digital whisper
Technologic Papers

VoLTE - מהפכת הקול

מאת תומר גל נצר

הקדמה

VoIP (Voice over Internet Protocol) - טכנולוגיה המאפשרת לנהל שיחות טלפון אך ורק בעזרת האינטרנט. VoIP ממיר את הקול לפקטות ושולח אותם ל-IP בצד השני של השיחה. החיבור בין שני הצדדים נעשה בצורה דיגיטלית בעזרת Packet Switching. מכיוון שחוץ מחומרה בסיסית וחיבור לאינטרנט, VoIP כמעט לגמרי תוכנתי, אפשר להשתמש בו במחשבים, בטלפונים, בטאבלטים ובכל מכשיר התומך בתוכנה שלו.

הדרך שבה VoIP ממיר את הקול במיקרופון לרצף ביטים נקראת Voice Packetization. בדרך זו ADC שומר את הקול דיגיטלית לפי מיפוי של הקול בהתאם לפרקי זמן קבועים. לאחר מכן, משתמשים בcodec מתאים כדי לדחוס את הקול לצורכו המתאים בהתחשבות בזמן ריצה ושימוש במקום ביחס לאיכות הקול הדרושה. מכיוון שכל פקטה ב-VoIP היא בגודל של 60 בייטים, למען התחשבות במקום, ה-codec מאבד חלק מאיכות הקול וגורם לאיכות השיחה לרדת במעט למען שליחה מהירה של כל פקטה. לאחר שהמידע הקולי נדחס ומוכן להישלח, הוא מחולק לפקטות, כך שכל פקטה מכילה 20-30 מילישניות של מידע קולי.

החיסרון המרכזי של VoIP הוא דרישה לאינטרנט מהיר ויציב, אחרת איכות השיחה תרד משמעותית. הסיבה לכך היא שאיכות שיחה דרך VoIP היא מטבעה מאבדת מידע (בגלל סוג דחיסת ה-codec, שהוא lossy), אך כאשר האינטרנט לא יציב יכולות לקרות הבעיות הבאות:

1. הבעיה הכי שכיחה היא Packet Loss. כאשר אחת מהפקטות אובדות במהלך המסע שלהן מהשולח למקבל, בין 20 ל-30 מילישניות מהשיחה נעלמות. בכל פעם שפקטה אובדת, הברות או מילים שלמות יכולות להיעלם ולגרום לשיחה להיות קטועה.

2. עוד בעיה נפוצה היא High Latency. היא נובעת כאשר יש חיבור איטי בין השולח למקבל (לרוב נובע ממהירות העלאה איטית, או מרחק רב בין השולח למקבל). אם החיבור איטי, יהיה "דיליי" בשיחה והיא תהיה איטית ולא ברצף טבעי.

3. בעיה נוספת היא Jitter. בעיה זו נוצרת מחיבור איטי ולא יציב בין השולח למקבל. Jitter מתרחש כאשר לכל פקטה לוקח זמן שונה להגיע מהשולח למקבל. אם יש jittering המקבל ישמע מילים עם הברות מוחלפות או קטיעות קצרות ומיד לאחר מכן רצף הברות מעורבבות.

ברוב המקרים, ניתן לפתור בעיות אלו כשיש bandwidth גבוה יותר או שאיכות השיחה הכללית טובה יותר ומפצה על חלק מהבעיות.

כיום משתמשים ב-VoIP בהודעות קוליות ושיחות המאפשרות חיבור משלל מכשירים (לדוגמא שיחת "וואטסאפ" שניתן לענות לה מהמחשב או מהטלפון). לאור כל הבעיות של VoIP משתמשים בטכנולוגיה זו רק בתור מוצא אחרון - כשחיבור מכל סוגי המכשירים הוא דרישה חשובה.

בגלל החסרונות של VoIP, חיפשו פרוטוקול אחר. בחיפוש זה מצאו את VoLTE. אבל לפני שאסביר על VoLTE, אצטרך להסביר על LTE.

