון איך נעשה זאת?

[ילדים, למישהו יש רעיון ל-Display Filter שיציג את פקטות ה-AirTag?]

ישנן כמה דרכים לגשת לעניין, האחת שמצאתי קלה יותר היא לפלטר רק על פקטות ששודרו על ידי מכשיר Apple – בעזרת השדה "Company ID" , כאשר הערך של Apple הוא 0x004c. כעת נפלטר עבור סוג השירות הספציפי שלנו, בשביל לנקות רעש מפרוטוקולי BLE אחרים של Apple (כדוגמת iBeacon או AirDrop), נוסיף את סוג השירות המבוקש: מסוג FindMy, שהערך 0x1219 מייצג אותו. סוג השירות מצוין תחת ה-Metadata ש-Apple הוסיפו ל-Payload של פקטת ה-BLE. נשלב את הכל, ונקבל את ה-Display Filter הבא:

```
btcommon.eir_ad.entry.company_id == 0x004c && btcommon.eir_ad.entry.data[0:2]
== 12:19
```

אחרי שהרצנו את הפלטור, מוצגות לנו פקטות השייכות ל-AirTag 😊



שימו לב שגודל הפקטות הוא 37 בתים, כמו שהזכרנו קודם. נבחר פקטה אחת לדוגמה, וננתח את הערכים שלה. עבור הודעת ה-BLE הבאה:

```
0x06 0x38 0x00 0x03 0x6a 0x64 0x02 0x0a 0x01 0x26 0x55 0x00 0x00 0x77 0x39 0x7a
0x99 0xd6 0xbe 0x89 0x8e 0x60 0x25 0x6a 0xc2 0x0a 0xe3 0x00 0xc4 0x1e 0xff 0x4c
0x00 0x12 0x19 0x50 0x39 0x71 0x09 0x18 0xd6 0x6e 0xfd 0xf9 0x3d 0x06 0x00 0xed
0x42 0x03 0x17 0x53 0x94 0xe0 0x02 0xb0 0xc4 0x94 0x02 0xaa 0x31 0x88 0xaf
```

נקבל את הערכים הבאים:

Company ID: **Apple** (0x004c) [index 30-31]

Protocol Type: **FindMy Network** (0x1219) [index 32-33]

Status Byte: 0x50 – **AirTag with Medium Battery Level** [index 34]

Encrypted Public Key (22 bytes): **0x397109118d66efdf93d0600ed420317539e402b0**

AirTag States

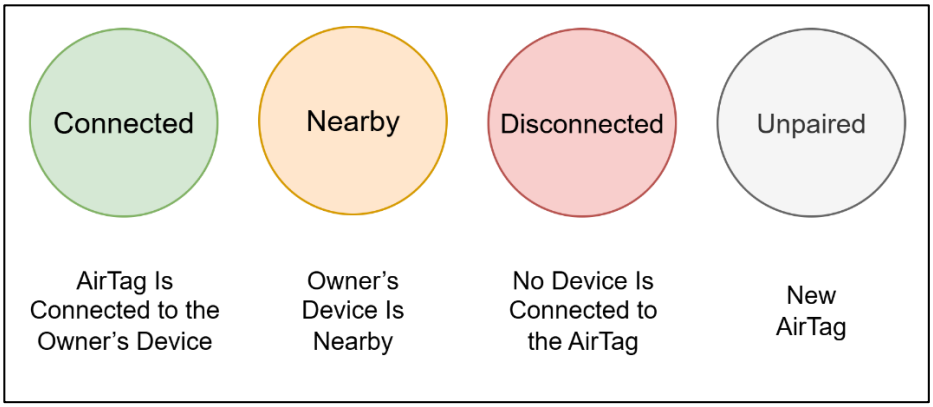
כעת נסביר על המצבים השונים בהם AirTag יכול להימצא (States), כאשר בהתאם לכך ישתנו הערכים בהודעות ה-Broadcast אותן הוא משדר, בנוסף ל-interval השידור:

Unpaired – מצב ברירת המחדל לפני תהליך ההצמדה. במצב הזה תגית ה-AirTag תשדר פקטות BLE עם כתובת ה-MAC הדיפולטית שלה וחלק מה-HASH של המספר הסריאלי שלה כל 33 מילישניות, על מנת למצוא מכשירים תומכי FindMy לבצע צימוד איתם ("קריאת הזיווג" של התגית).

