# הצד הטכנולוגי של כרטיס הרב-קו

מאת תומר חדד

## הקדמה

כולנו מכירים את כרטיס הנסיעה של ישראל - "רב-קו". אבל איך הוא עובד?

מאמר זה מיועד לאיש הטכנולוגי שתוהה או תהה אי פעם כיצד עובד כרטיס התחבורה הציבורית הזה שהוא משתמש בו כל הזמן. שאלות שהמאמר ינסה לענות עליהן הן:

- מה זה בכלל רב-קו ובאיזה טכנולוגיה הוא משתמש
- מה המבנה החמרתי שלו
- איזה מידע שמור עליו
- איך בדיוק מדברים איתו
- אילו סוגי מכונות קריאת כרטיסים קיימים
- מה התהליך שקורה בכל פעם שטוענים אותו בכסף או משלמים באמצעותו על נסיעה
- אילו מנגנונים קריפטוגרפים מגנים עליו, אם בכלל
- איזה מידע מגיע לחברות התחבורה ואיזה לא
- איך אני יכול לראות את כל זה בעצמי

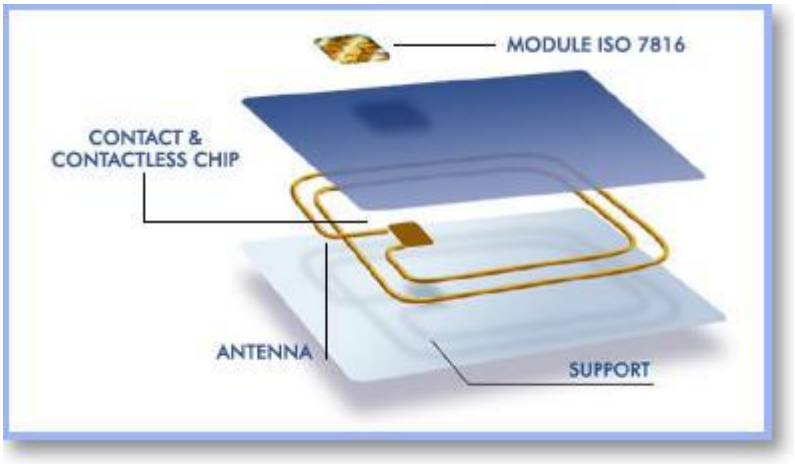
הבה נתחיל.

## רקע

חשוב לדעת שבניגוד לרושם הראשוני שאפשר לקבל, רב-קו הוא **<u>לא</u>** פשוט חתיכת פלסטיק עם פס מגנטי או כרטיס זכרון טיפש.

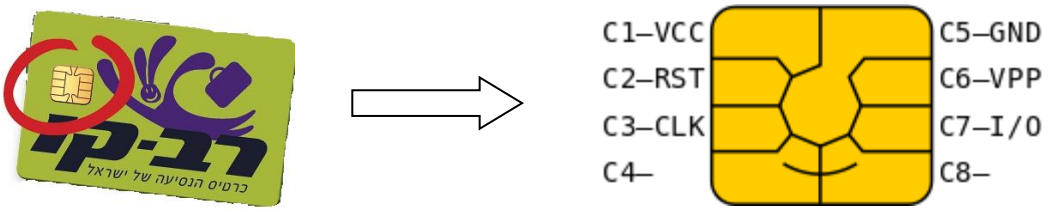
למעשה, רב-קו הוא כרטיס חכם עם מיקרו-מעבד (microprocessor) שמשתמש בתקן Calypso הבינלאומי לתחבורה ציבורית שעליו נדבר בהמשך.

הפלסטיק של הכרטיס מכיל בתוכו אנטנה לאורך ההיקף המלבני וצ'יפ זעיר (כמה מילימטרים על כמה מילימטרים) בעל יכולת *עיבוד וזכרון* משלו:




[מרכיבי כרטיס חכם, מתוך Calypso Handbook]

כיום הכרטיס מגיע עם לוחית זהב קטנה עם מגעים אלקטרוניים (contacts) שמאפשרים תקשורת גם ללא אנטנה (כלומר עם קורא כרטיסים).




[תפקידי המגעים בלוח הזהב ע"פ תקן ISO 7816]

***הערה***: בעבר כרטיסים ישנים של אגד הונפקו ללא מגעים, כך שהיה ניתן לתקשר איתם מרחוק בלבד (contactless).

למען הסר ספק אציין שגם כרטיסים ללא לוחית זהב עליהם ***כוללים*** בתוכם צ'יפ.

הצ'יפ של הרב-קו כולל בתוכו בנוסף למעבד גם זכרון EEPROM בגודל של מספר קילו-בייטים בודדים עד עשרות קילו-בייטים, RAM, ו-ROM עם *מערכת הפעלה* (כן כן) צרובה עליו.

## כרטיסים חכמים בעולם

רב-קו הוא בהחלט לא הכרטיס היחיד בתחום הכרטיסים החכמים. אותה טכנולוגיה (כרטוס חכם) נמצאת גם בכרטיסי אשראי (טכנולוגיית EMV), בכרטיסי SIM, בכרטיסי מפתח לבתי מלון, בכרטיסי זיהוי אלקטרוניים ואפילו בדרכונים ביומטריים (e-passports). בהקשרים של תחבורה ציבורית, במקומות רבים

בעולם יש שימוש בכל מני סוגים של כרטיסים חכמים שמן הסתם לאו דווקא משתמשים בתקן Calypso, שעליו ארחיב בנפרד.

האמת היא שהתקן בעל הפריסה העולמית הרחבה ביותר הוא דווקא של סדרת הצ'יפים MIFARE. דוגמאות למקומות שמשתמשים ב-MIFARE הם לונדון (עם כרטיס ה- Oyster), רוסיה (Moscow Metro), ברזיל (RioCard), סין (Yikatong), דנמרק (Rejsekort) ועוד ועוד.

לעומת זאת, דוגמאות למקומות שמשתמשים בהם ב-Calypso הם פריז (Navigo card), בלגיה ( MOBIB Card), פורטוגל (LisboaViva Card), וכמובן ישראל - שהיא גם המדינה הראשונה להשתמש ב-Calypso ברמה לאומית.

[פריסת תקן קליפסו בעולם, מתוך Calypso Handbook]

## קצת על החומרה

בישראל יש יותר מרק סוג רב-קו אחד, כאשר כל סוג בדרך כלל יוצר על ידי חברה אחרת בתקופה שונה. שם החברה שייצרה את הכרטיס רשומה בצדו האחורי של הכרטיס, בפינה הימנית העליונה. דוגמאות:

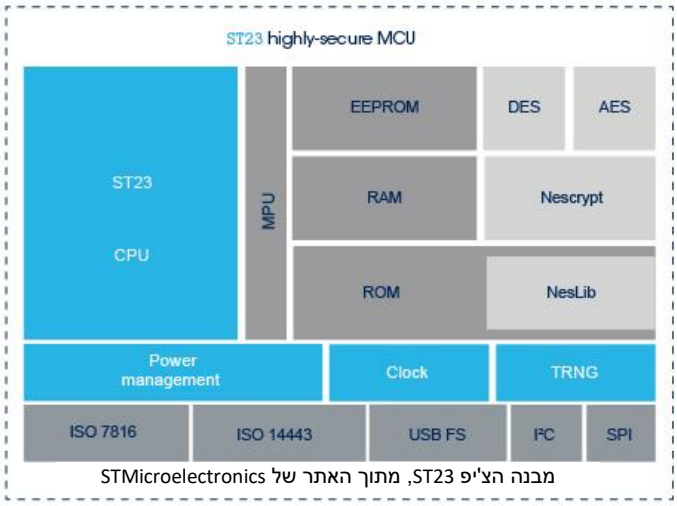
- ASK Tango / ASK FST P / ASK
- WATCHDATA
- Gemalto [GMY]
- OTI 10054127-1
- IDEMIA cityGo.

סוגי הכרטיסים השונים נבדלים אחד מהשני בין השאר בסוג החומרים המדויק שבשימוש, צורת חיבור האנטה לצ'יפ, הצ'יפ עצמו שבשימוש (מתוצרת חברה אחרת) ומערכת ההפעלה שצרובה עליו.

כך למשל, יצרנית אחד מסוגי הכרטיסים הנפוצים בישראל כיום היא חברת ASK הצרפתית (לאחרונה שינתה את שמה ל-Paragon ID). ASK אחראית על מערכת הפעלה בשם TanGo עליה מבוססים כרטיסי ASK TanGo. סדרת כרטיסים אחרת וותיקה יותר שהיא מפיצה בנוסף לכך היא ה-CD21, שעליה מבוססות גם לא מעט סוגי כרטיסים אחרים.

## הצ'יפ

גם סדרת TanGo וגם סדרת CD21 של ASK משתמשות במשפחות microcontrollers מתוצרת חברת STMicroelectronics כגון ST23 או ST19.



מבנה הצ'יפ ST23, מתוך האתר של STMicroelectronics

סדרות מתוצרת Watchdata ו-Gemalto, לעומת זאת, משתמשים בצ'יפים של יצרנית האלקטרוניקה הגרמנית Infineon ה-SLE77 או ה-SLE66, ומריצות את מערכת ההפעלה TimeCOS של חברת Watchdata.

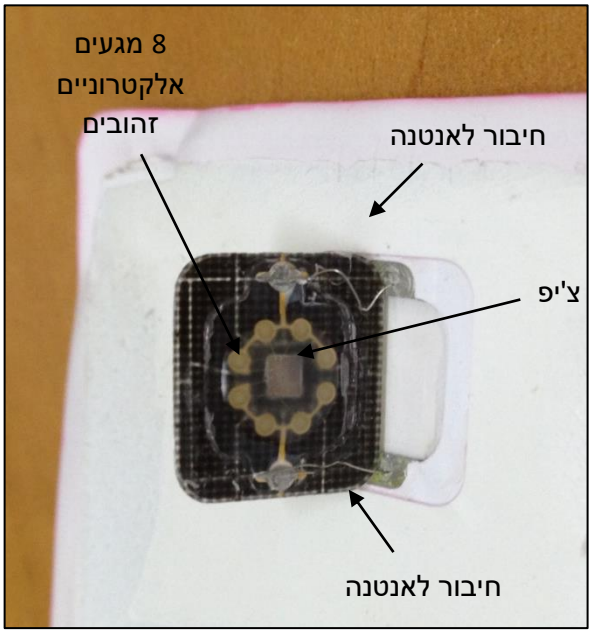
בדרך כלל מדובר על מעבד של 8 או 16 ביט ייעודי עם מהירות שעון של כ-MHz 30 וזכרון EEPROM של כ-80KB - 2.

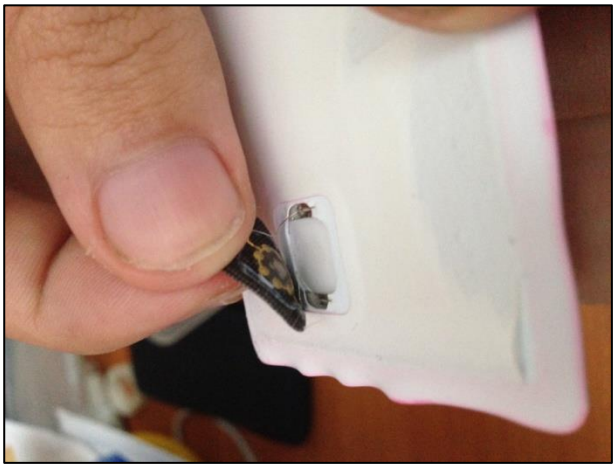
## דוגמה - חילוץ הצ'יפ מהכרטיס

על מנת להמחיש את המרכיבים הפיזיים של כרטיס חכם, נראה מה מגלים כאשר מפרקים אותו. בשלב ראשון, נשרה את הכרטיס באצטון ונקלף את שכבת המיתוג מפלסטיק:

[הורדת השכבה החיצונית ביותר של כרטסי רב קו]

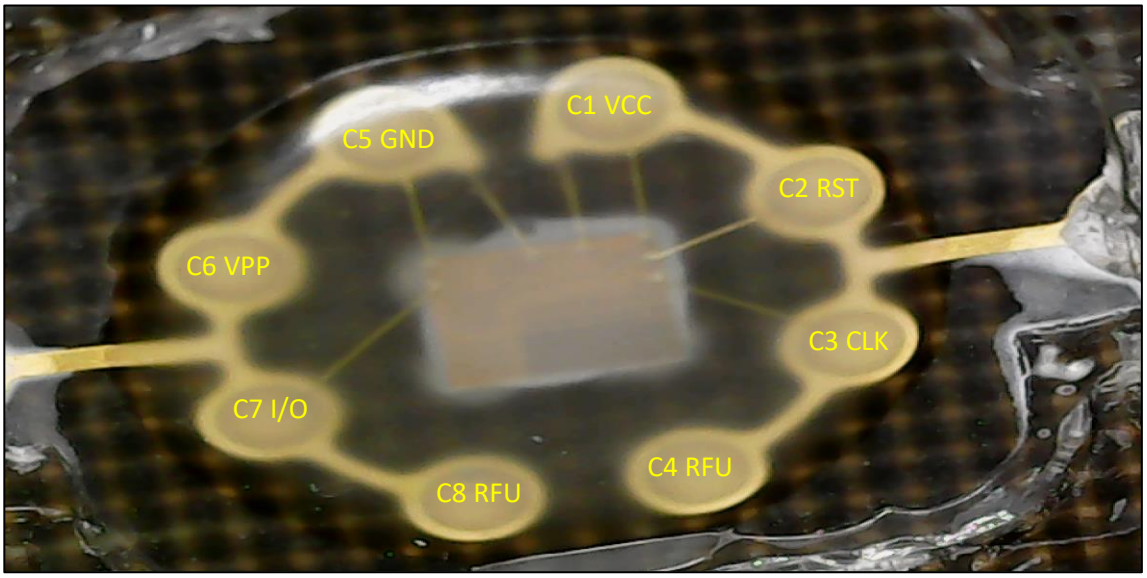
לאחר מכן נקלף את המודול האלקטרוני הזהוב עם האצבעות או בעזרת סכין חדה. נהפוך את המודול ונגלה את הצ'יפ שנמצא מאחוריו:



[הרמת המלבן האלקטרוני הזהוב שעל הכרטיס]

נסתכל על האזור הנ"ל מקרוב יותר בעזרת מיקרוסקופ, ונראה שהוא אכן תואם לתצורת הפינים ע"פ תקן ISO 7816-3:



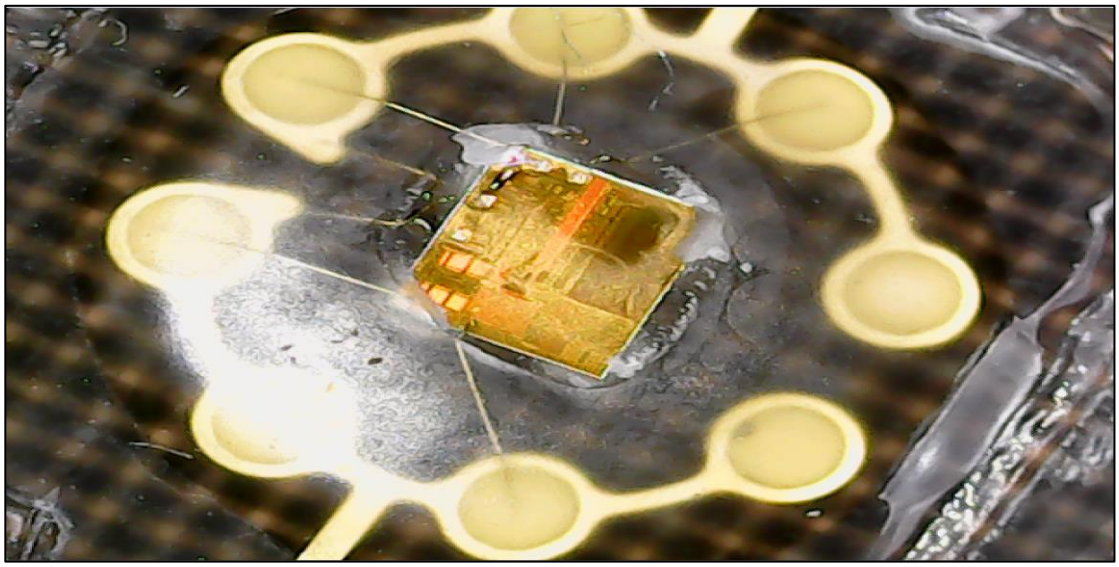

[ה-pinout האלקטרוני של צ'יפ רב קו מסוג Watchdata (מבעד למיקרוסקופ).ניתן לראות כיצד הפינים מחוברים לצ'יפ עצמו באמצעות חוטי חשמל זהובים.]