LTE (Long Term Evolution) - טכנולוגיית תקשורת אינטרנטית למכשירים ניידים הדומה במטרתה ל-VoIP. טכנולוגיה זו היא הדור הרביעי מסוגה (4G), אשר מהירה באופן משמעותי מקודמותיה בדור השלישי (3G), שהיה היבריד של המערכות UMTS ו-GSM (שתי מערכות המאפשרות להעביר מידע ממכשיר נייד. GSM קולט אינטרנט ממרחק גבוה יותר בקצב איטי יותר, ו-UMTS מהיר יותר פי 2 עד 18 מ-GSM אך שטח הכיסוי שלו נמוך. ההיבריד משלב בין כך שניתן יהיה לסמוך על האינטרנט וגם על המהירות כשצריך).

LTE משתמשת בפקטות כדי להעביר מידע. היא מתקשרת עם לווינים בעזרת היבריד של OFDMA ושל MIMO בתור ה-downlink (החיבור מהרשת למכשיר, דוגמה לכך היא הורדה של אפליקציה), וב-SC-FDMA בתור ה-uplink (החיבור מהמכשיר לרשת, דוגמה לכך היא העלאה של תמונה). בזכות הטכנולוגיה היעילה הזו, משתמשים ב-LTE עד היום לפעילויות הצורכות זרימת פקטות רבה ממכשירים ניידים - צפייה בסרטונים, משחקי טלפון, ועוד פעילויות הדורשות מהירות אינטרנט גבוהה.

VoLTE (Voice over LTE) - סוף סוף הגענו לנושא. עכשיו כשאנחנו יודעים את כל ההגדרות, אפשר להתחיל. VoLTE הוא פרוטוקול המבוסס על LTE ומושלך רק לתחום הטלפוניה.

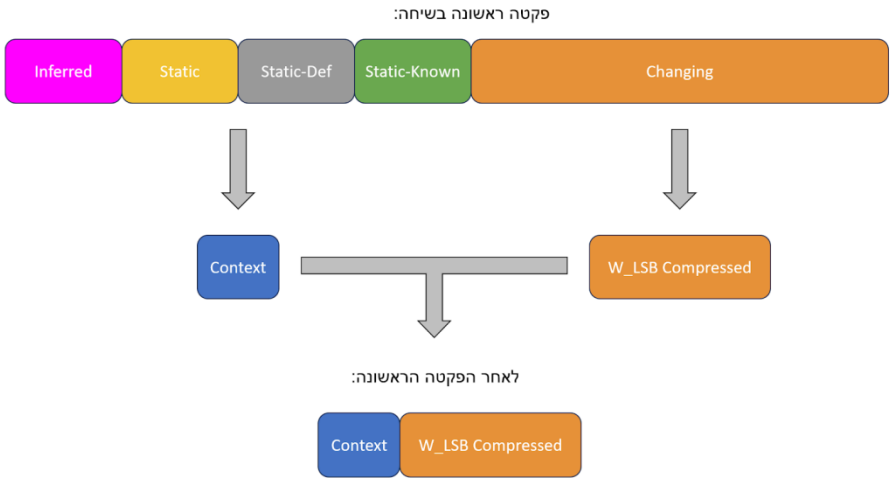
VoLTE מיועד אך ורק לשיחות בין שני מכשירים ניידים, כלומר הוא לא צריך לשלוח כמות משמעותית מהמידע ש-LTE משתמש כדי להכליל את הפקטה, מכיוון שהמידע נהיה סטטי. מה שנמחק מהפקטות הוא ב-header, כי 60% מה-metadata קבוע, ואין צורך לשלוח או לקבל אותו יותר מפעם אחת. לכן יצרו מנגנון שמכווץ את ה-header בשם RoHC. RoHC מנצל את העובדה שב-header דברים רבים לא משתנים במהלך השיחה, ומחלק אותו כך:

```
Inferred, Static, Static-Def, Static-Known, Changing
[IPv6 Header] = [Ver, Class, Flow ID, PL Len, Next Header, Hop Limit, Src IP, Dest IP] (40B)
[UDP Header] = [Src Port, Dest Port, Data Len, Checksum] (8B)
[RTP Header] = [Ver, Pad, Extension, RTP PL Type, Seq. Number, Timestamp] (12-17B)
```