Connected – התגית נמצאת בטווח ה-BLE של מכשיר הבעלים: במצב זה היא נמנעת מלפרסם את המפתח הפומבי שלה.

Nearby – מצב קצר בזמן בו התגית יכולה להימצא בו עד 15 דקות בקירוב. הוא מתחיל ברגע שהתגית יוצאת ממצב Connected (מאבדת חיבור עם מכשיר הבעלים). במצב זה התגית עדיין אינה זמינה לאיתור באמצעות רשת FindMy, שכן היא עוד לא משדרת את המפתח הפומבי המלא שלה.

Disconnected – אחרי מצב Nearby, התגית נכנסת למצב של "ניתוק". במצב זה התגית משדרת פקטת BLE עם המפתח המלא כל 2 שניות, מה שמאפשר למכשירים בסביבה להעלות את המיקום שלה לשרתים של Apple. אם תגית נמצאת במצב הזה יותר משלושה ימים, או שהבעלים מסמן אותה בצורה ידנית במצב זה באפליקציית FindMy, מכשירים תומכים שיסרקו אותה יציגו הודעה אותה הגדיר הבעלים, שמטרתה לעזור לו למצוא את החפץ האבוד.



לפני כמה שנים הייתי בסקי, ועלה איתי מישהו למעלית. במהלך הנסיעה, כשהיינו בגובה רב – מכשיר ה-Iphone שלו נפל אל תוך השלג. הוא ידע איפה הוא נפל, אבל כשחזר לשם אחרי שירד מהמעלית, הטלפון כבר לא היה שם.

זה היה ברור שהוא נלקח, והשאלה הייתה האם נלקח בשביל להחזיר לו אותו או שהוא נגנב.

עבר יום, והוא עדיין לא מצא אותו, ולצערו הרב מי שלקח אותו הכיר את יכולת האיתור מבוססות אינטרנט באמצעות Find My Iphone, לכן גם דאג לכבות אותו. אך מה שהוא לא ידע, שסמוך לאותו הזמן Apple הורידה עדכון שמאפשר את הפונקציונליות של איתור offline עבור אייפונים, ולכן אפילו שהמכשיר הגנוב היה כבוי - הוא עדיין שידר פקטות BLE של Offline Finding, שנקלטו על ידי מכשירי Apple אחרים בסמוך.

הוא הצליח לאתר את האייפון ברזולוציה כמעט מדויקת, והיה לו את הכתובת של המלון.

הוא פנה למשטרה והגיע ביחד עם השוטרים לקבלה של המלון. הגנב, שבמקרה חזר לחדר שלו ועבר על פניהם, נבהל ממראה השוטרים, עלה לחדר – והחזיר לו את הטלפון.

איזה מזל שהוא הוריד את עדכון התוכנה בזמן 😊

כעת, נתאר מבחינה טכנית את מה שקרה מאחורי הקלעים באותו אירוע. שימו לב שבמאמר אנחנו מדברים על תגית AirTag – אבל כמו שהזכרנו בתחילת המאמר, מדובר באותו הפרוטוקול בדיוק גם למכשירי Apple אחרים, ואפילו למכשירים ש-Apple לא יצרה - אלא רק תומכי FindMy.

רצף האירועים

מהרגע שהמכשיר מתנתק ממכשיר הבעלים (נכנס למצב Disconnected) הוא מתחיל לשדר הודעות מסוג Lost, שמכילות את המפתח הפומבי המתגלגל שלו (rotating public key) שנגזר מהמפתח הפומבי שקיבל מהמכשירים הבעלים בעת תהליך ה-Pairing. המפתח משתנה כל 15 דקות ומשודר כל 2 שניות. ערך המפתח נגזר מהמפתח המקורי באמצעות פונקציית גזירת מפתח (KDF).