מפה:

- **VCC**: Supply Voltage, כ-5 או 3.3 וולט של Direct Current (DC)
- **RST**: Reset Signal
- **CLK**: חיבור לסיגנל השעון החיצוני של הכרטיס, קובע את קצב העברת המידע
- **GND**: Ground (הארכה) או Reference Voltage

- **I/O**: הסינגל שעל גביו מקודדים את המידע עצמו (APDUs), לכרטיס וממנו.
- **VPP**: במקור היה אמור לספק זרם למחיקה ותכנות מחדש (Erase/Programming Voltage) של ה-EEPROM. החל משנות ה-90 המוקדמות, זרם זה לרוב מסופק ישירות ע"י הצ'יפ, מה שהפך פין זה למיותר. כיום אין לו שימוש אמיתי בד"כ אבל הוא חייב להיות קיים.
- **RFU**: פינים שהם "Reserved for Future Use".

בשלב זה נשרה את הצ'יפ באצטון שוב במידת הצורך ונקלף בזהירות שכבות "דבק" (epoxy resin) על גבי הצ'יפ (אם יש כאלה) - לקבלת הסיליקון.

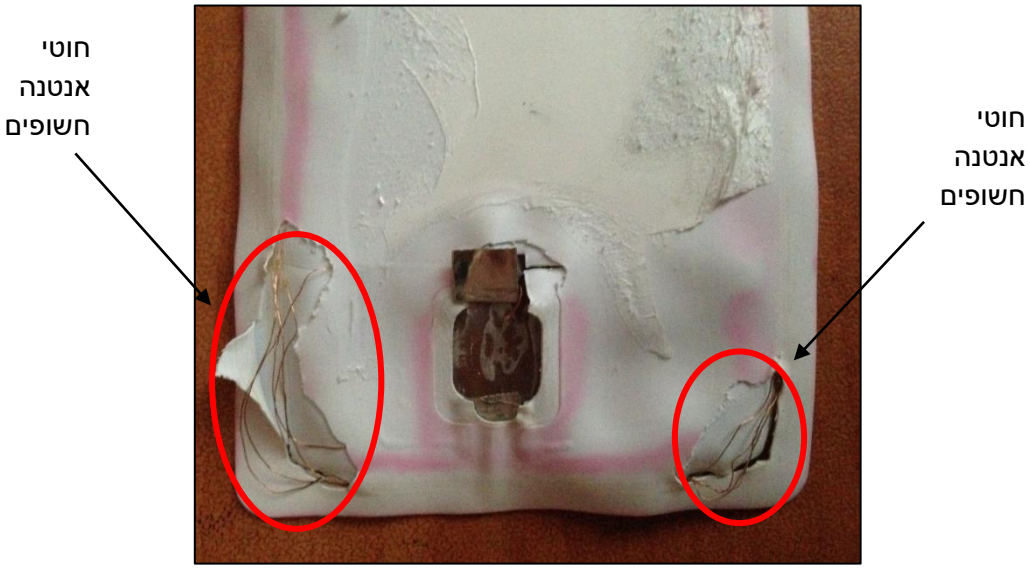


[אותו צ'יפ של Infinion (SLE77) עם שכבת הדבק מוסרת]

כעת, כאשר נסתכל אפילו מקרוב יותר, נוכל להבחין ברכיבים השונים ובשכבות הסיליקון שמרכיבות את הצ'יפ:

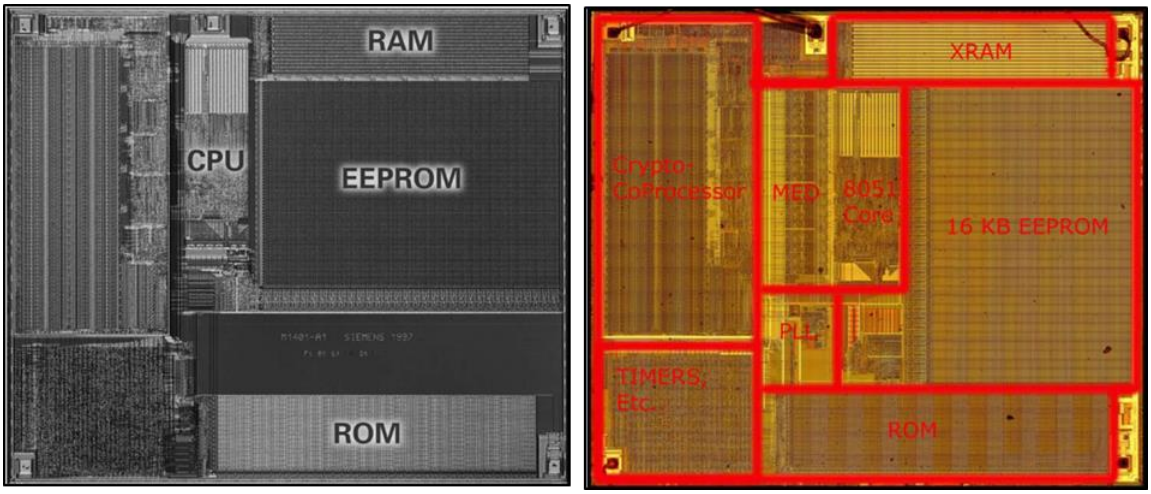
[הצ'יפ SLE77 בתצלום קירוב (הגדלה של כ-x1000).]

בנוסף לכך, אם נקלף עוד שכבות בכרטיס עצמו, נוכל לגלות את סלילי האנטנה שמשתרעים לאורך ההיקף ומתחברים לצ'יפ:




**דוגמה נוספת**

נסתכל על דוגמה לדגם ספציפי של microprocessor מתוצרת Infineon שאפשר למצוא בחלק מכרטיסי הרב קו של Watchdata: ה-SLE66CLX800PE.




[תמונות מיקרוסקופיות של דגמים ישנים מאותה הסדרה, כמו ה- SLE 66CX160S (משמאל) מתוך Smart Card Handbook ו-Black Hat 2008]

באופן כללי, סדרת ה-SLE 66 היא משפחת controller-ים של 8 או 16 ביט שמריצה גירסה מורחבת ו-proprietary של ארכיטקטורת 8051 של אינטל. היא נמצאת בשימוש במוצרים רבים נוספים, כדוגמת כרטיסי SIM.

ובמקרה הספציפי של SLE66CLX8000PE מדובר על צ'יפ בעל המאפיינים הבאים:

- 240KB של ROM
- 1.7KB של RAM
- 80KB של EEPROM (מכאן שם הצ'יפ)
- מתח עבודה:1.62V - 5.5V
- יחידות DES, RSA ואפילו Elliptic Curve Cryptography
- יחידת CRC
- יחידת True Random Number Generator
- מהירות שעון עד 30MHz

סדרה זו כוללת גם פיצ'רי הגנה מפני מתקפות פיזיות:

- EEPPROM ו-ROM שמוצפנים בצורה ייחודית עבור כל צ'יפ
- Bus Scrambling
- חיישני תאורה וטמפרטורה שמנטרלים את הצ'יפ כשמישהו פותח אותו
- "Active Shield" שמהווה שכבת הגנה פיזית מעל המעגל האלקטרוני
- רנדומליזציה אוטומטית של ה-Power Profile - כנגד מתקפות Power Analysis.
- ועוד

## התוכנה

אחרי שהבנו על איזה סוג של רכיב אנחנו מדברים, הגיע הזמן להבין מה הדבר הזה מריץ ואיך זה עוזר לעולם.

### קליפסו

כאמור, הממשק התוכנתי של רב-קו מבוסס על תקן לכרטיסי תחבורה ציבורית בשם Calypso ("קליפסו"), ולכן הוא חשוב לנו על מנת להבין איך נראה המידע ששמור על הכרטיס ואיך הוא מגן עליו.

בעקרון מדובר על תקן אירופאי שמגדיר את הפקודות, את צורת שמירת המידע ואת האבטחה על כרטיסים חכמים בכל מה שקשור לטיפול בתחבורה ציבורית. התקן מתבסס על ISO 7816 ו-ISO 1443 לשמירת המידע על הכרטיס ולפקודות השונות.

Calypso נוסד ב-1993 בצרפת (בשיתוף פעולה בין חברת התחבורה בפריז **RATP** וחברת הכרטיסים החכמים **Innovatron**), צבר תאוצה באירופה בתחילת שנות האלפיים ולאחר מכן אומץ ע"י ישראל ב-2007.

הגירסה האחרונה של התקן היא revision 3.2 שיצאה ב-2013, אבל הגירסה הנפוצה בעולם היא דווקא revision 2.4, שבה רוב הרב-קווים כיום משתמשים.

### דוקומנטציה

על מנת למצוא את מאפייני התקן אפשר להשתמש באחד משני אתרי האינטרנט של הסטנדרט: האתר הכללי (CNA ,Calypso Network Association) והאתר הטכני (Calypso Technical Standard).

באתר הטכני ניתן למצוא מספר מסמכים פומביים שמתארים את מאפייני התקן בכלליות, אבל רוב מסמכי התקן הטכניים באמת והמפורטים יותר הם **מסווגים** כך שלאזרח הממוצע קשה יותר להבין בדיוק מה קורה בכרטיס מאחורי הקלעים - במיוחד באזורי האבטחה.

הסטנדרט והאתר הטכני מנוהל ומפותח כיום ע"י ארגון בשם *Spirtech*.

אבל לפני שנכנס לפרטים של Calypso ונראה מה הוא קובע באשר לפרוטוקול האבטחה של רב-קו, נצטרך קודם כל להבין פרוטוקולים אחרים ובסיסיים יותר שקובעים את הנתונים של רוב הכרטיסים החכמים בעולם.

## תקני ISO בסיסיים

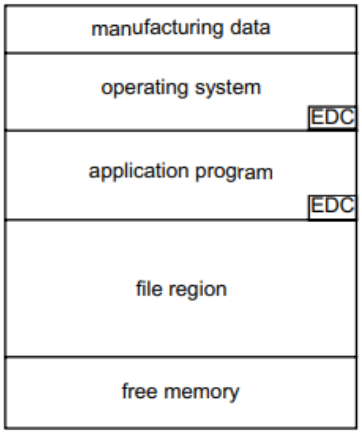
באופן כללי, כשמדברים על כרטיסים חכמים יש שני תקנים בינלאומיים רלוונטיים שמסבירים לנו איך הם עובדים:

- ISO/IEC 7816: התקן לכרטיסים חכמים בעלי ממשק מגע אלקטרוני (with contacts) שניתנים לקריאה ע"י קורא כרטיסים שולחני.
- ISO /IEC 14443: התקן לכרטיסים חכמים שנשלטים "מרחוק" באמצעות NFC ( contactless smart cards).

התקן ISO 7816 הוא מה שמעניין אותנו בעיקר מכיוון שהוא זה שמפרט איך נראה המידע ששמור על הכרטיס (לוגית) ואת הפרוטוקול האפליקטיבי שמשמש כדי לומר לכרטיס מה לעשות.

ספציפית, המידע הנ"ל נמצא בחלק הרביעי של התקן ("*Part 4: Organization, security and commands for interchange*").

## ארגון המידע על הכרטיס




[דוגמה מופשטת לחלוקה אפשרית של המידע על גבי ה-EEPROM (מתוך The Smartcard Handbook)]

המידע ששמור ב-EEPROM לא שוכב שם סתם כך אלא מאורגן בצורה מסוימת, מן הסתם.

הכרטיס שומר מידע על החוזים שטעונים כרגע לנוסע, הזכויות שצבורים בהם, וגם מידע נוסף כפי שנראה בהמשך.

## קבצים

התקן ISO 7816 (שעליו קליפסו מתבסס) קובע שצורת האחסון של מידע על כרטיסים חכמים היא בצורה של **קבצים**, ובצורה הזו גם ניתן להתייחס אליו.

על פי התקן, יש שני סוגי "קבצים" שנתמכים:

- **Dedicated File (DF)** - קובץ שמשמש פשוט כתיקייה, כלומר קבצים אחראים יכולים להיות הבנים שלו. DF-ים יכולים לייצג אפליקציות או מידע אחר.
- **Elementary File (EF)** - מכיל מידע. המידע ניתן לשליפה בצורה של אחד מכמה מבני נתונים אפשריים: raw, Tag-Length-Value, records.

התקן מגדיר גם עוד מינוח אחד:

- **Master File (MF)** - כינוי ל-DF שהוא ה-root של מערכת הקבצים. המזהה שלו הוא תמיד 0x3F00.

אצל קליפסו המידע בתוך הקבצים שמור בצורת records - פיסות מידע קטנות בגודל קבוע או משתנה. ה-record-ים בתוך כל קובץ ממוספרים מ-1 ועד מספר ה-record-ים באותו קובץ, וכך גם מתייחסים אליהם.

בכרטיסי רב קו, הגודל של כל ה-record-ים שמכילים מידע מעניין הוא 29 בייטים בדיוק.

נציין גם שלכל קובץ משויך metadata ("File Control Information - FCI") שניתן לשליפה ע"י פקודת SELECT FILE (עוד על פקודות בהמשך).