- **Inferred**: מידע משתנה שניתן להבינו משאר הפקטה - במקרה שלנו זה אורך ה-Payload ואורך ה-Data.
- **Static**: המידע הסטטי נקבע בתחילת השיחה, ונשאר קבוע לאורך כל השיחה. הוא יכול להשתנות בין שיחה לשיחה.
- **Static-Def**: מידע קבוע שמגדיר את כל זרימת הפקטות ברשת - לאן להגיע ובאיזה port.
- **Static-Known**: מידע הידוע מראש לשני הצדדים. במקרה שלנו שני הצדדים יודעים מראש מהי גרסת ה-RTP.
- **Changing**: המידע הדינאמי של הפקטה. הוא משתנה בין פקטה לפקטה ולא ניתן לחזותו. הוא כולל בתוכו מספור של הפקטות (Sequence Number) למען השמעה של הפקטות למשתמש לפי הסדר (במקרה שפקטה אחת נשלחה אחרי השנייה והגיעה לפניה), ובשביל לזהות אם פקטה אבדה בדרך (קפיצה במספר הסידורי).

בתחילת השיחה, RoHC ישלח הודעה אחת מלאה וטבלה שממספרת סדר למשתנים. כך מקבל ההודעה ידע איך לפרק את המידע המוצפן בעתיד, ובו-זמנית יקבל את ה-20-30 מילישניות הראשונות בשיחה למען התחלה מהירה.

כעת, RoHC יצפין וישלח אך ורק את המידע הדינאמי, בעזרת דחיסה בשם W_LSB. בקצרה, W_LSB שומרת "חלון" (Buffer) של העבר) של הפקטות האחרונות שנשלחו. לפני ש-VoLTE שולח את הפקטה החדשה, W_LSB מחפשת בחלון את הפקטה הכי דומה לה. לאחר שמוצאת את הפקטה הרצויה, היא מציינת את הפקטה המדוברת ושולחת רק את ההבדלים בין הפקטות. לאחר כל הסידורים וההצפנות, ה-metadata יהיה באורך של בין שניים לשלושה בייטים ואורך ה-data יהיה בין 60 ל-65 בייטים:



קיימים מצבים שונים ל-RoHC שמשנים את דרך הדחיסה בהתאם למצבים שונים. למידע נוסף מומלץ לראות את הסרטון בקישור הבא, שמדבר על RoHC לעומק ומסביר על כל המצבים והסוגים השונים של הפרוטוקול.

AMR-WB

לפני שנעסוק בנושא AMR-WB, נצטרך להבין את שלושת הפרוטוקולים הבאים:

LPC (Linear Predictive Coding) - זהו פרוטוקול בסיסי השומר כמות של פקטות המאחסנות מידע קולי (מספר קבוע בין 5 ל-40), ומטרתו לחזות את הפקטה הבאה. תחילה למען פישוט הבעיה, LPC מניח שכל המידע הקולי שהוא מקבל, נוצר אך ורק על ידי הדברים הללו:

- באזר - צליל "מרובע" מושלם, כמו במשחקי מחשב ישנים.
- רעש סטטי אקראי - מדמה את הרעשים שיוצאים מתנועות הפה. דוגמאות בולטות הן האותיות "פ" או "ס", אך הרעשים תקפים גם לחריקת שיניים, צקצוק של הלשון בעת פתיחת הפה ועוד מקרים רבים.
- LPC מניח שכל הצלילים שהוא מקבל נשמעו דרך צינור, בתור הזנחה של תנועת פה כללית.

LPC מתחיל בכך שהוא מנקה את רעשי הרקע בעזרת טכניקה שנקראת Inverse Filtering (מומלץ לקרוא את המאמר של Matti Airas, Hannu Pulakka, Tom Bäckström ושל Paavo Alku העוסק בדרך בה הטכניקה עובדת) ושומר לבוא העת את הרעשים הסטטים ה"קיצוניים" (מעבר חד מרעש סטטי חלש לחזק, כדי להפריד בין דיבור לרעשי רקע חלשים).