איך כיצד המפתח הפומבי מגיע בכלל אל ה-AirTag? זה קורה בתהליך ה-Pairing, בו אנחנו מצמידים את ה-AirTag החדש ל-Iphone\Ipad שלנו: ובאמצעות שבב ה-NFC אנחנו מעבירים את ה-"Master Public Key", המפתח הפומבי הראשון ממנו יגזרו כל שאר המפתחות הפומביים שה-AirTag ישדר במהלך חייו. שימו לב שגם ה-Master Public Key מוגדר כסוג של סוד, מכיוון, כמו שנראה אחר כך, ברגע שהוא ידוע – ניתן יהיה לחשב את כל המפתחות שיגזרו ממנו, וכך לבקש את כל המיקומים של אותו ה-AirTag מהשרתים של Apple. מכיוון שכך הדבר, גם ה-Master Public Key מועבר מעל תווך "מוצפן" (התקשורת מוצפנת בעזרת ערך ידוע מראש שמוגדר hardcoded ב-firmware של ה-AirTag ומכשיר הבעלים, ובנוסף לכך מתבצעת החלפת מפתחות).

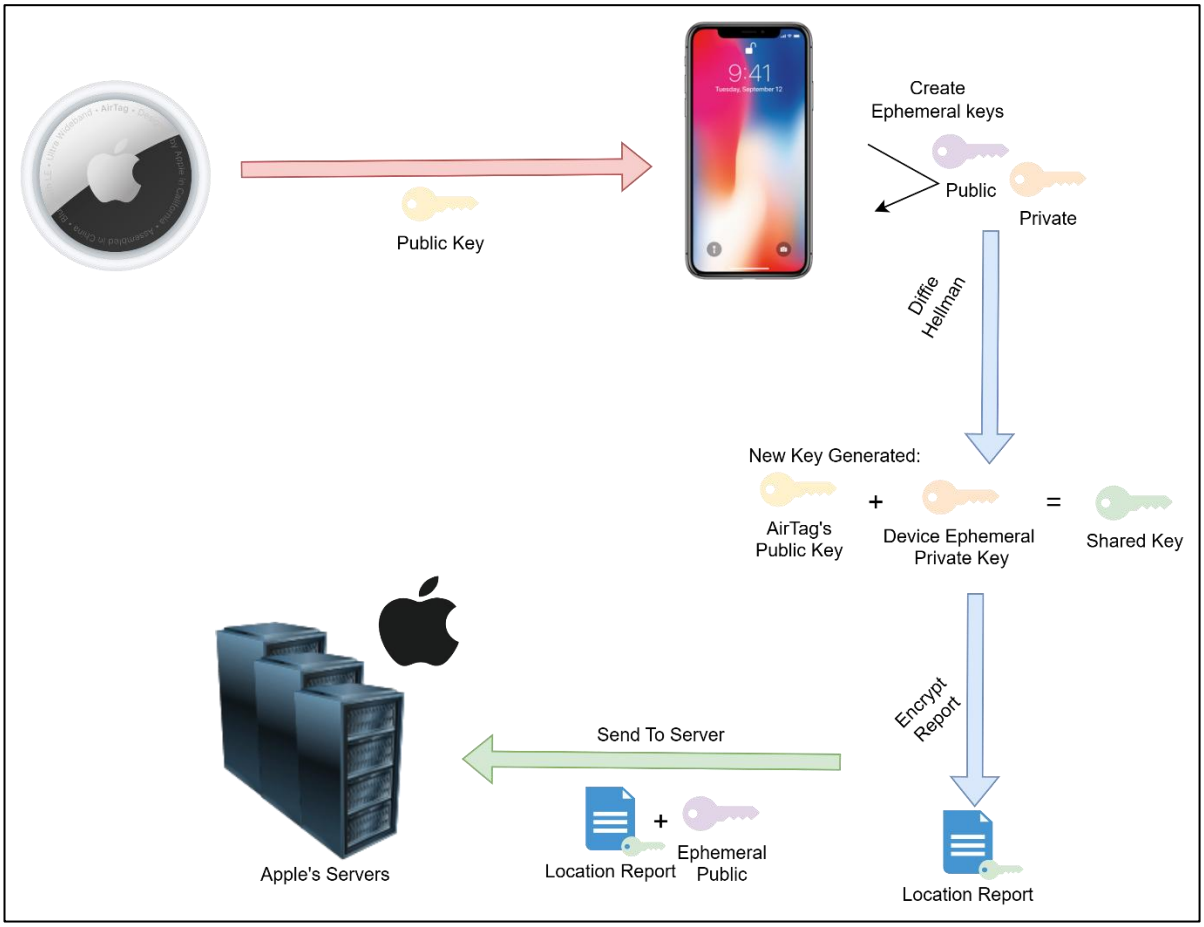
כעת נחכה עד שיהיה מכשיר שתומך ברשת FindMy בסביבה. מהרגע שזה קורה, וכנראה אחרי לא הרבה זמן (בכל זאת, רשת עם מעל מיליארד משתתפים), "המכשיר המאתר" קולט את שידור ה-BLE מסוג Lost, ומחלץ ממנה את המפתח הפומבי.