ה-metadata כולל מידע כגון:

- ה-ID הארוך של הקובץ ("LFI" - Long File Identifier, 2 בייטים)
- ה-ID הקצר שלו ("SFI" - Short File Identifier, 5 ביטים)
- סוג הקובץ ( DF או EF),
- הרשאות - Access Conditions (באילו modes ניתן לקרוא ולכתוב אליו)
- מספר וגודל ה-record-ים שבו אם קיימים
- כו'.

## אפליקציות

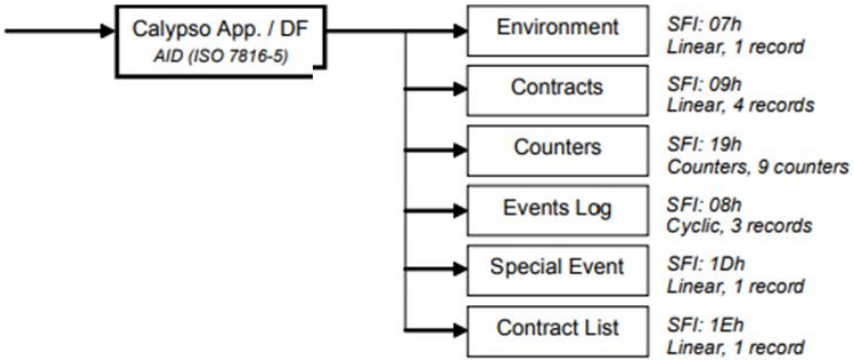
כרטיסי קליפסו כוללים את האפשרות לתמוך ביותר מאפליקציית קליפסו אחת באותו כרטיס. למשל, ניתן תיאורטית להשתמש באותו כרטיס גם בשביל תחבורה וגם בשביל ארנק אלקטרוני (EP) או אפליקציה תומכת Calypso אחרת.

לכן, DF-ים בכרטיס משויכים עם אפליקציות מסוימות, כאשר לכל אפליקציה משויך Application Identifier ייחודי (AID). טווחי ה-AID-ים שקיימים לכרטיסים חכמים בעולם מנוהלים על ידי התקן ISO 7816 - 5.

על כן, בתקן Calypso מוגדרת פקודה מיוחדת לבחירת אפליקציות (SELECT APPLICATION), שהיא למעשה כמו פקודת SELECT FILE על ה-DF הרלוונטי, אבל מחזירה גם מידע ייחודי על האפליקציה שנבחרה.

**הקבצים על הכרטיס** (או: מה שמור על הרב-קו שלי)

במסמכים הפומביים של Calypso ניתן למצוא את הדרישה לחלוקה *הכללית* של סוגי הקבצים שחייבים להופיע בכרטיסים שלהם והסבר על המטרה של כל אחד מהם:




[מתוך המסמך של Calypso]

ניתן לראות שהתקן מחייב הימצאות של לפחות ששת הקבצים האלה, תחת האפליקציה הראשית:

1. Environment - קובץ שמכיל מידע כללי על בעל הכרטיס וכו'
2. Counters - קובץ שסופר זכויות נסיעה
3. Contracts - קובץ שמכיל מידע על החוזים שטעונים כרגע
4. Events Log - קובץ שמכיל לוג אירועים אחרונים
5. Special Event - קובץ שמכיל לוג אירועים מיוחדים אחרונים
6. Contract List - קובץ שמכיל רשימת חוזים פעילים

להלן מבנה עץ התיקיות על הכרטיס:

```
├──0002
│       ICC
├──0003
│       ID
├──2000
│   ├──2001
│   │       Environment
│   ├──2004
│   │       AID
│   ├──2010
│   │       Event Logs
│   ├──2020
│   │       Contracts
│   ├──2040
│   │       Special Events
│   ├──2050
│   │       Contract list
│   └──2069
│           Counters
├──2100
│       RT2
└──2F10
        Free
```

להלן פירוט הקבצים ותוכנם בכרטיסי רב קו:

- (DF, Path: /2000) "Ticketing"/"RT": התיקייה הראשית שתחתיה נשמרים הקבצים של אפליקציית Calypso.
- **AID** (EF, Path: /2000/2004): מכיל את ה-Application Identifier של אפליקציית Calypso, "TIC.ICA1"
- **Environment** (EF, Path: /2000/2001, SFI: 07h): מידע כללי על הכרטיס.
  - נתונים על הנפקת הכרטיס - החברה המנפיקה, תאריך ההנפקה, תאריך התפוגה של הכרטיס, מספר סידורי, סוג אמצעי תשלום
  - מידע על בעל הכרטיס - תאריך לידה, תעודת זהות, ארגון.
  - רשימת פרופילים פעילים (המקנים הנחות): מבוגר / סטודנט / נוער / משטרה / כו'. בכל רגע יכולים להיות רשומים בכרטיס עד 2 פרופילים.

- **Contracts** (EF, Path: /2000/2020, SFI: 09h): מכיל מידע על חוזים שטעונים לכרטיס.
  כל record בקובץ זה מייצג חוזה. פרטים שנשמרים על כל חוזה הם:
  - סוג החוזה (Extended Ticket Type - ETT) - חופשי חודשי, חופשי יומי, כרטיסיות למספר נסיעות מוגבל (2, 5, 15, 20...), וכמובן "ערך צבור".
  - תוקף החוזה - ממתי עד מתי החוזה תקף, באילו אזורים, באילו מפעילים
  - נתונים על קניית החוזה - מאיזה מכשיר הוא נקנה, מתי, מאיזה מפעיל
  - נתוני תעריפים
  - האם כולל אופציה לנסיעות המשך
  - משך הזמן המוגבל לנסיעות המשך (בדרך כלל 90 דקות)
  - ועוד.

- **Contract list** (EF, Path: /2000/2050, SFI: 1Eh): "אסמכתאות"/רשימת חוזים - מכיל record אחד שאמור להיות כתוב בו אילו חוזים פעילים כרגע, עם ציון עדיפויות.

- **Event Logs** (EF, Path: /2000/2010, SFI: 08h): יומן אירועים - כל record בקובץ זה מייצג אירוע שקרה עם הכרטיס. פרטים שיכולים להישמר עבור כל אירוע:
  - סוג האירוע ("circumstances") - נסיעה (חיוב), נסיעת מעבר, טעינת חוזה, טעינה ושימוש מיידי, הנפקת כרטיס וכו'
  - לאיזה חוזה רלוונטי
  - כלי התחבורה (אוטובוס עירוני/בין-עירוני, רכבת/רכבת קלה, ...)
  - מספר הקו
  - זמן האירוע, גודל החיוב, קוד המחיר
  - החברה המפעילה (אגד/מטרופולין/דן/...)

- קוד תחנת העלייה (ע"פ ה-GTFS), מספר הנוסעים, מספר הרכב (!)

כאשר נגמר המקום בקובץ זה, האירוע הישן ביותר נמחק ממנו (כי הוא Cyclic).

- **Special Events** (EF, Path: /2000/2040, SFI: 1Dh): יומן אירועים "מיוחדים" - קובץ זה כולל record-ים שמייצגים אירועים חריגים שקרו עם הכרטיס, כמו שגיאות במהלך נסיונות חיוב.
- **Counters** (EF, Path: /2000/2069, SFI: 19h): מונים עבור כל החוזים - זהו קובץ עם record אחד שמכיל 9 ערכים ספירים, כאשר כל אחד מהם רלוונטי ספציפית לחוזה מסוים (ע"פ אינדקס). ערכים ספירים הם, לרוב:
  - כמות הנסיעות שנותרו, או במקרה של חוזה "ערך צבור" - **כמה כסף נותר בכרטיס.** גודל כל ערך ספיר הוא 3 בייטים בדיוק.

> ***הערה:*** *זהו קובץ ה-Counters האמיתי היחיד. שאר קבצי ה-Counters הם Simulated Counters שמשקפים את התוכן של קובץ זה בתאימות ל-Calypso 1.*

כפי שניתן לראות כרטיס הרב קו אכן מכיל את הקבצים שנדרשים על פי קליפסו, אבל מבנה מערכת הקבצים המדויק ("File Structures") נתון לבחירת המשתמש (=משרד התחבורה).

במפרטי הכרטיס CD97 מוצעים שמונה מערכות קבצים אפשריים. משרד התחבורה עושה שימוש ב- File Structure מספר #6. לכן:

- ישנם שתי אפליקציות Calypso על הכרטיס: "Ticketing" ו-"RT2" (לא בשימוש).
- בקובץ Environment יש שני record-ים
- בקובץ Event Logs יש מקום לעד 6 אירועים (records) אחרונים
- בקובץ Contracts יש מקום לעד 8 חוזים (records)
- בקובץ Special Events יש מקום לעד 4 אירועים (records) אחרונים
- בכל קובץ Counter יש record אחד

כמו כן מבנה מערכת הקבצים שבחר משרד התחבורה (#6) מגדיר גם קבצים שלא נחשבים הכרחיים במפרטי Calypso, ב-root של מערכת הקבצים:

- **ICC** (EF, Path: /0002, SFI: 02h): מידע על ייצור הכרטיס - פורמט המידע בקובץ זה תלוי בסוג הכרטיס המדויק (למשל, בכרטיסי CD21 הוא עשוי להכיל אפסים). הקובץ כולל בדרך כולל את המספר הסריאלי של הכרטיס, היצרן, מדינת הייצור וכדומה.
- **Free** (EF, Path: /2F10, SFI: 01h): קובץ חופשי - זהו הקובץ היחיד שתמיד ניתן לקרוא ולכתוב ממנו ללא הגבלות.
- **ID** (EF, Path: /0003, SFI: 03h): מידע אישי - אמור לכלול מידע רגיש על המשתמש שהקריאה אליו מוגנת באמצעות PIN. לא ברור אם רב קו עושה שימוש בקובץ הזה (נראה שלא).

- **RT2** (DF, Path: /2100): אפליקציית קליפסו השנייה - זוהי תיקייה שמכילה קבצים מקבילים לגמרי במשמעותם לאפליקציה "RT" (אך עם LFI-ים שונים), במטרה לתת לכרטיס אפשרות להכיל עוד סוג "כרטיס" לשימוש אחר. מכיוון שרב קו לא עושה שימוש באפליקציה הזו, כל הקבצים בה מכילים אפסים כברירת מחדל - חוץ מקובץ ה-AID (2100/2104/), שמכיל "TIC.ICA21".

### מבנה הנתונים ב-record-ים

על פי תקן קליפסו, פורמט המידע המדויק שמשמש כדי לייצג אינפורמציה בכרטיס תלוי ב"מודל הנתונים" שהגדיר המפעיל. מודל הנתונים בישראל מבוססים על "מפרטי קליפסו הישראליים" - אוסף מסמכים שמבוססים על תקן קליפסו המקורי והותאמו לישראל. מסמכים אלו מוגדרים *כמסווגים* גם כן ולא ניתן לקבלם אלא לאחר חתימת הסכם NDA.

עם זאת, כמובן שעצם הסיווג של המסמכים לא עצר אנשים מלפרסר את הפורמט בכל זאת. כמו כן, באתר של משרד התחבורה ניתן למצוא מסמכים רבים שמפרטים את המשמעויות של חלק מהשדות בקבצי הכרטיס, וגם כוללים טבלאות שממירות ערכים מספריים לתיאורים מילוליים (רלוונטי לחלק מהשדות).

להלן דוגמה לפורמט:

דוגמה 7

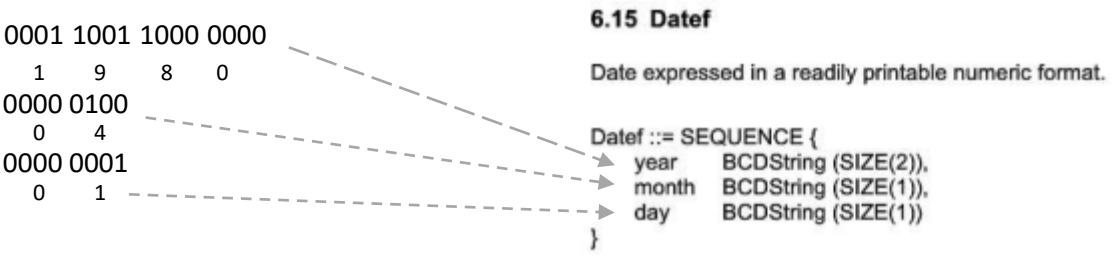
| דוגמה לאמצעי כרטוס מסוג "ערך צבור" – משותף | | |
|---|---|---|
| שדה | ערך | bits |
| version number | גרסה 000 | 000 |
| country id | קוד ישראל = 376h | 001101110110 |
| issuer id | על פי טבלת מפעילים (ג'י. בי. טורס =8) | 00001000 |
| application number | מספר סידורי רציף בתוך מפעיל 8 | 00000000000000000000000001 |
| issuing date | תאריך הנפקת הכרטיס 1/4/2005 = 3012 | 00101111000100 |
| end date | 8 שנים לאחר הנפקה 1/4/2013 = 5934 | 11011010111010 |
| pay method | תשלום חד פעמי בעת רכישה | 001 |
| birth date | תאריך לידה – (עשרוני) 1980/04/01 | 00011001100000000000010000000001 |
| company | לא מוגדר | 00000000000000 |
| company id | לא מוגדר | 00000000000000000000000000000000 |
| id number | ת.ז. = תשעיות | 11101110011010110010011111111111 |
| profile 1 | נוסע רגיל/לא מוגדר | 000000 |
| profile 1 date | לא נדרש | 00000000000000 |
| profile 2 | 0 | 000000 |
| profile 2 date | לא נדרש | 00000000000000 |

[דוגמא ל-record הראשון של קובץ ה-Environment. מתוך "אוגדן נהלי כרטוס", מהדורה 02]

באופן כללי, השדות במבנים השונים (Contract, Event, Environment, ...) מבוססים במידה כזו או אחרת על תקני Intercode (NF P 99-405) הצרפתיים, אך כאמור פורמט הנתונים מוגדר באופן מלא רק במפרטי קליפסו הישראליים.

בנוסף, הכרטיס עושה שימוש מסוים בתקן EN-1545 הצרפתי שמפרט את הקידוד הבינארי של סוגי נתונים שונים (תאריך, שעה, מטבע, יום בשבוע, ...) על כרטיסים חכמים.

למשל:




*[הפורמט שבו רב-קו משתמש כדי לקודד את תאריך הלידה של המשתמש (מקובץ ה-Environment), על פי EN-1545]*

## המחשה #1 - הצגת הנתונים השמורים בכרטיס

בחלק זה נראה קצת יותר בפירוט איך אפשר לדבר ולראות את תוכן הכרטיס הזה בעולם האמיתי.

כאמור, מכיוון שכרטיסי רב-קו נוטים לבוא גם עם ממשק contactless וגם עם ממשק contact (מה שנקרא dual-interface), על מנת לקרוא מידע גולמי מהכרטיס קיימות שתי אפשרויות ריאליות: להשתמש בטלפון תומך NFC או להשתמש בקורא כרטיסים חכמים מסוג PC/SC שמתחבר למחשב (שעובד על מגע או על NFC). אני בחרתי באפשרות השנייה.