לאחר ש-LPC מבודד את הבאזר מהרעשים הסטטיים, הוא מחשב ומוצא את העוצמה של הבאזר והתדר שלו, ושולח את שניהם יחד עם הרעשים הסטטים ששמר.

כאשר הצד השני מקבל תדר ועוצמה, הוא יידע להשמיע צליל באזר בעוצמה ובתדר שהוא קיבל, ללא הרעשים הסטטיים. כעת הוא יוסיף את הרעשים הסטטים שנשמרו ויקבל את צליל הבאזר המקורי ללא רעשים סטטים חלשים.

לאחר פתרון הבעיה של צליל באזר קבוע, נחזור לבעיה המקורית.

כעת, LPC יתייחס לדיבור בתור באזר שיכול לשנות את התדר והעוצמה שלו, מאחורי צינור שיכול לשנות את הצורה שלו (הרעשים הסטטים אקראיים בשני המקרים ולכן אין צורך לסבך אותם).

תחילה, LPC יוריד את הרעשים הסטטים בעזרת Inverse Filtering.

מכיוון שהבאזר משנה את התדר והעוצמה שלו, LPC ינסה למצוא תבנית בצורה שבה הבאזר משתנה. הוא ישמור 30-50 "פריימים" של דיבור (0.9-1.5 שניות של דיבור), וינסה למצוא צירוף לינארי שחוזה את הצלילים העתידיים בצורה הכי מדויקת (עם הסיכויים הכי גבוהים להצלחה). בשביל לחזות את הרעש הסטטי, LPC יפעיל אלגוריתם "רקורסיית לבינסון" וישמור את המקדמים. מכיוון ש-LPC לא מדויק לחלוטין, הוא יחסר את המידע שעיבד מהמידע המקורי, וישלח גם אותו.

הצד השני מקבל מקדמים ושארית שמראים בדיוק את ההווה, ויכולים לחזות בסיכויים גבוהים את 30-50 הפריימים הבאים.

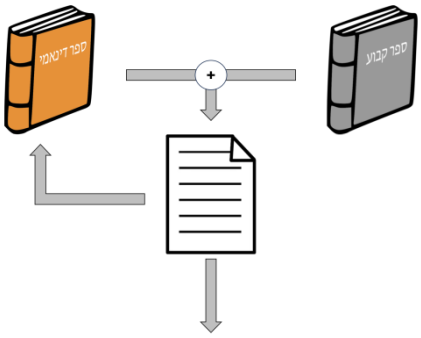
CELP (Code Excited Linear Prediction) - זהו אלגוריתם המבוסס על LPC, עם הבדל אחד קטן. הפרוטוקול יודע שדיבור אנושי הוא חזרתי וצפוי. לדוגמה - במהלך שיחה לא סביר שבן אדם לפתע ישנה את קולו לקול של אדם אחר. לכן ניתן להבין שגבהים ותדרים שונים במהלך השיחה יחזרו על עצמם פעמים רבות. בנוסף בין שיחות שונות קיימים דמיונות רבים בקולות. לדוגמה - במהלך שיחה לא סביר שאדם לפתע יתחיל לנגן בחצוצרה או שיעשה קולות רבים לא אנושיים.

לכן, CELP שומר "ספר" קבוע שמאחסן בתוכו תדרים ועוצמות נפוצים שאנשים משמיעים במהלך שיחה ממוצעת. הספר הזה נקרא "ספר קבוע", מכיוון שהוא לא משתנה בין שיחות שונות, והוא שמור חומרתית בפרוטוקול. בנוסף, CELP שומר עוד "ספר" בשם "ספר דינאמי", מכיוון שבתחילת השיחה הוא מתחיל ריק, אך משתנה במהלכה.

תחילה, CELP יעבוד בצורה זהה ל-LPC - הוא ימצא את כל המקדמים, הרעש הסטטי והשארית.