לאחר מכן, **מיוצרים בצד של המכשיר המאתר** צמד מפתחות (פרטי ופומבי) א-סימטריים נוספים מסוג ECC P-224. נקרא להם מפתחות "זמניים" (Ephemeral) כעת מבוצע ECDH – "Elliptic curve Diffie Hellman" באמצעות המפתח הפרטי החדש שנוצר והמפתח הפומבי שהתקבל מה-AirTag, בשביל לייצר מפתח חדש (נקרא לו "הסוד המשותף").

Diffie Hellman הוא פרוטוקול שיתוף מפתחות המאפשר לשני צדדים לקיים תקשורת מאובטחת מעל תווך לא מאובטח. הוא עושה זאת באמצעות יצירת שני מפתחות א-סימטריים בשני הצדדים המתקשרים, והחלפת מפתחות פומביים בין השנים. לאחר מכן הם יכולים לסכם על סוד משותף, כדוגמת מפתח הצפנה סימטרי, ולהעביר אותו בין הצדדים, בלי שגורם המצותט לשיחה יקלוט אותו. למי שרוצה להבין יותר טוב את הנושא של הצפנה מקצה לקצה מוזמן לקרוא את המאמר המרטיט של עידן שכטר.

נסכם את ה-ECDH שמתרחש כך: המפתח הפרטי החדש שנוצר מוצפן באמצעות המפתח הפומבי של ה-AirTag (ב-Elliptic Curve זה עובד עם הכפלת נקודות, אבל נסכם את זה בתור הצפנה)והמפתח הפומבי החדש שנוצר מצורף לדיווח המיקום אותו יקבל הבעלים בהמשך. כעת קיבלנו "סוד משותף" שמורכב מהמפתח הפרטי הזמני ומהמפתח הפומבי של ה-AirTag.

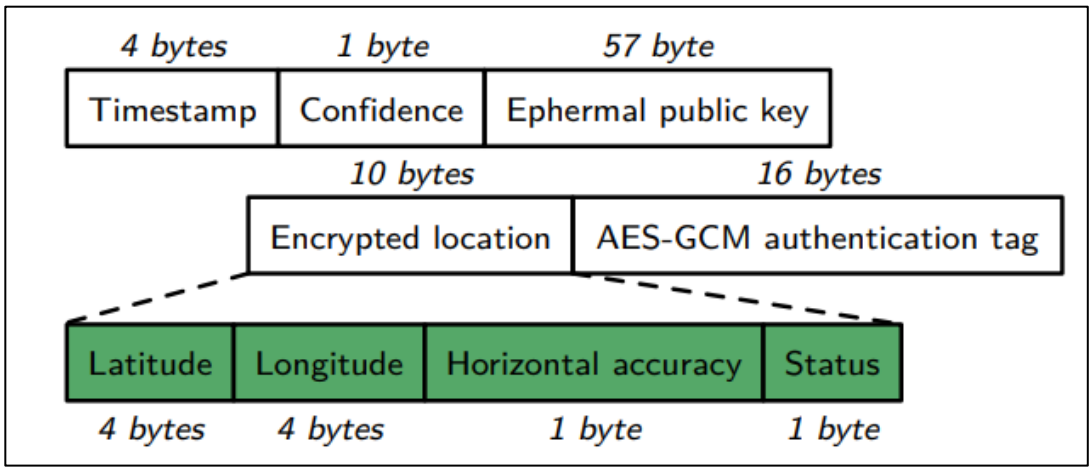
כעת מופעלת הפונקציה לגזירת מפתח Ansi x.963 KDF, כאשר המפתח הפומבי של ה-AirTag הוא ה-Entropy ו-sha-256 היא פונקציית הגיבוב, וכך קיבלנו **מפתח סימטרי חדש**, שנגזר מאותו "סוד משותף".