[קוראי כרטיסים חכמים למחשב (Gemalto IDBridge CT30 , Identiv uTrust 3700)]

כעת נרצה להשתמש בתוכנות שמאפשרות פרסור נתונים\תקשורת עם הכרטיס.

על מנת להציג את המידע ששמור על הכרטיס באופן נח מומלץ להשתמש בתוכנת הקוד הפתוח הנהדרת cardpeek, שיכולה לפרסר נתונים של מגוון סוגי כרטיסים: SIM, EMV, Navigo, MIFARE, e-passports, ואפילו רב-קו (במידה חלקית).

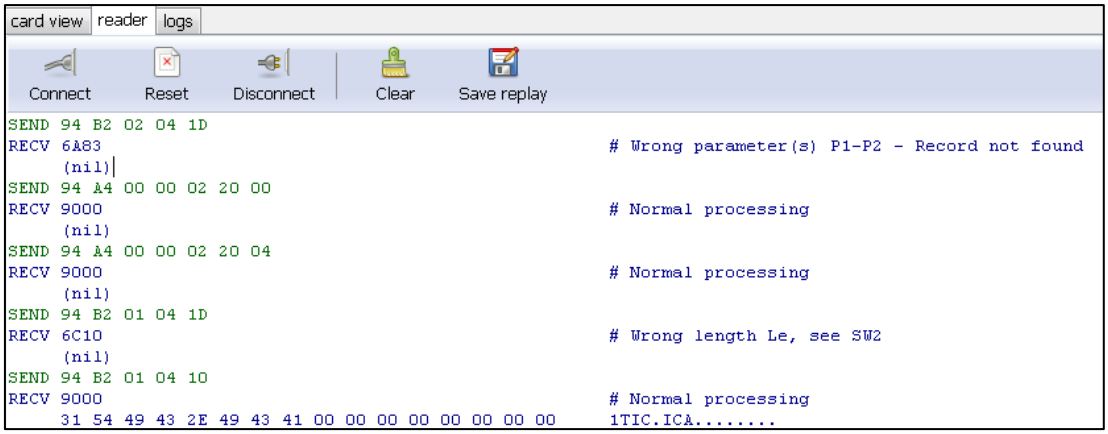
תוכן הכרטיס נראה כך ב-cardpeek:

[דוגמה לחלק מתוכן כרטיס רב-קו טיפוסי כפי שנראה בתוכנה. המשתנה "Balance" לא באמת נמצא בקובץ ה-Contracts אלא נלקח מתוך קובץ ה-Counters]

כפי שניתן להתרשם מכמות הביטים שתופס כל שדה, מדובר בפורמט בינארי יעיל למדי (כפי שראינו בחלק על הפורמט).

*הערה*: פרסור התוכן של כרטיסים בתוכנה זו מתבצע באמצעות סקריפטי LUA. ניתוח של הסקריפטים הרלוונטיים לרב-קו והמחרוזות שלו יכול לעזור משמעותית בהבנת הפורמט.

פיצ'ר שימושי נוסף ב-cardpeek הוא האפשרות לראות מה היא עושה מאחורי הקלעים ע"י הצגה של פקודות ה-APDU שמועברות בין קורא הכרטיסים לכרטיס, במסך ה-reader:

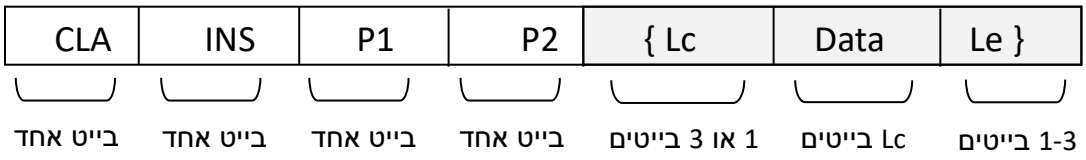


## פרוטוקול התקשורת של כרטיסים חכמים

אז איך מדברים עם כרטיסים חכמים?

הן בזמן תקשורת contactless (קרינה אלקטרומגנטית) והן בזמן תקשורת contact (מגעים אלקטרוניים), כרטיסים חכמים עושים שימוש באותו פרוטוקול אפליקטיבי.

ע"פ ISO 7816-4, הפורמט הכללי של כל פקודות ה-APDU שנשלחות לכרטיסים חכמים נראה ככה:



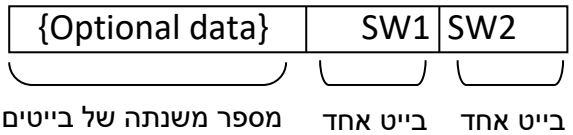

נסביר על כל אחד מהחלקים:

- **CLA**: "מחלקת ההוראה" (תעשייתי, proprietary וכו'). התקן מגדיר כל מני ביטים עם משמעויות שונות בתוך השדה הזה (כמו secure messaging). פרקטית, במקרה של רב-קו הערך של השדה הזה הוא לרוב 0x94.
- **INS**: ההוראה עצמה כגון SELECT FILE (0xA4), READ RECORDS (0xB2) וכו'.
- **P1**: הפרמטר הראשון בשביל הוראה המבוקשת. מן הסתם המשמעות של הפרמטר משתנה בהתאם למשמעות הוראה.
- **P2**: הפרמטר השני בשביל הוראה המבוקשת, אם יש.

כל השדות של החלק השני של ה-APDU ("body") הם אופציונליים ומועברים לכרטיס רק אם יש עוד מידע בגודל משתנה שרוצים להעביר בנוסף לפרמטרים, או שיש צפי למספר כלשהוא של בייטים שיחזור כתגובה מהכרטיס.

- **Lc**: אורך ה-Data שרוצים להעביר (בבייטים). *הערה: יש פה קידוד של extensible integer כך שאם רוצים להעביר יותר מ-255 בייטים, שמים 0 ואז שני הבייטים הבאים יהוו את האורך.*
- **Data**: מידע שרוצים להעביר, בנוסף לפרמטרים.
- **Le**: מספר הבייטים המקסימלי שמצופה מהכרטיס להחזיר כתגובה.

הפורמט של תשובות הכרטיס הוא פשוט הרבה יותר:



למעשה, פורמט התשובה פשוט מורכב ממידע אופציונלי שמוחזר לקורא (במידה והוא ביצע פקודה שמחזירה מידע) ועוד status word (SW1 + SW2) שנועד לומר לקורא הכרטיסים האם הפעולה שהוא ניסה לבצע הצליחה, ואם לא (כלומר קרתה שגיאה כלשהיא) מהי השגיאה.

התקן כמובן מפרט על כל הערכים האפשריים של SW1 ו-SW2 ומה המשמעות של כל אחד מהם. בקצרה, 0x9000 מציין שהפעולה הצליחה (Normal Processing), והערכים 0x67XX עד 0x0x6FXX מציינים שגיאה כלשהיא במידע שהתקבל.

פירוט של כלל ערכי השגיאה האפשריים אפשר למצוא כאן (לא כולל שגיאות ספציפיות ל-Calypso).

## פקודות של כרטיסים חכמים (בקצרה)

התקן ISO 7816-4 מגדיר בסך הכל 39 פקודות שונות, אבל מציין גם שלא כל הכרטיסים החכמים חייבים לתמוך בכולן או בכל האפשרויות של כל אחת מהן.

חמשת הפקודות הנפוצות ביותר שמשמשות כדי לקרוא ולכתוב מידע מהכרטיס הן:

- **SELECT FILE (INS=0xA4)**: בוחר "קובץ" (Elementary file, EF) או "תיקייה" (Dedicated file, DF) על פי ID, נתיב או שם מסוים, כפי שמפורט בפרמטרים P1 ו-P2. הקובץ שנבחר הופך להיות הקובץ הנוכחי והפקודות הבאות יכולות להתייחס אליו.
- **READ RECORDS (INS=0xB2)**: קורא ומחזיר record-ים מ-elementary file מסוים (במידה ואותו קובץ אכן מפורמט בצורת records).
  - P1 מכיל את מספר ה-record שממנו מתחילים לקרוא
  - P2 מכיל את ה-SFI של הקובץ המבוקש (00 = הקובץ הנוכחי) + דגלים
- **UPDATE RECORD (INS=0xDC)**: מעדכן את התוכן של record קיים בתוכן חדש.
  - P1 מכיל את מספר ה-record שנרצה לעדכן
  - P2 מכיל את ה-SFI של הקובץ בו ה-record קיים (00 = הקובץ הנוכחי) + דגלים
- **APPEND RECORD (INS=0xE2)**: מוסיף record חדש ל-elementary file קיים.
  - P1 מכיל 00
  - P2 מכיל את ה-SFI של הקובץ שאליו ה-record יתווסף (00 = הקובץ הנוכחי) + דגלים.
- **GET RESPONE (INS=0xC0)**: פקודה שזמינה רק במצב contact ומשמשת על מנת לקבל את התגובה של הפעולה הקודמת, או ליתר דיוק "תגובה שלא היה ניתן לקבל ב-transmission protocol הנוכחי" (ראו פירוט ב-ISO 7816 Part 3).

***הערה**: דוגמה בולטת למקרה שבו צריך להשתמש בפקודה הזו היא כאשר רוצים להשתמש בכל שלושת השדות של ה-body (גם לשלוח הרבה מידע וגם לקבל הרבה מידע), אבל לא ניתן לקודד את Le.*

כדי להמחיש מה עוד הכרטיס יכול לעשות, הנה פקודות נוספות שמוגדרות בתקן (לא כולן בהכרח נתמכות בכל סוגי הכרטיסים):

- INCREASE, DECREASE
- SEARCH BINARY ,READ BINARY, WRITE BINARY, UPDATE BINARY, ERASE BINARY
- GET DATA, PUT DATA
- CREATE FILE, ACTIVATE FILE, DEACTIVATE FILE, DELETE FILE
- SEARCH RECORD, WRITE RECORD
- VERIFY, GET CHALLENGE, MANAGE CHANNEL
- EXTERNAL AUTHENICATE , INTERNAL AUTHENTICATE

מלבד פקודות אלה כרטיסים חכמים יכולים גם כמובן לתמוך בפקודות שמיועדות ספציפית לסוג הכרטיס (למשל - לכרטיסי SIM יש פקודות משלהם) או בפקודות proprietary שלא מתועדות בתקנים ציבוריים, כפי שנראה אצל Calypso בהמשך.

## הסנפות

ב-Wireshark יש dissector חלקי של ISO 7816, ובעזרת USBPcap נוכל להסניף USB בזמן התקשורת עם הכרטיס ולראות בדיוק מה הפקודות שנשלחות אליו בזמן אמת. למעשה קוראי כרטיסים חכמים שמתחברים למחשב (PC/SC) מדברים עם המחשב מעל USB באמצעות פרוטוקול שנקרא USB CCID, ומעל הפרוטוקול הזה מקודדות פקודות ה-APDU של ISO 7816-4.

זה נראה כך, למשל, במקרה של פקודת SELECT FILE:

```
▲ ISO 7816 Command APDU
    [Response in frame 2]
  ▷ Class: Unknown (0x94)
    Instruction: Select file (0xa4)
  ▲ Parameters
    ▲ Parameter 1: 0x00 (selection control)
        Selection control: Select MF, DF or EF (0x00)
    ▲ Parameter 2: 0x00 (selection options)
        .... 00.. = File control information request: Return FCI, optional template (0x0)
        .... ..00 = Occurrence: First or only occurrence (0x0)
    Length field Lc: 0x02
    APDU Body: 2000
```

[הפקודה הראשונה ש-cardpeek שולח לכרטיס כדי לבדוק שהוא עונה לבקשות SELECT]

### דיבור תכנותי ו-ATR

אם נרצה לשלוח לכרטיס פקודות תכנותית באמצעות סקריפט, נוכל להשתמש בספריית pyscard או בספריית card לפייתון. לדוגמה:

```
>>> from card.ICC import ISO7816
>>> ravkav_card = ISO7816(0x94)
>>> r = ravkav_card.SELECT_FILE(0,0,[0x20,0x20]) # contracts file
```

אני השתמשתי בספרייה זו, למשל, על מנת לבצע אוטומציה של סריקת פקודות ומערכת קבצים בכרטיס. למעשה, עוד לפני שנשלחת הפקודה הראשונה (בקוד שלמעלה, בעת יצירת האובייקט ISO7816), הכרטיס תמיד מעביר לקורא פיסת מידע שנקראת Answer to reset (ATR) בעת החיבור הראשוני.

ה-ATR נועד לכלול מידע ראשוני על הכרטיס כגון ה-transmission protocols הנתמכים (T=0, T=1) וגם "historical bytes". במקרה של כרטיסי קליפסו ה-historical bytes כוללים לפעמים את המספר המזהה של הכרטיס (8 בייטים שגם מודפס עליו פיזית - בצורה דצימלית), כך שניתן לזהות אותו חד-חד ערכית בעת החיבור, וגם מידע על סוג הכרטיס הספציפי.

נבקש לראות את ה-ATR של הכרטיס שאיתו אנחנו עובדים:

```
>>> ravkav_card.ATR_scan()
```

ונקבל את התוצאה הבאה:

```
smartcard reader: Gemalto USB Smart Card Reader 0
smart card ATR is: 3B 6F 00 00 80 5A 0A 07 06 20 04 2C 02 73 BF DB
ATR analysis:
TB1: 0
TC1: 0
supported protocols T=0
T=0 supported: True
T=1 supported: False
        clock rate conversion factor: 372
        bit rate adjustment factor: 1
        maximum programming current: 25
        programming voltage: 5
        guard time: 0
nb of interface bytes: 2
nb of historical bytes: 15
None
historical bytes: 80 5A 0A 07 06 20 04 2C 02 73 BF DA 82 90 00
```

נשים לב שב-ATR של כרטיסי Calypso לפעמים מקודד מידע מעניין ביותר, מעבר למידע הסטנדרטי של ISO 7816-3. לדוגמה, ב-ATR שלעיל:

- 0A - מציין שסוג הצ'יפ שעל הכרטיס הוא ST16RF52
- 07 - מציין שהכרטיס תואם לתקן Extended Calypso application, Calypso revision 2
- 06 - מציין את מספר ה-File Structure (#6)
- 20 - מציין שייצרן התוכנה הוא Calypso Networks Association

- **04 2C** - מציינים את גירסאות התוכנה (ROM, EEPROM)
- **02 73 BF DA** - ה-4 בייטים התחתונים של המספר הסריאלי של הכרטיס (0041140186)

קומבינציית הנתונים הספציפית האלו, למשל, מתאימה לנתונים של כרטיס מסוג CD97-BX משנות ה-2000. זהו למעשה שקר, כי במקרה הזה מדובר בכרטיס CD21 מודרני שנמצא במצב תאימות ל-CD97. בכרטיסים אחרים (למשל Watchdata) נקבל נתוני אמת.

## תקשורת ללא מגע

בשימוש היום-יומי בכרטיס צורת התקשורת הנפוצה יותר היא כמובן ללא מגע - בעזרת NFC. פשוט מניחים את הכרטיס על משטח עם אנטנה מתאימה (בין אם מכונה ייעודית או סמארטפון) וניתן להעביר מידע.