ראשית, הוא יחפש בספר הדינאמי את ה"עמוד" עם הנתונים הכי קרובים לשמע הנקלט ויחסר את העמוד מהנתונים המקוריים. לאחר מכן, הוא יחפש בספר הקבוע את העמוד עם הנתונים הכי קרובים להפרש. כעת, CELP יוכל להוסיף לספר הדינאמי את הסכימה של שני העמודים שמצא (מכיוון שאולי זהו צליל שיישמע שוב), וישלח את מספרי העמודים לצד השני.

כשהצד השני יקבל את שני המספרים, הוא יעבור לעמודים האלה, יסכום את הנתונים בהם, ויוסיף לספר הדינאמי שלו את הסכום. בצורה הזאת CELP הצליח לשלוח דיבור של 30-50 פריימים בעזרת שני מספרים בלבד.



ACELP (Algebric Code Excited Linear Prediction) - זהו פרוטוקול המבוסס על CELP, פרט להבדל אחד - דרך ייצור הספר הדינאמי. הדרך שבה ACELP מייצר את הספר הדינאמי היא בעזרת משוואות אלגבריות. הוא מוצא משוואות אלגבריות בספר הדינאמי, ומשלים אותן למשוואות אחרות ליצירת משוואה שיוצרת את הקול. הוא פרוטוקול שצורך הרבה חישובים ולכן יותר איטי, אך מחזיר תוצאות דחוסות יותר באיכות טובה יותר.

AMR-WB (Adaptive Multi-Rate Wideband) - זהו codec מתקדם המבוסס על ACELP, הנועד להעביר תקשורת קולית בצורה הטובה ביותר. הוא יכול לחזות פקטות עתידיות למען מניעת Packet Loss וייעול שליחת נתונים בעזרת ACELP. יש לו Echo Cancellation וגם Noise Reduction בזכות LPC, ותכונות רבות נוספות המאפשרות את חווית השיחה האידיאלית.

VoLTE משתמש ב-codec מסוג AMR-WB. בתחילת השיחה מתקיימת בדיקה לגבי האם שני הצדדים תומכים ב-AMR-WB. אם כן, השיחה תנוהל באמצעות codec זה. בתנאי שאחד מהצדדים לא יכול להשתמש ב-codec הזה, השיחה תנוהל באמצעות AMR-NB (זהו codec פשוט הנוצר במיוחד בשביל לתמוך בדיבור תחת אינטרנט בסיסי). AMR-WB משפר את איכות השמע, ותומך בטווח תדרים רחב יותר (50-7,000Hz לעומת 50-6,400Hz). קיימים יתרונות רבים לשימוש ב-AMR-WB, אך עמם מגיעים גם שני חסרונות עיקריים. החיסרון הראשון ב-AMR-WB הוא ה-bitrate הגדול. כלומר הוא דורש רשת עם bandwidth הרחב פי 2 משל AMR-NB. החסרון השני הוא ש-AMR-WB היא טכנולוגיה מתקדמת הקיימת רק במכשירים חדשים. אם לאחד מהמכשירים אין את ה-codec הזה, לא תהיה אפשרות לנהל את השיחה בעזרתו.

כעת, כל פקטה בסה"כ עדיין קטנה יותר מזו של LTE וגם ה-bandwidth הדינאמי גדול יותר. לכן השיחה זולה יותר, מהירה יותר, וגם השמע בה ברור יותר. מומלץ לקרוא את המאמר של Disscom שמפרט ומעמיק בנושא גלי רדיו וכיצד הם מנוהלים - נשלחים, נקלטים ומתורגמים.

חשוב לציין ש-VoLTE משתמש ב-QoS למען תעדוף תעבורה של השיחה מעל שאר הפקטות במכשיר. טכניקה זו משפרת את כמות הזמן ששיחה נפתחת, את ה-ping בשיחה, ומנמיכה את הסיכויים לניתוק בין שני הצדדים. VoLTE מאפשר חיבור בו זמני של שיחה ושל פקטות אחרות ויודע לסנן ביניהם בצורה יעילה. לכן אפשר לנהל שיחה בעזרת VoLTE וגם לגלוש באינטרנט בו זמנית.