לאלה מאיתנו שלא עקבו עד כה – עם קצת מתמטיקה, המיקום מוצפן בעזרת מפתח סימטרי שנגזר מהמפתח הפומבי של ה-AirTag.

סוף סוף יש לנו מפתח שבעזרתו נוכל להצפין את המידע! איך רגע, עוד לא סיימנו. מה בדיוק אנחנו מצפינים? המכשיר המאתר מרכיב הודעה שנקראת Location Report" שמכילה מידע חשוב אודות המיקום, לדוגמה:

- נקודת ציון נוכחית (latitude + **Longitude**)
- זמן נוכחי (Timestamp)
- רמת וודאות במיקום (Accuracy)
- המפתח הפומבי הזמני שנוצר במכשיר המאתר



[מקור: Security and Privacy of Apple's Crowd-Sourced Location Report]

כאשר בעלי המכשיר יקבל את הדיווח על המיקום, הוא יפענח אותו בעזרת הפעולה ההפוכה: הוא יבצע Diffie Hellman עם המפתח הפומבי הזמני שקיבל מהדיווח והמפתח הפרטי של ה-AirTag, וכך הוא יחשב את המפתח הסימטרי בעזרתו הוצפן הדיווח.

המכשיר המאתר לא ימהר לשלוח את הדיווח לשרתים של Apple ברגע שהוא מרכיב אחד, אלא יעדיף לשלוח אותם בקבוצות (Batch), בשביל לחסוך בבטרייה ובנפח תעבורה. ממחקרי מהנדסה לאחור שבוצע לפרוטוקול ה-Offline Find (ספציפית זה של אוניברסיטת Darmstadt) נמצא שהזמן החציוני מהרגע שהדוח מורכב עד שהוא נשלח ל-Apple הוא 26 דקות, ויכולות לעבור אפילו כמה שעות אם המכשיר המאתר נמצא במצב חיסכון סוללה.

כלומר, נעשה פה שיקול מצד Apple בכל הנוגע לחיסכון בסוללה ותקשורת נתונים מצד המכשירים המאתרים (לאלה מאיתנו שמתעצבנים על הכפייה להשתתף ברשת)

כל Report כזה שמועלה לשרתים של Apple מועלה עם **מזהה**. אותו מזהה מורכב מערך ה-sha-256 של המפתח הפומבי המתגלגל שה-AirTag שידר, שנגזר מה-public master key שהועבר ממכשיר הבעלים בתהליך הצימוד.

בשביל להעלות דיווח, המכשיר המאתר מבצע בקשת POST ל-Endpoint הבא:

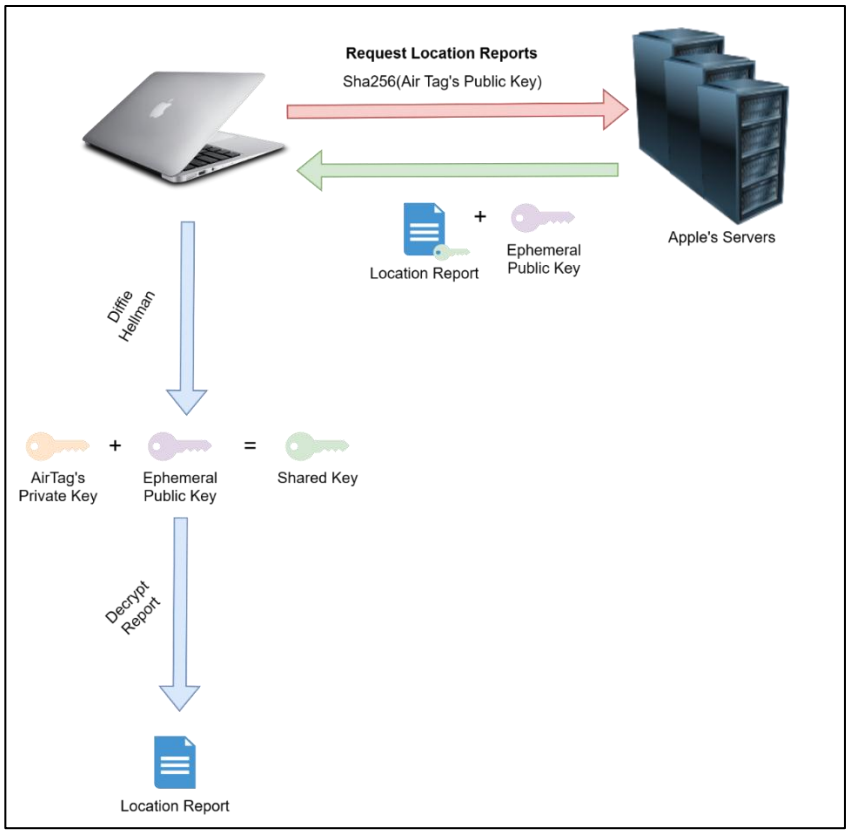
```
https://gateway.icloud.com/acsnservice/submit
```

כעת, בשביל לקבל דיווח, על מכשיר הבעלים לחשב את ערכי ה-sha-256 האפשריים עבור ה-AirTag שלו (הרי גם לו ידוע ה-public master key) ולתשאל עבורם את השרת. הוא יקבל בחזרה את ה-Reportים המוצפנים, אותם הוא יכול לפענח בעזרת המפתח הפרטי שבבעלותו, והמפתח הפומבי שנמצא בדיווח.

בשביל לקבל דיווחים, מכשיר הבעלים מבצע בקשת POST ל-Endpoint הבא:

```
https://gateway.icloud.com/acsnservice/fetch
```

נדגים זאת בסרטוט הבא, שמייצג פענוח Location Report על ידי מכשיר הבעלים:



שימו לב מה קרה כאן – המידע מוצפן **מקצה לקצה**, ורק הבעלים של המכשיר יוכל לפענח אותו באמצעות ה-Private key שמאוחסן אצלו, וגם Apple או רשויות החוק לא יהיו זמינים למידע הזה. אך רגע, אם הצמדתי את ה-AirTag שלי במכשיר אחד, אני עדיין אוכל לאתר אותו במכשיר אחר בבעלותי! איך קורה הדבר?

ובכן, אותו מפתח פרטי "סודי" שיצרנו בתחילת התהליך לא מאוחסן רק על גבי מכשיר הבעלים, אלא גם מסונכרן ל-**iCloud Keychain** ששייך ל-Apple ID אתו אנחנו מחוברים למכשיר.

iCloud Keychain הוא מאין "מחזיק מפתחות" וירטואלי, בו מאוחסנים הסודות ששמרנו במכשירי ה-Apple שלנו, כדוגמת שמות המשתמש וסיסמה לאתרים, כרטיסי אשראי, סיסמאות Wifi, Root certificates ומפתחות פרטיים כדוגמת אלה שמשמשים אותנו ל-Offline Finding. כל הערכים השמורים תחת הכספת הזאת (שנמצאת ב-iCloud, הענן של Apple) גם הם מוצפנים מקצה לקצה באמצעות מפתח ייחודי, שנמצא רק במכשירים שבבעלותנו.