התקשורת ללא מגע מתבצעת בתדר 13.56 MHz ומצופה לעבוד בטווח של לא יותר מ-10 סנטימטרים - כמו כל מכשיר NFC סטנדרטי.

הכרטיס מקבל את החשמל שדרוש לו על מנת לעבוד בעזרת השראה אלקטרומגנטית מקרינה זו.

התקן הרלוונטי (ISO 14443-2) מגדיר שני סוגי כרטיסים אלחוטיים (A ו-B, שנבדלים אחד מהשני בין השאר בשיטות ה-modulation שבשימוש), כאשר כרטיסי Calypso לרוב משתמשים ב-Type B. מהירות התקשורת בטכנולוגיה זו היא כמה מאות קילו-בייטים ביטים לשנייה.

באנדרואיד, מבחינה תכנותית, ניתן לתקשר עם כרטיסים חכמים באמצעות המחלקה IsoDep שמתאימה לתקן ISO 14443-4. בעזרת הפונקציה transceive אפשר לשלוח פקודות עם אותו פרוטוקול אפליקטיבי שראינו מקודם:

```
IsoDep tag = IsoDep.get(tagFromIntent);
tag.connect();

 byte[] SELECT = {
    (byte) 0x00, // CLA Class
    (byte) 0xA4, // INS Instruction
    (byte) 0x04, // P1  Parameter 1
    (byte) 0x00, // P2  Parameter 2
    (byte) 0x0A, // Length
    0x63,0x64,0x63,0x00,0x00,0x00,0x00,0x32,0x32,0x31 // AID
};
byte[] result = tag.transceive(SELECT);
```

בדרך זו ממומשת התקשורת עם הכרטיס באפליקציית רב-קו אונליין ואפליקציות אחרות שמציגות את החוזים והמידע האישי שעל הרב-קו.

## אבטחה

בשלב זה נשאלת כמובן נשאלת השאלה המתבקשת:

מה מונע מכל אחד לבצע פקודת INCREASE על קובץ ה-Counters ולשנות לעצמו את הערך הצבור, או - לחילופין, לבצע פקודת UPDATE RECORD על קובץ ה-Contracts ולשנות את סוג החוזה לחוזה שמאפשר נסיעות בלתי מוגבלות? הבה ננסה:

```
>>> r = ravkav_card.SELECT_FILE(0,0,[0x20,0x20]) # contracts file
>>> print ravkav_card.WRITE_RECORD(2, 4, [0x01]*29)

['WRITE RECORD apdu: 94 D2 02 04 1D 01 01 01 01 01 01 01 01 01 01 01 01 01 01 ... 01',
'sw1, sw2: 69 82 - checking error: command not allowed: security status not satisfied', ..]
```

קיבלנו שגיאת security.

אז זה לא עד כדי כך קל, כמובן. אם כך, איך מתבצע חיוב של הכרטיס (בזמן נסיעה באוטובוס) או טעינה מחדש שלו?

### הרשאות

כאמור, לכל קובץ מוגדרים סט הרשאות משלו (Access Conditions ב-FCI). ההרשאות תלויות בקטגוריית הפקודה שמנסים לבצע ובKey- שמשתמשים בו במידה ויש Secure Session פעיל.

להלן ההרשאות לכל קובץ כפי שמוגדרות ב-File Structure #6:

| פעולה מבוקשת / קובץ | Read | Update | Write* / Decrease | Append / Increase |
|---|---|---|---|---|
| ICC | Always | Never | Never / Key <= 1 | - |
| ID | PIN | Session, Key <= 2 | Never | - |
| Environment | Always | Session, Key <= 1 | Never | - |
| Events Log | Always | Session, Key <= 3 | Session, Key <=3 | Session, Key <=3 |
| Special Events | Always | Session, Key <= 3 | Never | - |
| Contract List | Always | Session, Key <= 3 | Never | - |
| Contracts | Always | Session, Key <= 2 | Session, Key <=3 | - |
| Counters | Always | Session, Key <= 2 | Session, Key <=3 | Session, Key <=2 |
| Free | Always | Always / Any Key | Always | - |

* פקודת WRITE RECORD יכולה להפוך ביטים ל-1 (Logical OR) בלבד.

מהטבלה לעיל אפשר לראות שיש הכרח בקיום של **Secure Session** כלשהוא כדי לכתוב לכרטיס מידע משמעותי.

### מודל האבטחה של Calypso - ה-Secure Session

הרעיון העיקרי ש-Calypso מציעים כדי למנוע מכל אחד לעשות ברצונו בכרטיס הוא קונספט ה- Secure Session. Secure Session מתחיל על ידי שליחת הפקודה *Open Secure Session* לכרטיס ומסתיים על ידי שליחת הפקודה *Close Secure Session*.

הכרטיס מאפשר כתיבה על record-ים מסוימים רק בזמן שיש Secure Session פעיל. לכן, כאשר מכשיר כרטוס ("terminal") או מחשב כלשהוא בא לשנות את הערך הכספי ששמור על הכרטיס, הוא קודם כל שולח פקודות Open Secure Session, לאחר מכן מבצע פקודת Update Record, ולבסוף שולח פקודת Close Secure Session.

בסוף ה-session, **שני הצדדים מייצרים MAC** (Message Authentication Code) באורך 8 בייטים מתוך כל התקשורת שעברה בין הכרטיס לקורא במהלך אותו session. ה-MAC מיוצר בהתבסס על מפתח סודי משותף באורך 16 בייטים ומוודא באופן הדדי.

### תהליך עדכון מידע על הכרטיס

ממבט מלמעלה, תהליך עדכון נתונים טיפוסי על רב קו נראה כך:

1. ה-teminal שולח לכרטיס פקודת SELECT APPLICATION כדי לבחור את תיקיית Calypso, ומקבל בחזרה את המספר הסריאלי של הכרטיס ומידע נוסף.
2. ה-terminal שולח לכרטיס פקודת Open Secure Session בצירוף challenge (מספר אקראי) וזיהוי של סוג המפתח שאיתו התקשורת תחתם (1, 2, או 3). הכרטיס מחזיר כתגובה לפקודה זו challenge משלו, את גרסת המפתח ששמור אצלו (Key Version and Category - KVC), ומספר פרטי מידע נוספים.
3. ה-terminal מבצע פעולות על הכרטיס, בהתאם לפעולה המבוקשת: קורא record-ים, מעדכן אותם ומשנה counter-ים (באמצעות פקודות UPDATE RECORD, READ RECORD, INCREASE או DECREASE). בכל שלב כזה, כל המידע האפליקטיבי שעובר בין הקורא לכרטיס (בשני הכיוונים) - מועבר לפונקצית hash שמעדכנת digest* באורך 8 בייטים.
4. בסוף ה-session, ה-digest **מוצפן** עם המפתח המבוקש על מנת לקבל את ה-MAC הסופי.
5. ה-terminal שולח לכרטיס פקודות Close Secure Session בצירוף ארבעת הבייטים העליונים של ה-MAC שהוא חישב. בתגובה, הכרטיס משיב עם ארבעת הבייטים התחתונים של ה-MAC שהוא חישב.
6. ולבסוף, אם החתימה שהכרטיס קיבל לא נכונה, כל המידע שנכתב יבוטל מייד. אם החתימה שה-terminal קיבל לא נכונה, הכניסה לא תאושר (לצורך העניין, יידלק אור אדום).
7. ה-terminal שולח לכרטיס פקודת GET CHALLENGE "מזויפת" ללא פרמטרים שמאשררת את קבלת התגובה ל-Close Secure Session.

*התוכן של פקודת ה-Open Secure Session נכלל ב-MAC גם כן, אבל ה-status bytes ספציפית של פקודה זו לא נכללים.

### חישוב ה-MAC ואלגוריתמי ההצפנה

על פי מפרטי קליספו, לשם ביצוע האימות וחישוב ה-MAC הכרטיס וה-terminal ישתמשו באחד מאלגוריתמי ההצפנה הבאים: DES (הישן), DES-X (Calypso 2+), Triple DES (Calypso 3.1+), או AES (Calypso 3.2+). מאחר שרוב הכרטיסים בארץ משתמשים ב-Calypso 2, אלגוריתם ההצפנה הוא **DES-X**, כלומר אלגוריתם חישוב ה-MAC יהיה:

1. מאתחלים את ה-digest עם נגזרת של המפתח
2. מעדכנים את ה-digest במידע התקשורתי שעבר ב-session, פקודה אחר פקודה
3. לאחר שה-hash הושלם, מצפינים אותו עם DES-X כך:
   א. מקסרים את ה-hash בחצי הראשון (8 בייטים ראשונים) של המפתח
   ב. מצפינים את התוצאה בעזרת DES עם **החצי השני** של המפתח
   ג. מקסרים שוב את התוצאה בחצי הראשון של המפתח

במקרה זה, על פי מסמכי Calypso, פונקציית ה-hash נקראת *Innovatron Hash*. לצערנו, את הפרטים המדויקים של פונקציה זו אי אפשר למצוא בדוקומנטציה פומבית. כאשר משתמשים ב-TDES או ב-AES, לעומת זאת, אלגוריתם חישוב ה-MAC יהיה CBC-MAC (Algorithm 1, Padding 2, כפי שמתואר בתקן ISO 9797-1).

### מפתחות סודיים

על מנת לחתום על התקשורת ולייצר את ה-MAC, הכרטיס וה-terminal משתמשים באחד מ-3 מפתחות סודיים, כאשר לכל אחד מהם יש מזהה משלו בהתאם לפעולה המבוקשת. בתהליך פתיחת ה-secure session, קורא הכרטיסים מציין באיזה סוג מפתח ברצונו להשתמש (Key Index), והכרטיס מחזיר לו את גרסת המפתח שמוגדרת אצלו (key version and category - KVC), כפי שנראה עוד בהמשך.

סוגי המפתחות האפשריים הם:

- Debit Key (Key Index=3) - כדי לחייב את הכרטיס (תיקוף)
- Load Key (Key Index=2) - כדי לטעון חוזים חדשים לכרטיס
- Issuer/Personalization Key (Key Index=1) - כדי לכתוב לכרטיס מידע גלובלי על בעליו וכו'

כל מפתח הוא באורך 16 בייטים ושמור בצורה מוגנת בזכרון הכרטיס. ככל שה-Index של מפתח כלשהוא קטן יותר, כך הוא בעל הרשאות גבוהות יותר בהכרח.

כל סט מפתחות ששמורים על כרטיס מסוים הם **ייחודיים לו** והם נגזרים משני דברים: המספר הסריאלי של אותו כרטיס ומפתח ראשי מתאים (master key). בצורה כזו, גם אם המפתחות הסודיים של כרטיס מסוים התגלו איכשהו, שאר הכרטיסים בעולם נשארים מוגנים (מה שנקרא מנגנון ה-diversification של המפתחות).

**ה-Transaction Counter**

כדי למנוע מתקפות Brute Force, הכרטיס סופר במונה בשם ה-Transaction Counter את מספר ה-Secure Sessions שהתקיימו בכל חייו (ליתר דיוק, את מספר הפעמים שהתבצע Open Secure Session או עוד מספר פקודות אחרות). המונה נשמר ב-EEPROM.

ה-Transaction Counter הוא מספר באורך 24 ביטים שבדרך כלל מתחיל מערך בסביבות 200,000 ויורד ב-1 בכל פעם שמבוצע Open Secure Session. כאשר ה-Transaction Counter מגיע ל-0, **הכרטיס ננעל** ולא ניתן יותר לבצע Secure Sessions או לשנות keys.

נשים לב שהפקודה Get Challenge (INS=0x84) מחזירה את ערך ה-Transaction Counter בשלושת הבייטים העליונים שחוזרים:

```
>>> ravkav_card.sr_apdu([0x94, 0x84, 0x00, 0x00, 0x8])

['GET CHALLENGE apdu: 94 84 00 00 08',
 'sw1, sw2: 90 00 - normal processing: command accepted: no further
qualification', (144, 0),
 [3, 10, 32, 192, 135, 134, 196, 205]]
```

בדוגמה זו ערך ה-Transaction Counter הוא 03 0A 20 או 199,200 בדצימלי.

בנוסף לכך מונה זה מהווה חלק מה-challenge שהכרטיס מחזיר כתגובה לפקודת ה-Open Secure Session. אציין שבצורה זו האנטרופיה של ה-challenge יורדת מ-4 בייטים כפי שהיה ניתן לחשוב לבייט אחד בלבד.

## המחשה #2 - טעינת רב-קו

בחלק זה אדגים כיצד ניתן לראות במציאות את תהליך הטעינה מחדש של כרטיס רב-קו בכסף, כפי שהסברתי מקודם.

כידוע, ניתן לטעון רב-קו בהינתן מחשב עם קורא כרטיסים או טלפון תומך NFC בעזרת תוכנות ייעודיות כגון רב-קו אונליין או HopOn. כיצד התוכנות האלו פועלות?

ובכן, רב-קו אונליין דורש התקנה של תוכנה ייעודית על המחשב (RavKavOnline.exe) שהיא בעצם תוכנת Go (או בעבר ג'אווה) שמדברת עם שרת WebSocket מרכזי.

השרת שולח לתוכנה זו (הקליינט) את פקודות ה-APDU שיש לשלוח לכרטיס, והקליינט בתורו מחזיר לשרת את התשובות שחזרו מהכרטיס. Design זה מבטיח כמובן שלא ניתן יהיה לחלץ שום מפתחות או אלגוריתמי חתימה מה-client, מכיוון שהלוגיקה המעניינת באמת נמצאת בצד שרת בלבד.

אם כך, הבה נבקש לקנות חוזה "ערך צבור" חדש ב-30 שקלים ונראה מה עובר בקו ה-USB בעזרת Wireshark.

### ההסנפה

כאשר נעבור על הפקודות שהוקלטו, במעבר ראשוני עליהן הכל יראה די סטנדרטי בשביל ISO 7816: בוחרים קבצים, קוראים קצת record-ים וכולי, עד שלפתע נראית פקודה לא סטנדרטית בעלת INS=0x8A - Open Secure Session:

```
Select file
Response APDU (to Select file)
Get response
Response APDU (to Get response)
Get response
Response APDU (to Get response)
Unknown instruction
Response APDU (to Unknown instruction)
```

[פקודת ה-Open Secure Session כפי שנראית בהסנפה]

כפי שאנחנו רואים, Wireshark לא הצליח לפרסר את הפורמט של פקודה זו מכיוון שהיא לא סטנדרטית והוא לא מכיר אותה.

התיעוד המדויק לפקודה Open Secure Session הוא לא פומבי לגמרי, אבל ב-GitHub של Calypso ניתן למצוא פרויקט אחד בשם kayple-java שיודע לשלוח ולפרסר את הפקודה Open Secure Session.