רקע היסטורי

בשנת 1991, VoIP פורסם לציבור על ידי חברת מחקר ופיתוח בשם VocalTec. החברה הפקידה פרוטוקול חדש בשם Digital Voice Transfer, שנחשב לתחילת התפתחות VoIP כפי שאנו מכירים היום. ב-1993, VocalTec השיקה את המוצר הראשון שלה בשם VocalChat, שהותקן על מחשבים אישיים ואפשר למשתמשים לתקשר באמצעות DVT.

עם התפשטות האינטרנט והשיפור ברוחב הפס ברשתות, VoIP התחיל להתפשט יותר. חברות רבות כגון Microsoft, Intel, Radvision החלו לפתח ולמכור מערכות VoIP מתקדמות למשתמשים עסקיים ופרטיים. היתרון העיקרי של VoIP היה היכולת להציע שיחות טלפון במחירים נמוכים יותר משיחות הטלפון המסורתיות.

ב-1995, VocalTec פרסמו טלפון שיכול לבצע שיחות בעזרת VoIP. ואיך קראו לו? Internet Phone או בקצרה iPhone (לא ה-iPhone שאתם מכירים, מוצר שונה של חברה אחרת תחת אותו שם).

בשנת 2003 פותחה Skype. האפליקציה הזו תרמה לדמוקרטיזציה של VoIP בקנה מידה עולמי.

מכיוון שלאחר פיתוח Skype כולם החלו להשתמש ברשת במכשירים ניידים לתקשר בזמן אמת, נוצרה עדיפות ליצור פרוטוקול חדש המיועד לייעול שליחת נתונים בין מכשירים ניידים.

חברת בשם Third Generation Partnership Project כנראה גם חשבה כך, כי יצרה פרוטוקול חדש ושמו Long Term Evolution, או בקצרה LTE.

חברת טלפונים יפנית בשם NTT Docomo נירמלו לראשונה את LTE ב-2004, וב-2006 Nokia הציגו בשידור חי את LTE לתקשורת. הם הציגו שני אנשים שמשתפים סרטון באיכות HDTV בזמן ששיחקו משחק אינטראקטיבי.

ב-2007 בפברואר, חברה בשם Ericsson הציגו לעולם לראשונה LTE המצליח לשלוח פקטות בקצב 144 Mbit/s. שבעה חודשים לאחר מכן, NTT Docomo הדגימו מהירויות עולות על כל דמיון של 200 Mbit/s.

ב-2008 ב-Mobile World Congress הוצגה לראשונה השיחה הראשונה שהופעלה לחלוטין בעזרת פרוטוקול LTE! ה-"שיחה" הזו הייתה Proof of Concept של שיחה המופעלת על מכשיר קטן שבקושי אפשר להגדיר בתור טלפון. אך בספטמבר 2009, ב-MWC הוצג לראשונה שיחה שהופעלה בעזרת פרוטוקול LTE, אבל בין טלפונים אמיתיים!

זוהי הייתה קפיצת דרך אחת מני רבות, אך אחת מקפיצות הדרך המשמעותיות של LTE היו ב-2011. בשנה הזו חברה בשם Sri Lanka Telecom Mobitel הדגימה לראשונה פיתוח של LTE בשם 4G LTE - עם מהירויות של עד 1,000Mbit/s! בתקופה הזו, בשל הפרסום הרב של שיחות ניידות בעזרת LTE ופיתוח רוחב הקו, היה נדרש שיפור רציני בפרוטוקול לשיחות טלפון ניידות.

בשנת 2012, חברת רשת ותקשורת בשם MetroPCS ענתה על דרישה זו בכך שפיתחה את **VoLTE** (הגענו לנושא!) יחד עם הטלפון LG Connect 4G שלנצח יהיה הטלפון הראשון עם טכנולוגית VoLTE.

ביוני 2014, שני ספקי אינטרנט אמריקאים גדולים, Verizon Wireless ו-Wireless AT&T, פרסמו שהם מאפשרים שיחות "VoLTE-to-VoLTE" ללקוחות שלהם. ב-2015 החברות החלו לעבוד על דרך לנהל שיחת VoLTE-to-VoLTE בין חברות שונות. הם שיתפו פעולה והשתמשו ברשת של חברת ציוד לתקשורת Alcatel-Lucent. הפרויקט נמשך עד נובמבר 2017.