כעת, כאשר הבעלים של המכשיר יפתח את אפליקציית FindMy ויבחר במכשיר הנאבד, הוא יחשב את ערכי המפתחות האפשריים לקבלת ה-Reportים מהשרת באמצעות ה-public key שבבעלותו ובבעלות ה-AirTag, וזה יחזיר לו בתשובה את רשימת ה-Reportים. ניתן לקבל מהשרתים של אפל את היסטורית המיקומים עד **שבוע אחורה** (Fun fact, המידע הזה לא זמין דרך אפליקציית FindMy, אבל כן דרך תשאול ה-API של iCloud - כך ניתן להרכיב ככה גרף מלא של תזוזת המכשיר, ולהסיק מידע מעניין נוסף כדוגמת מקומות שהיו בביקור באופן קבוע). בפועל יוצג לנו באפליקציית FindMy המיקום האחרון שבו אותרה התגית, בחישוב גס. לאחר שנתקרב מספיק אל התגית, בשאיפה נהיה בטווח ה-UWB (Ultra Wide Band) שהוא בין 10 ל-30 מטרים, ואז נוכל לקבל הוראות הכוונה ישירות אל המכשיר (במידה ואנחנו מחזיקים במכשיר תומך, אייפון 11 ומעלה).

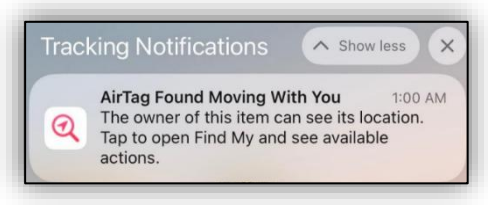
וזהו, מצאנו את ה-AirTag שלנו 😊

[קרדיט: מחזיק המפתחות שלי]

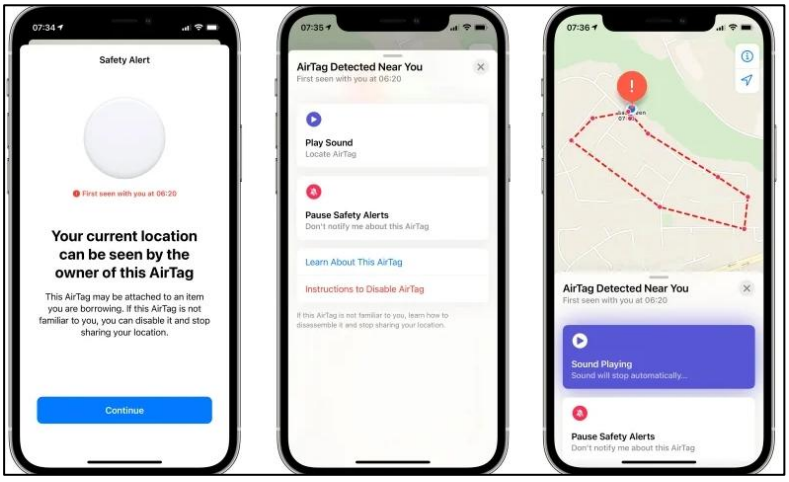
פרטיות ב-AirTag

ניכר ש-Apple השקיעו הרבה מחשבה בפרטיות במהלך תכנון ה-AirTag. זה מתחיל עוד מתכנון הפרוטוקול - שכולל החלפת מפתחות קבועה שמונעת איתור של תגית על ידי גורמים המאזינים ברקע, ועד להצפנה מקצה לקצה שמונעת מ-Apple עצמה לגשת אל המיקומים. אך מה לגבי פרטיות של אנשים שכלל לא קנו תגית AirTag, ויכולים ליפול קורבן למעקב?

ובכן, במהלך השנים האחרונות קרו לא מעט מקרים בהם אנשים נעקבו באמצעות התגיות האלו, מה שהוביל לתביעה כנגד Apple על ידי חלק מהקורבנות. בעקבות מקרים כמו אלו, Apple הוסיפה במהלך השנים מנגנוני אבטחה שנועדו לשמור עלינו מפני מעקב לא רצוי.