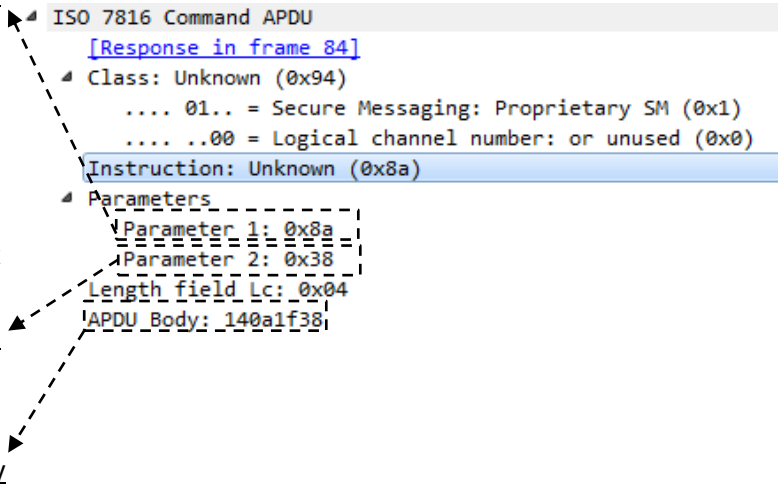
אז אם נסתכל בתוכן של הפקודה כפי שהגיעה בהסנפה נראה שאכן:

**P1**: כולל את ה-ID של המפתח שנרצה להשתמש בו (במקרה זה זהו מפתח הטעינה, כלומר Key Index=2) וגם את מספר ה-record שנרצה לקרוא, על פי הנוסחה:

0x80 + recordNumber * 8 + KeyIndex

**P2**: כולל את ה-SFI של הקובץ שנרצה לקרוא, ע"פ הנוסחה SFI * 8

Body: 4 בייטים של challenge




[תוכן פקודת ה-Open Secure Session שנשלחה לכרטיס בזמן טעינה]

התגובה לפקודה זו מצד הכרטיס היתה:

```
0000   60 03 09 c2 3a 20 00 06 ec 0a 04 a1 c5 a6 82 f5   `..Â: ..ì..¡Å¦.õ
0010   51 19 98 12 27 00 00 00 00 00 03 14 cd 45 80 00   Q...'.....ÍE..
0020   00 00 00 00                                       ....
```

**בסגול** - KVC (גרסת המפתח). **באדום** - ה-challenge שהכרטיס ייצר. **בירוק** - "Ratification Data", שנשלח רק במידה וה-session הקודם לא עבר ratification עדיין. **בכחול** - ה-record שביקשנו לקרוא (במקרה זה התוכן של Environment).

נשים לב לערך ה-challenge שהכרטיס החזיר. אם נבצע פקודות Open Secure Session נוספות שוב ושוב, נראה שה-LSB של ה-challenge נראה אקראי לחלוטין, בעוד שהחלק הנותר נראה כמו מונה שיורד - זהו ה-transaction counter.

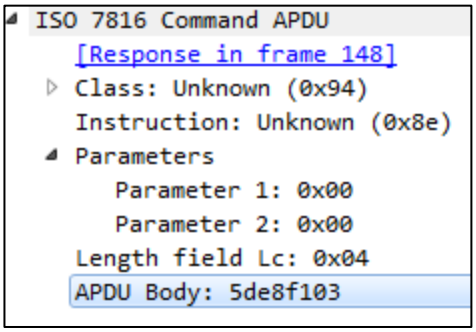
לאחר פתיחת ה-session וקריאת קובץ ה-Environment, אנחנו רואים שמבוצעות פקודות Read Record רבות שקוראות את הדברים הבאים, לפי הסדר:

1. ה-record הראשון בקובץ ה-Counters (=כמה כסף צבור בכרטיס)
2. ה-record הראשון בקובץ ה-Contracts List
3. כל ששת ה-record-ים בקובץ ה-Events
4. כל שמונת ה-record-ים בקובץ ה-Contracts
5. כל ארבעת ה-record-ים בקובץ ה-Special Events

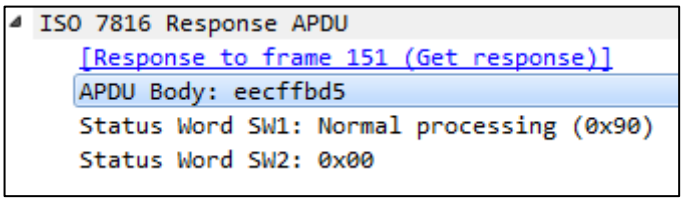
ואז:

1. מבוצע Update Record ל-record הראשון בקובץ ה-Contracts, כדי להחליף את החוזה הישן בחוזה החדש שנרכש
2. מבוצע Append Record(s) לקובץ ה-Events, כדי לתעד את פעולת הטעינה
3. מבוצע Increase (שוב מופיע כ-Unknown Instruction) ל-record הראשון בקובץ ה-Counters כדי להגדיל את כמות הכסף הצבור. ה-data בפקודה זו הוא 3750 בדצימלי - שזה סכום הכסף שהחלטתי לטעון.

לבסוף, ניתן לראות בהסנפה עוד Unknown Instruction עם INS=0x8E - הלוא היא פקודת Close Secure Session שכוללת את החלפת החתימות (MAC) הגורלית:




[התשובה לפקודת Close Secure Session מצד הכרטיס בצירוף החתימה שלו (body)]




[פקודת Close Secure Session מצד הקורא בצירוף החתימה שלו (body)]

## עוד על אבטחת הטרנזקציות

האבטחה בקליפסו לא מסתכמת ב-Secure Session וחישוב MAC-ים, אלא ממשיכה גם לצד של נותן השירות.

### רכיבי אבטחה מרכזיים - SAM-ים

המפתחות הראשיים השונים (master keys) שמסוגלים לשנות את המידע בכרטיס שמורים ברכיב אבטחה ייעודי - Security Application Module (SAM), שיכול לשבת ב-terminal או במכרז שירות מרוחק.

SAM-ים הם למעשה כרטיסים חכמים בעצמם ויש להם סט שלם של פקודות משלהם:

- SELECT DIVERSIFER (0x14) - על מנת להכין את ה-Key המתאים לכרטיס מסוים ע"פ המספר הסריאלי שלו
- GET CHALLENGE (0x84) - על מנת לחשב את ה-challenge שיינתן לכרטיס
- DIGEST INIT (0x84) - על מנת לאתחל את חישוב ה-hash עם נגזרת של המפתח
- DIGEST UPDATE (0x8C) - על מנת לעדכן את ה-hash בזוג פקטות נוסף
- DIGEST CLOSE (0x8E) - על מנת לסיים את חישוב החתימה
- DIGEST AUTHENICATE (0x82) - על מנת לבדוק האם חתימת הכרטיס נכונה

**סוגי SAM-ים**

כפיצ'ר אבטחה נוסף, ישנם כמה סוגים של SAM-ים, כאשר כל אחד מהם משמש למטרה ספציפית וכולל מפתחות שמאפשרים לו לעשות את הפעולה הספציפית הזאת בלבד.

סוגי ה-SAM-ים שנעשה בהם שימוש בישראל הם:

- **SAM-CL (Card Load)**: כולל מפתחות המאפשרים טעינת חוזים חדשים ותיקוף (Load Key). נמצאים במכונות תיקוף באוטובוסים, ברכבות וכו'.
- **SAM-CV (Card Validation)**: כולל מפתחות המאפשרים תיקוף בלבד (Debit Key).
- **SAM-CP (Card Personalization)**: כולל מפתחות המאפשרים פרסונליזציה, כלומר כתיבה לקובץ ה-Environment (Issuer Key).

נמצאים בעמדות הנפקה של המפעילים ברחבי הארץ (עמדות "על הקו").

- **SAM-CPP-K/SAM-CPP (Card Pre-Personalization):** מאפשרים אתחול כרטיסים (Pre-Personalization). תהליך זה כולל הטבעת מפתחות בכרטיס ויצירת הקבצים, למשל.

אתחול כרטיסי רב קו מתבצע ע"י יצרן חיצוני שרכיבי ה-SAM הנ"ל מושאלים לו, או בארץ ע"י מרכז השירות.

כמו כן ישנם שני סוגי SAM-ים בשימוש שמשמשים כדי לשלוט ב-SAM-ים אחרים:

- **SAM-SL (SAM Load)**: משמש כדי לשלוט במספר המקסימלי של טעינות ש-SAM מסוג SAM-CL יכול לבצע.
- **SAM-SP (SAM Personalization)** - זהו ה-SAM היחיד שיכול לאתחל SAM-ים אחרים - ולעדכן מפתחות. לכן הוא נקרא ה-"Master SAM". רכיב זה נמצא תחת האחריות של מרכז השירות והוא נשמר במקום קבוע, מלבד הפעמים שבהם הוא מאתחל רכיבי SAM אחרים.

בישראל, שלושת סוגי ה-SAM-ים האחרונים שהוזכרו (SAM-SL, SAM-SP, SAM-CPP/SAM-CPP-K) מאובטחים היטב: הם נשמרים בכספות כאשר הם לא בשימוש, ויש להם גיבוי בקריית הממשלה בירושלים. בנוסף, מנוהל יומן מעקב אחר תנועות של כל הרכיבים ומתבצעות ספירות מלאי מעת לעת.

### Ratification (אשרור) וניתוק חיבור פתאומי

ב-Calypso מתגאים בפיצ'ר אבטחתי נוסף בשם ratification. מנגנון זה דואג לכך שבמידה וקשר הרדיו (או המגע) בין הכרטיס לבין הקורא מתנתק באופן פתאומי במהלך secure session, לעולם לא ייווצר מצב שבו רק חלק מהמידע שאמור היה להיכתב באמת נכתב. כלומר, או שכל הפעולות שאמורות היו להתבצע אכן מתבצעות (כולל אימות החתימות) - או שאף אחד מהפעולות לא מתבצעת.

פיצ'ר זה בא לפתור את המקרה שבו הקישור מתנתק **לאחר** שנשלחת פקודת Close Session עם חתימת הקורא, אבל **לפני** שהכרטיס עונה לפקודה זו עם החתימה שלו. בשלב זה הכרטיס כבר חויב (כי הוכח שהקורא לגיטימי), אבל הקורא לא יכול לאמת את לגיטימיות הכרטיס.

ההשלכות מבחינת אבטחה: אי אפשר להתחיל secure session, לעדכן כסף ולברוח. אם וכאשר הקשר בין הקורא לכרטיס יתנתק מבלי שה-secure session נסגר, כל הפעולות שכתבו מידע יבוטלו מיידית.

### עוד על ה-Secure Session

רעיונות ה-Secure Session וה-Ratification של קליפסו מוגנים באמצעות פטנטים שמתארים את אופן הפעולה שלהם: כל עוד הכרטיס במצב Session, מידע שנכתב באמצעות אחת מפקודות הכתיבה יתועד בצד, מבלי לאבד את הערכים של המידע המקורי.

כאשר אותו מידע ייקרא, יוחזרו הערכים שנשמרו בצד, ולא הערכים האמיתיים. המידע ייכתב לתאים המתאימים ב-EEPROM רק ברגע שה-session ייסגר.

על כן, הכרטיס שומר modifications buffer בגודל מסוים (בדרך כלל 215 בייטים). כאשר ה-buffer הזה מתמלא בזמן ה-session, לא ניתן יותר לבצע פקודות שכותבות מידע וה-session מתבטל.

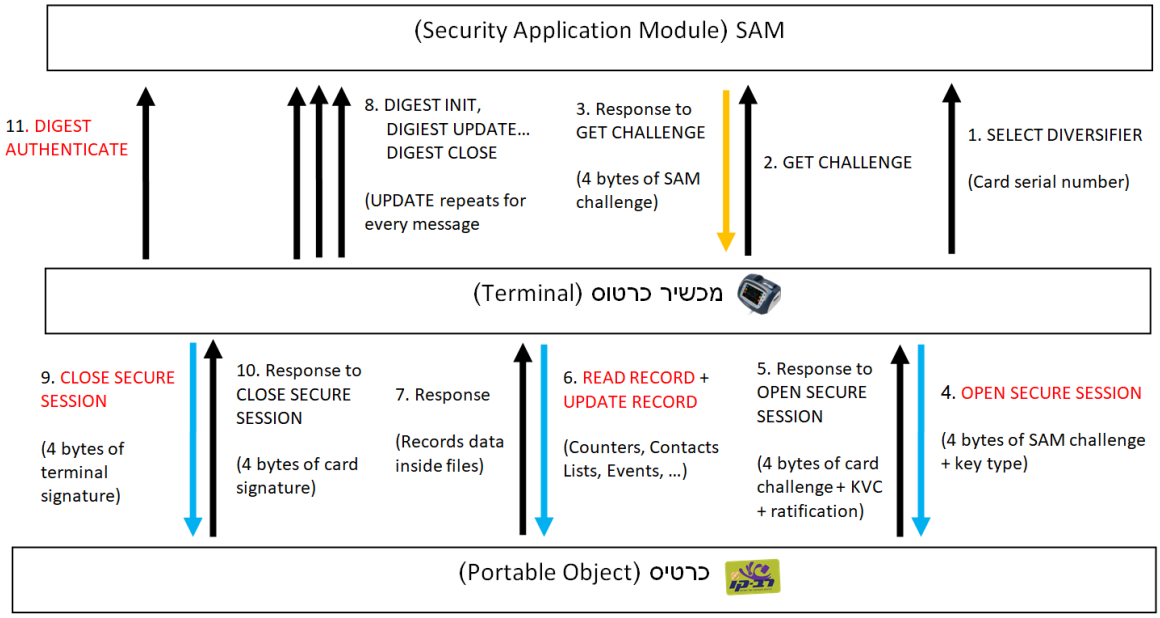
## ההשלכות

כפי שראינו למעלה, מודל האבטחה הזה מוכיח לפחות תיאורטית כמה דברים בו-זמנית:

1. את האמינות של הכרטיס ל-terminal (כדי שלא ניתן יהיה לטעון שיש לך יותר כסף ממה שיש לך באמת)
2. את האמינות של ה-terminal לכרטיס (כדי שלא ניתן יהיה לטעון את הכרטיס בכסף מבלי לשלם לנותן השירות האמיתי)
3. את אמינות המידע שמועבר בניהם (כדי שלא ניתן יהיה לבצע MITM ולערוך את המידע באמצע או לבצע replay attack)

### סיכום ה-flow המלא

ניתן לראות את Flow החיוב במלואו בקוד Demo_ValidationTransaction.java או בתרשים המסכם הבא:




## פקודות נוספות

מלבד פקודות שמוגדרות ב-ISO 7816-4 ופקודות אחרות שמוגדרות במפרטי קליפסו, התוכנה שצרובה על ה-ROM בכרטיס יכולה כמובן לממש פקודות ייעודיות אחרות.

חלק מהפקודות הייעודיות מממשות פקודות שמוגדרות בתקן EN 726-3. להלן טבלה מסכמת על סמך ניסויים שביצעתי:

| סוג כרטיס \ פקודה | ASK CD21 (ST19, 2010) | ASK FST P (2019) | ASK TanGo (2010) | Watchdata (SLE77/SLE66, 2016) | OTI 10054127-1 (CD21, 2010) | IDEMIA cityGo (ST23, 2019) |
|---|---|---|---|---|---|---|
| **0x4 (DEACTIVATE FILE)** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** |
| 0xe (ERASE BINARY) | | | | נתמך | | |
| 0x10 (FABRICATION READ) | נתמך | | | | נתמך | נתמך |
| 0x1c | נתמך | | | | נתמך | נתמך |
| 0x1e | נתמך | | | | נתמך | נתמך |
| **0x20 (VERIFY PIN)** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** |
| **0x30 (DECREASE)** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** |
| **0x32 (INCREASE)** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** |
| **0x38 (DECREASE MULTIPLE)** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** |
| **0x3a (INCREASE MULTIPLE)** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** |
| 0x44 (ACTIVATE FILE / REHABILITATE) | נתמך | נתמך | נתמך | נתמך | נתמך | נתמך |
| 0x52 (CHANGE SPEED) | נתמך | | | | נתמך | |
| 0x53 | נתמך | | | | נתמך | |
| 0x76 | | נתמך | נתמך | | | |

| | | | | | | |
|---|---|---|---|---|---|---|
| 0x78 | | נתמך | נתמך | | | |
| 0x7c (SV GET) | נתמך | | | נתמך | | |
| 0x82 (EXTERNAL AUTHENTICATE / MANAGE SECURE SESSION) | | | | נתמך | | נתמך |
| **0x84 (GET CHALLENGE)** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** |
| 0x86 (GENERAL AUTHENTICATE / GIVE RANDOM) | נתמך | נתמך | נתמך | | נתמך | נתמך |
| **0x8a (OPEN SECURE SESSION)** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** |
| 0x8c (ABORT SECURE SESSION) | נתמך | נתמך | נתמך | נתמך | נתמך | נתמך |
| **0x8e (CLOSE SECURE SESSION)** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** |
| 0xa2 (SEARCH RECORD) | | נתמך | נתמך | | נתמך | נתמך |
| **0xa4 (SELECT FILE / SELECT APPLICATION)** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** |
| 0xb0 (READ BINARY) | נתמך | נתמך | נתמך | נתמך | נתמך | נתמך |
| **0xb2 (READ RECORD(S))** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** |
| **0xb3 (READ RECORDS MULTIPLE)** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** |
| 0xb4 | | נתמך | נתמך | | | |
| 0xb6 (READ RECORD STAMPED) | נתמך | נתמך | נתמך | | נתמך | נתמך |
| 0xb8 (SV RELOAD) | נתמך | | | נתמך | נתמך | |
| 0xba (SV DEBIT) | נתמך | | | נתמך | נתמך | |
| 0xbc (SV UNDEBIT) | נתמך | | | נתמך | נתמך | |
| 0xbe (GET ATR) | נתמך | נתמך | נתמך | נתמך | נתמך | נתמך |
| 0xbf | | נתמך | נתמך | | | |
| **0xc0 (GET RESPONSE)** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** |
| 0xc4 | | נתמך | נתמך | | | |
| 0xca (RETRIEVE DATA) | | נתמך | נתמך | | | נתמך |
| **0xd0 (WRITE BINARY)** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** |
| **0xd2 (WRITE RECORD)** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** |
| 0xd4 | | | | נתמך | | |
| **0xd6 (UPDATE BINARY)** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** |
| **0xd8 (CHANGE KEY / CHANGE PIN)** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** |
| 0xda (SET DATA) | | | נתמך | | | נתמך |
| **0xdc (UPDATE RECORD)** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** |
| 0xe0 (CREATE FILE) | נתמך | | | נתמך | נתמך | נתמך |
| 0xe1 (GET AVAILABLE MEMORY) | נתמך | | | | נתמך | נתמך |
| **0xe2 (APPEND RECORD)** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** | **נתמך** |
| 0xf2 (STATUS) | נתמך | | נתמך | | נתמך | נתמך |

השורות המודגשות מייצגות את הפקודות שמוגדרות במפרטי קליפסו (ניתן לראות שאכן כל הכרטיסים תומכים בהן). כל שאר הפקודות, כפי שאפשר לראות, לא תמיד נתמכות (כמו פקודות Stored Value), שלא לדבר על פקודות proprietary לא מתועדות בעליל.

## מכונות כרטוס

הנוסע הממוצע בוודאי נתקל בחייו בכל מני סוגים של מכונות בעלות יכולת קריאת רב-קו אלחוטית (בעזרת NFC). גם כאן, ישנם הרבה סוגים של מכונות: כספומטים של רב-קו, עמדות חיוב או הטענה באוטובוסים, שערים ברכבות ובמרכזי שירות.

לרוב מדובר על אחד משני סוגי מכונות:

- Ticketing Vending Machines (TVM) - "כספומטים" של כרטיסים שמסוגלים לטעון כרטיסי רב-קו בחוזים חדשים או להציג את המידע ששמור עליו
- Validators / Driver's Consoles - מכונות קטנות ליד הנהג או בכל מקום אחר שמאפשרות תיקוף (validation) של הכרטיס בעת נסיעה.

סוגי המכונות המדויקים משתנים ממפעיל למפעיל. במקרה של אגד, למשל, ניתן להיתקל פעמים רבות במכונות של חברת TransWay הישראלית.

*[Roadzter TIM 7020 (מימין) ו-Voyageur Onboard 6000 (משמאל), מתוך האתר של TransWay]*

**המפרט הטכני:** אנדרואיד 4 כמערכת הפעלה, GPS פנימי, מודם סלולארי, תמיכה ב-Calypso, ב-Mifare, ועוד. נתונים על השימוש בכרטיסים החכמים נשלחים למרכז השירות על גבי המודים הסלולארי, בזמנים קבועים, וגם נשמרים על מודול ייעודי של המכונה (DCU) לצורכי גיבוי.

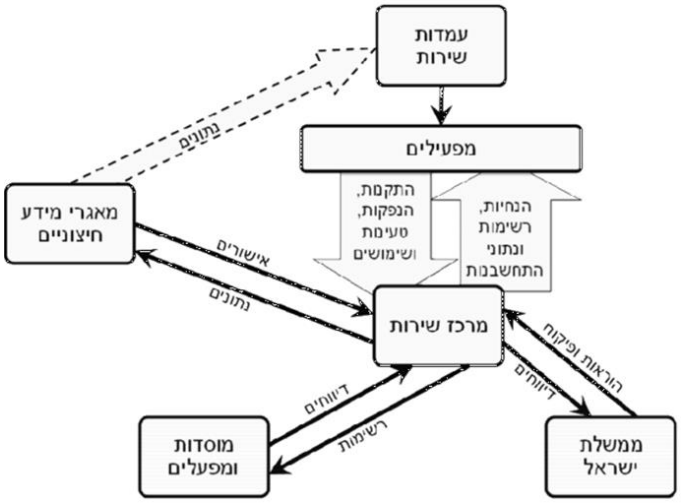
אצל מפעילים אחרים (למשל מטרופולין, קווים או דן) ניתן לראות שימוש במכונות תיקוף וקונסולות של חברת Conduent (לשעבר Affiliated Computer Services (ACS)/Xerox), כגון ה-VPE 415 או ה-PCE 415. המכונה הנ"ל (ה-Proxibus PCE-415), לעומת הקודמת, בעלת מסך מגע ומריצה Windows CE על מעבד ARM.

חברה ישראלית אחרת שהייתה הראשונה לספק קונסולות נהג התומכות בתקן קליפסו הישראלי היא חברת Symcotech, שאחראית גם על חלק משערי התיקוף ברכבת ישראל.

לבסוף, חברה נוספת שיש לה שימוש בישראל היא חברת ACS (Advanced Card Systems) הבינלאומית, שמייצרת, מלבד כרטיסים חכמים ומערכות הפעלה לכרטיסים חכמים, גם קוראים ומכונות כרטיסים בעיצוב מותאם אישית.

## Back-Office ומכרז השירות

על פי החוק להסדרת השימוש בכרטיס חכם בישראל, מידע על הנוסע (שם, תאריך הנפקה וכו') וכן המידע על הנסיעות האחרונות (מספר קו, מספר רכב, זמן) לא נשמר רק על שבב הכרטיס אלא גם מגובה במאגרי מידע ייעודיים של המפעילות, שמדווחים לאחר מכן למרכז השירות של הרשות הארצית לתחבורה ציבורית.




*[תרשים העברת נתונים במערכות Back-Office, מתוך "אוגדן נהלי כרטוס"]*

מה גם שעל פי החוק מאגרים אלו חייבים להיות רשומים בפנקס מאגרי המידע בישראל.

המידע על הטרנזקציות נשלח למאגרי המפעילים לפחות ברמה יום-יומית, אם לא מספר פעמים ביום. משם, לאחר מכן, בתדירות קבועה, המידע נשלח למרכז השירות של התחבורה הציבורית בישראל. כידוע, כל לקוח רשאי לבקש שחזור של תוכן הכרטיס ממאגרי המידע האלו במקרה של אובדן - והמידע ישוחזר לאחר מקסימום 72 שעות.

### בין המפעיל למשרד התחבורה

נתוני הטרנזקציות של רב-קו הם רק חלק מהתנועות שמועברות בסופו של דבר למשרד התחבורה. למעשה, כל מפעיל תחבורה ציבורית בישראל מחויב להעביר למרכז השירות נתונים מכמה סוגים באופן קבוע. כל סוג נתונים מועבר בצורה של **קבצים** קטנים עם שם ומבנה מוגדר היטב. סוגי הנתונים האפשריים הם:

- **נתוני תנועות** - נתונים על השימוש השוטף של נוסעים ברב קו שלהם. אציין שבכל תנועה כזו מועברים הנתונים הבאים:
  - סוג התנועה (תיקוף רגיל, טעינה ושימוש, ביטול טעינה וכו')
  - מספר הכרטיס
  - סכום הכסף שחיוב

- מזהה נסיעה, מספר הקו, הכיוון, זמן תחילת הנסיעה
- סוג מכשיר הכרטוס, מספר המכשיר
- ערך מונה ה-SAM-CL באותו רגע
- התוכן הרלוונטי של קבצי ה-Environment וה-Contracts (החוזה המתאים) כפי שהופיעו בכרטיס
- התוכן של שלושת ה-Events האחרונים, כפי שהופיעו הכרטיס

- **נתוני פרסונליזציה**: נתונים שהמפעיל מעביר למכרז השירות על הנפקות כרטיס (אישי, אנונימי, ...)
- **נתונים על SAM-ים:** נתונים שהמפעיל מעביר למכרז השירות על התקנות SAM-ים חדשים ואתחול שלהם, דיווח על תקלות או גניבות SAM-ים וכולי
- **דיווח לרשימה השחורה:** נתונים שהמפעיל מעביר למכרז השירות כאשר יש כרטיס שמועמד להיחסם, לדוגמה בעקבות אובדן, השחתה או זיוף שזוהה
- **הפצת רשימה שחורה:** נתונים שמרכז השירות מפיץ למפעילים מדי פעם, שכוללים את מספרי הכרטיסים שנחסמו ואת רמת הסיכון של כל אחד מהם

פרטים נוספים על מערכות ה-Back Office הם מחוץ ל-scope של המאמר.

## כיווני מחקר להמשך

לאחר הבנת המנגנונים שבעזרתם הכרטיס פועל, עדיין עולות שאלות רבות אותן לא סקרתי, כגון:

- עד כמה מאובטח ה-hash ה-proprietary של קליפסו והאם אפשר ליצור בו collisions?
- האם אפשר לנצל את האנטרופיה היחסית נמוכה של ה-challenge כדי לייצר replay attack?
- מה עושות הפקודות הלא מתועדות בעליל?
- האם אפשר לחלץ את התוכנה משבב ה-ROM בעזרת מתקפות חומרתיות (מיקרוסקופ אלקטרונים, Power Analysis וכו')?
- האם אפשר לנצל את העובדה ששרתי הענן של רב קו מאפשרים ליצור MAC תקין על מידע בעזרת כל סריאל שנרצה?
- האם יש דרך לקרוא מידע מ-Internal EFs, ובינהם קובץ המפתח?
- ועוד

## סיכום

במהלך המאמר קיבלנו טעימה מעולם הכרטיסים החכמים, ראינו איך אפשר להגן על המידע ששמור עליהם, ובנוסף ראינו בכלליות איך התקן של הכרטיס ממומש בארץ ואילו חברות מעורבות. למרות שיש עוד פוטנציאל עומק רב בטכנולוגיות אלה ופרטים נעלמים, כולי תקווה שהמאמר סיפק את סקרנותכם לפחות כמו שהוא סיפק את שלי.

ומה לגבי העתיד של הכרטיסים האלה? בכמה מדינות קליפסו כבר מפעילה את התקן שלה על Java Cards במקום על כרטיסים מסורתיים, וספציפית גם כאפליקציית Java שמותקנת ישירות על ה-SIM במכשיר הסמארטפון (בשיתוף פעולה עם מפעילי סלולר ויצרניות), שיכול לעבוד אפילו כשהמכשיר מכובה.

בנוסף לכך, לאחרונה גם יש שימוש גובר והולך באמלוץ כרטיסים (Host Card Emulation, HCE) על גבי הסמארטפון ובקונספט של (Account Based Ticketing (ABT, בו הכרטיס הוא פשוט מזהה מאובטח לחשבון, וכל הפרטים שמורים במערכות ה-Back Office.