1. "AirTag Found Moving With You" כנראה המנגנון הכי יעיל. כאשר תגית ה-AirTag נמצאת במצב Disconnected, המכשיר שבבעלותנו (בין אם אחד של Apple או אפילו Android) יזהה אם התגית זזה ביחד איתנו למשך מספר שעות, או הגיעה איתנו למיקום שאנחנו מגיעים אליו הרבה (לדוגמה הבית) ויציג את ההתראה לאחר מספר דקות, כולל את היסטוריית המיקומים שהתגית עקבה אחרינו. זה אכן מנגנון עוצמתי, אבל ניתן לעקיפה על ידי שינוי המפתח הפומבי שמשודר. ניתן לעשות זאת באמצעות מכשירים דמויי AirTag שמנצלים את פרוטוקול FindMy: הם יריצו תוכנה אחרת כדוגמת OpenHayStack, פרויקט Open Source שמשתמש ברשת ה-Offline Finding של Apple, ונותן לנו שליטה מלאה על המידע המשודר, כדוגמת תוכן המפתח. בנוסף, נמצא שאם נערוך את שדה ה-Status של פקטות ה-BLE שמשודרות במצב Disconnected להיות שונה מזה של AirTag או AirPods, ההודעה לא תופיע - מנגנון האבטחה הזה מופעל רק אם המעקב מתבצע על ידי אחד משני אלו, ולא על ידי Iphone או מכשיר לא מזוהה אחר.



2. ניגון צליל – לאחר שתגית ה-AirTag סומנה ככזאת חשודה, ניתנת האופציה ב-IOS לגרום לה לצפצף. האפשרות הזאת מתאפשרת גם באפליקציית "Tracker Detect" ש-Apple פרסמה ל-Android, אפליקציה שפותחה על ידם לאחר הלחץ הציבורי שהופעל עליה בעקבות מקרי המעקב שנעשו באמצעות תגיות ה-AirTag.
3. ניגון צליל אוטומטי – בשביל לשפר זיהוי תגיות חשודות גם עבור אנשים שלא מסתובבים עם מכשיר נייד, לאחר שלושה ימים שהתגית רחוקה מבעליה היא תיכנס ל-"Lost Mode". במצב זה, התגית תנגן צליל כל 6 שעות במהלך תנועה (מזוהה באמצעות חיישן ה-Accelerometer).

סיכום

כולנו יכולים להסכים על כך שתגיות ה-AirTag הן דבר מאוד שימושי. הן יכולות לעזור לאתר את המזוודה האבודה בשדה התעופה, את חיית המחמד שברחה, את צרור המפתחות ששוב שכחנו, או אפילו את שלט הטלוויזיה שנעלם באחת הספות - הן פשוטות לשימוש, זולות ומדויקות.

אך דווקא בגלל היתרונות האלה, זה הופך אותן לכלי מסוכן בידי גורמים עוינים. אם בעבר מעקב אחרי בן אדם ברמת דיוק שכזאת היה מתאפשר רק על ידי גופי אכיפת חוק או בילוש, כעת כל אדם עם 30 דולר יוכל לרכוש תגית כזאת ולעקוב אחרי מי שירצה. כמובן שישנם מנגנוני אבטחה שנועדו למנוע זאת, אך גם אותם אפשר לעקוף עם מספיק מוטיבציה.

למרות זאת, אני באופן אישי שמח שקונספט ה-Offline Finding נמצא איתנו, ואומנם Apple היא לא זאת שהמציאה אותו, אבל כמו חידושים רבים שהיא הביאה לעולם – היא זאת שגרמה לו לעבוד כל כך טוב, והיא זאת שסחפה יצרנים אחרים לבוא אחריה, וכעת אי אפשר יהיה להתעלם מזה.

אני חושב שהיתרונות של תגית ה-AirTag עולים על החסרונות ללא ספק, והאופן בו התגית הקטנטנה הזאת בנויה, ועוד יותר מכך – הצורה בה היא מתקשרת, פשוט ראויים להערצה 😊 .

מקורות מידע

- Blind My — An Improved Cryptographic Protocol to Prevent Stalking in Apple's Find My Network
- Who Can Find My Devices? Security and Privacy of Apple's Crowd-Sourced Bluetooth Location Tracking System
- Track You: A Deep Dive into Safety Alerts for Apple AirTags
- Dissecting the Encryption Protocols Inside Apple AirTags
- Apple AirTag Reverse Engineering