## מקורות / לקריאה נוספת

**Calypso**

- מידע כללי על קליספו:

https://en.wikipedia.org/wiki/Calypso_(electronic_ticketing_system)

https://www.calypsonet-asso.org/what-is-calypso-mobile

https://www.calypsonet-asso.org/content/ticketing-transaction-layers

https://www.eclipse.org/lists/iot-wg/pdfjHD1WVBg2J.pdf

https://web.archive.org/web/20170722120235/http://contactless-world.com/category/calypso/

- Calypso Handbook - חוברת בעלת מידע על הטכנולוגיה והארגון:

https://www.calypsonet-asso.org/content/handbook-document

- Calypso Functional Specification - המסמך הטכני הפומבי המרכזי:

https://www.innovatron.fr/CalypsoFuncSpecification.pdf

- שאר המסמכים הטכנולוגיים הפומביים של קליפסו:

http://www.calypsostandard.net/public-documents

- פירוט על סוגי כרטיסי קליפסו:

https://www.calypsonet-asso.org/content/products

https://www.calypsonet-asso.org/content/calypso-certified-products

http://www.calypsostandard.net/po-certification/certified-products

https://www.calypsonet-asso.org/providers

https://www.innovatron.fr/CalypsoProducts-Cards.pdf

http://www.ask-contactless.com/Portals/0/Flyers/Cards/CalypsoCard_flyer.pdf

- מידע על SAM באתר Spirtech:

https://spirtech.com/software-house-security-sam

https://spirtech.com/docs/Products/SAM-S1/en/SamS1E-v6.pdf

- הסברים כלליים על האבטחה של קליפסו, באתר של CNA:

https://www.calypsonet-asso.org/node/28

https://www.calypsonet-asso.org/content/secure-transaction

https://www.calypsonet-asso.org/secure

- מצגת על Calypso בתעשייה:

http://ceesca.org/uploads/presentations/02-2013/Calypso%20-%20CEESCA%20Kranjska%20Gora%20022013.pdf

- הדגמה של כרטיסי קליפסו בפעולה (באתר הטכני):

http://demo.calypsostandard.net/

- רשימת Validator-ים שעברו אימות ע"י קליפסו:

https://calypsostandard.net/registration-process/registered-products

- הסברים על שילוב Calypso בישראל:

https://www.calypsonet-asso.org/news/interoperability-ticketing-israel

http://cna.adcet.com/index.php/8-archives/49-

**<u>תקנים</u>**

- ISO/IEC 7816 -התקן לכרטיסים חכמים contact, חלק רביעי: פרוטוקול תקשורת אפליקטיבי וארגון מידע:

http://www.embedx.com/pdfs/ISO_STD_7816/info_isoiec7816-4%7Bed21.0%7Den.pdf

- Reference מקוצר ל--ISO 7816-4:

http://cardwerk.com/smart-card-standard-iso7816-4-section-5-basic-organizations

http://cardwerk.com/smart-card-standard-iso7816-4-section-6-basic-interindustry-commands/

https://en.wikipedia.org/wiki/Smart_card_application_protocol_data_unit

- ISO/IEC 7816 - התקן לכרטיסים חכמים contact, חלק שלישי:מגעים ותקשורת אלקטרונית:

http://read.pudn.com/downloads132/doc/comm/563504/ISO-IEC%207816/ISO%2BIEC%207816-3-2006.pdf

- רשימת status words (SW) אפשריים והמשמעות שלהם:

https://www.eftlab.com/knowledge-base/complete-list-of-apdu-responses/

- ISO/IEC 1443 - התקן לכרטיסים חכמים contactless:

http://www.emutag.com/iso/14443-2.pdf

http://www.emutag.com/iso/14443-4.pdf

- EN 1545 (טיפוסי נתונים):

https://www.evs.ee/preview/evs-en-1545-1-2015-en.pdf

https://www.evs.ee/preview/evs-en-1545-2-2015-en.pdf

- Intercode (NF P 99-405):

http://billettique.fr/spip.php?rubrique20

- CCID (פרוטוקול USB לתקשורת עם כרטיסים חכמים):

https://en.wikipedia.org/wiki/CCID_(protocol)

- NFC:

https://en.wikipedia.org/wiki/Near-field_communication

- PS/SC:

https://en.wikipedia.org/wiki/PC/SC

- ATR:

https://en.wikipedia.org/wiki/Answer_to_reset

http://www.icedev.se/proxmark3/docs/ISO-7816-3.pdf

## כלים ותוכנות

- Cardpeek - תוכנה להצגת מידע ששמור על כרטיסים חכמים מסוגים שונים (תומך רב קו):

http://pannetrat.com/Cardpeek/

- ספריות פייתון לתקשורת עם כרטיסים חכמים באמצעות מחשב:

https://github.com/LudovicRousseau/pyscard

https://github.com/mitshell/card

- Calypso Explorer - תוכנה לקריאת מידע מכרטיסי קליפסו:

https://tech.springcard.com/2010/calypso-explorer/

- APDUScanner, כלי שסורק פקודות ומערכות קבצים של כרטיסים חכמים:

http://apduscanner.sourceforge.net/

- Smart Card ToolSet, תוכנה לחקירת כרטיסים חכמים:

http://www.scardsoft.com/index.php?Theme=Soft_v3Server

- Calypso Inspector, פרויקט נוסף שקורא כרטיסי קליפסו:

https://github.com/ABeaujet/CalypsoInspector

- קוד שיודע לפרסר את ה-ATR של כרטיסי קליספו (חלקית):

https://github.com/elafargue/smart-tools/blob/master/atr/lib/calypso-xplore.html

- אתרים שיודעים להבין ATR-ים של כרטיסים:

https://smartcard-atr.appspot.com/

http://www.ruimtools.com/atr.php

- קוד ג'אווה מטעם Calypso שיודע לבנות ולפרסר פקודות קליפסו:

https://github.com/calypsonet/keyple-java/tree/develop/java/component/keyple-calypso/src/main/java/org/eclipse/keyple/calypso/command/po/builder

- תיעוד אנדרואיד למחלקה IsoDep:

https://developer.android.com/reference/android/nfc/tech/IsoDep.html

- מדריך להסנפת ISO 7816-4 על גבי USB CCID באמצעות Wireshark:

https://ludovicrousseau.blogspot.com/2019/08/iso-7816-4-spy-using-wireshark.html

## קריאת מבנה הנתונים על רב קו

- קוד פרסור רב קו שנכתב בשפת Go:

https://github.com/ybaruchel/ravkav-sdk-go/tree/master/card/normalizers

- קוד ה-LUA של cardpeek שמפרסר את פורמט הנתונים של רב קו (בחלקו):

https://github.com/L1L1/cardpeek/blob/master/dot_cardpeek_dir/scripts/calypso/c376n3.lua

https://github.com/L1L1/cardpeek/blob/master/dot_cardpeek_dir/scripts/etc/ravkav-strings.lua

- קוד שמפרסר רב-קו וקליפסו, מתוך פרויקט בשם ISRDriverConsole:

https://github.com/NadavPoliti/ISRDriverConsole/tree/master/app/src/main/java/com/isr/isrdriverconsole/M/SmartCard/CalypsoCard

- קוד נוסף שמנסה לפרסר רב-קו (תחת פרויקט בשם metrodroid):

https://github.com/metrodroid/metrodroid/pull/70/files

## כרטיסים אחרים מבוססי Calypso

- פוסט טכנולוגי על כרטיסי תחברה חכמים ועל כרטיס ה-Navigo הצרפתי בפרט:

http://www.piratesmag.com/xxx/pass.html

- מאמר שחוקר את כרטיס ה-Libon VIVA הפורטוגלי:

https://web.archive.org/web/20181027152739/http://downloads.pannetrat.com/get/f2fed7a73d962024e94f/viva-report.pdf

- הסברים על הטכנולוגיה והתיקוף של כרטיס OPUS הקנדי:

https://www.revolvy.com/page/OPUS-card?cr=1

- הסברים טכנולוגיים על המידע שנאסף מכרטיס IMOB באיטליה:

http://tramaci.org/anoptiblog/2009/9/index.py

## חברות

- רשימה של חברות המעורבות בתעשיית Calypso, כולל חלוקת אחריות:

https://www.innovatron.fr/licensees.html

http://cna.adcet.com/index.php/products/suppliers

**<u>חברות כרטיסים</u>**

- (ASK TanGo) ASK / Paragon ID:

http://ask-contactless.com/en-us/applications/transport.aspx

https://paragon-id.com/en/solutions/dual-interface-smart-cards

http://ask-contactless.com/NewsEvents/News/tabid/126/ctl/ArticleView/mid/559/articleId/66/language/en-US/ASK-launches-TanGO-at-Cartes-2004-a-universal-contactless-platform.aspx

- (TimeCOS) Watchdata:

https://www.watchdata.com/timecos-transport-cardwatchdata/

http://www.watchdata.com/class.php?id=29

http://read.pudn.com/downloads197/doc/927684/WatchData_TimeCOS2.9.pdf

- Gemalto:

https://www.gemalto.com/transport/contactless-cards

https://www.gemalto.com/brochures-site/download-site/Documents/transport-calypso.pdf

- OTI:

https://en.globes.co.il/en/article-1000099726

https://www.securetechalliance.org/oti-awarded-mass-transit-ticketing-project-to-supply-cards-for-israels-mass-transit-system/

https://www.otiglobal.com/contactless_ticketing/

- IDEMIA (לשעבר Morpho, Oberthur):

https://web.archive.org/web/20170713041039/https://www.morpho.com/sites/morpho/files/morpho_payment_calypso_cl_4p_gb.pdf

**<u>חברות צ'יפים</u>**

- STMicroelectronics:

http://pdf.dzsc.com/88889/5310.pdf

https://www.st.com/resource/en/data_brief/st19wr02.pdf

https://www.ssi.gouv.fr/uploads/IMG/certificat/dcssi-cible_2006-02en.pdf

https://www.st.com/content/ccc/resource/sales_and_marketing/promotional_material/flyer/84/49/47/21/74/15/43/69/flcalypso1010.pdf/files/flcalypso1010.pdf/jcr:content/translations/en.flcalypso1010.pdf

https://www.st.com/en/secure-mcus/cd21-rev3.html

https://www.st.com/resource/en/data_brief/cd21-rev3.pdf

- (SLE 77, SLE 66) Infineon:

https://www.mouser.com/datasheet/2/196/Infineon-Chip_Card_Security_ICs_Portfolio_2017-SG--776471.pdf

http://static6.arrow.com/aropdfconversion/48639423124ad32532b264761f907d9521461086/spi_sle66clx800pe_1209.pdffolderiddb3a304325305e6d012572d28cdf626dfileiddb3a304325305e6d01259c9.pdf

https://www.infineon.com/cms/en/product/security-smart-card-solutions/security-controllers/sle-77/

**חברות Validator-ים / Terminal-ים**

- (TIM 7020) Transway:

http://www.transway.co.il/driver-operated-ticket-sales-ticket-machine

http://www.transway.co.il/ticket-management-system-success-stories/public-transport-ticketing-system-israel/bus-ticketing-systems-egged

http://www.transway.co.il/driver-operated-ticket-sales-ticket-machine/roadzter-tim-7020-ticketing-machine

- Symcotech:

https://web.archive.org/web/20180831162824/http://symcotech.co.il/apage/11916.php

https://web.archive.org/web/20180831215141/http://symcotech.co.il/apage/16470.php

- (Affiliated Computer Services) ACS / Conduent:

https://en.wikipedia.org/wiki/Affiliated_Computer_Services

https://www.xerox.com/downloads/europe/events/transport-publics/PCE415-EN.pdf

http://dump.webgui.ws/imob/vpe_415_en.pdf

- (Advanced Card Systems) ACS:

https://www.acs.com.hk/en/product-lines/1/smart-cards-smart-card-operating-systems/

**שונות**

- Smart Card Handbook - כולל מידע מפורט על תהליך הייצור של כרטיסים ועל הצ'יפים שלהם:

http://index-of.co.uk/Etc/Smart%20Card%20Handbook.pdf

- מצגות/הרצאות על הנדסה הפוכה של צ'יפים "חכמים", מאת Christopher Tarnovsovsky:

https://www.blackhat.com/presentations/bh-dc-08/Tarnovsky/Presentation/bh-dc-08-tarnovsky.pdf

https://www.youtube.com/watch?v=62DGIUpscnY

- מצגות על כרטיסים חכמים והתקפות כנגדיהם:

http://opensecuritytraining.info/SmartCards_files/SmartCards.pdf

- מסמך שמשווה בין טכנולוגיות כרטוס שונות. עמוד 14 כולל אלגוריתמים קריפטוגרפים:

http://www.smart-ticketing.org/downloads/ifm-project/D3.3_201002.pdf

## מקורות בעברית

- מידע כללי מאוד על רב-קו:

https://www.gov.il/he/departments/guides/multi_line_card

- "אוגדן נהלי כרטוס" - אוסף מסמכים מפורטים המתארים את אופן השימוש של כרטיסי רב קו בישראל על פי תקן קליפסו, באתר משרד התחבורה:

https://www.gov.il/he/departments/general/smart_ticketing

- מסמך מפורט על מערכת הכרטוס במטרונית (באתר הממשלה):

https://www.gov.il/BlobFolder/guide/agreements_for_operating_service_lines_in_public_transportation/he/matronic_tender_attache_d.pdf

- מכרז המתעסק בעמדות ההנפקה של רב קו, בשמירת פרטי הנוסעים במרכזי שירות ועוד. באתר של מינהל הרכש הממשלתי:

https://www.mr.gov.il/Files_Michrazim/297123.pdf

- החוק להסדרת השימוש בכרטוס החכם בתחבורה הציבורית (2013):

https://www.nevo.co.il/law_word/Law11/41680.doc

- הנחיות על מאגרי מידע של מפעילי כרטוס חכם (הרשות להגנת הפרטיות):

https://www.gov.il/he/departments/policies/public_transport

- הערות על המידע האישי שנאסף ונאגר מכרטיסי רב-קו:

https://law.acri.org.il//he/wp-content/uploads/2012/01/bus-card080112.pdf

https://www.digitalrights.org.il/files/2011-06-01/ravkav.pdf

- הנחיות על מכשירי תיקוף בתחבורה ציבורית (משרד התחבורה):

https://www.gov.il/BlobFolder/generalpage/guidelines_for_the_implementation_of_means_of_validation_in_public_transportation/he/1.76 הנחיות למימוש תיקוף בתחבורה הציבורית גרסה.pdf

- מכרז לבחירת ייעוץ לטכנולוגיית כרטוס חכם (חברת AMCG):

https://www.amcg.co.il/wp-content/uploads/2016/07/מכרז-יועץ-כרטוס-לפרסום.pdf

- חלק מפנקס מאגרי המידע בישראל שמותר לפרסום:

https://data.gov.il/dataset/pinkas/resource/fd56bf5b-7918-4906-99e4-b0e5102ae268

- מידע על כרטיסי רב-קו של ASK, באתר קומפיוטר גרד:

https://cguard.co.il/rb-qv-krtis-hnsieh-hhkm-wl-iwral.html

- כתבות על העתיד של רב-קו - תשלום באמצעות הסמארטפון:

https://www.calcalist.co.il/money/articles/0,7340,L-3770634,00.html

https://bizzness.net/תשאירו-את-הרב-קו-בבית-ב-2020-תשלמו-בסלולר/

https://www.pc.co.il/news/229515/

https://www.pc.co.il/featured/281817/

- כתבה "ישראלי בן 17 פרץ את כרטיס הנסיעה האלקטרוני רב-קו" באתר דה-מרקר:

https://www.themarker.com/technation/world/1.1764841

- פורומים בתפוז

http://www.tapuz.co.il/forums/viewmsg/394/176621949

http://www.tapuz.co.il/forums/viewmsg/394/171241300/

http://www.tapuz.co.il/forums/viewmsg/394/137430064//

http://www.tapuz.co.il/forums/viewmsg/394/157660655/

## מוצרים ישראליים לקריאת רב-קו

- רב-קו אונליין ו-HopOn:

https://ravkavonline.co.il/he/

https://hopon.co.il/read

- איפה בוס:

https://play.google.com/store/apps/details?id=il.co.mitug.WhereBus&hl=iw

- אפליקציית "רב קו" של אוטובוס קרוב:

https://play.google.com/store/apps/details?id=com.mosko.ravkav

- פרויקט של חברת Korentec (לא זמין):

https://www.korentec.co.il/portfolio-לאתר-ואפליקציית-web-native-android/