בנובמבר 2017 עד 2020, VoLTE נהיה נגיש לציבור בזכות הפרויקט, התפרסם וכעת כמעט כל הטלפונים משתמשים בו.

ReVoLTE Attack

אוי לא! בשנת 2019 חבורה של חוקרים מאוניברסיטת Ruhr מצאו פרצה ב-VoLTE! הם קוראים לפרצה שמצאו בשם ReVoLTE Attack, והיא מנצלת בעיית אבטחה חמורה שקיימת ב-VoLTE. ליתר דיוק, בעיית האבטחה כבר הייתה קיימת במקור ב-LTE והבעיה הזו השתרשרה ל-VoLTE. בעיית האבטחה היא בעיה של שימוש במפתחות הצפנה חוזרים. החוקרים גילו שמפתח ההצפנה תלוי באנטנה המחוברת לשיחה ולא בשיחה אינדיבידואלית - לכל אנטנה יש מפתח הצפנה קבוע משלה.

לפיכך, החוקרים יצרו מתקפת Stream Cipher העובדת באופן הבא:

1. להסניף (לקרוא ולשמור את הפקטות) את השיחה שרוצים לשמוע - השיחה שהוסנפה מוצפנת.
2. מיד לאחר סיום השיחה, להתקשר לאחד מצדדי השיחה - בתנאי והיו מספיק מהירים, והשיחה תתנהל על אותה אנטנה.
   א. להסניף את השיחה החדשה - השיחה המוסנפת גם מוצפנת לפי אותו Stream Cipher.
   ב. להקליט את השיחה הנשמעת - הקול מהצד השני, לא מוצפן.

לאחר כל השלבים הללו, נקבל את הדברים הבאים:

1. שיחה A מאורך *n* ביטים מוצפנת במפתח K - אנחנו יודעים את $A \oplus K$.
2. א. שיחה B מאורך *m* ביטים מוצפנת במפתח K - אנחנו יודעים את $B \oplus K$.
   ב. שיחה B מאורך *m* ביטים לא מוצפנת - אנחנו יודעים את B.

מכיוון שאנחנו יודעים גם את B וגם את $B \oplus K$ (ושניהם מאורך שווה) נפעיל xor על שניהם ונקבל:

$$B \oplus (B \oplus K) = (B \oplus B) \oplus K = 0 \oplus K = K$$

לכן נראה שקיבלנו את מפתח ההצפנה! אבל זהו לא בדיוק K, אלא זהו K באורך *m* ביטים - זאת מכיוון שרק *m* ביטים מ-K שומשו בשביל להצפין את B. נקרא ל-K מאורך *m* בשם K_m.

כעת, נפעיל את K_m על m הביטים הראשונים של השיחה המוצפנת $A \oplus K$, ונקבל:

$$(A \oplus K)_m \oplus K_m = (A_m \oplus K_m) \oplus K_m = A_m \oplus (K_m \oplus K_m) = A_m \oplus 0 = A_m$$

ונראה שקיבלנו את m התווים הראשונים של השיחה **הלא** מוצפנת. לפיכך, בתנאי והשיחה B יותר ארוכה מ-A, נוכל להשיג את כל השיחה A, אחרת נוכל להשיג m ביטים מ-A.

למזלנו, היוצרים של ReVoLTE לא השתמשו בפרצה הזו למטרות זדוניות והתריאו ספקי אינטרנט רבים לגביה. למרות שהרוב המוחלט של הספקים הוסיפו עוד הצפנות מעל ה-Stream Cipher ו/או שינו את המערכת לייצור מפתחות כך שלא יהיו חזרות, חלק מהספקים לא עשו דבר בנידון.

בשביל להימנע משיחות לא בטוחות, החוקרים יצרו אפליקציה הבודקת אם האנטנה המחוברת לטלפון שלך מוגנת מ-ReVoLTE Attack או לא. אני אישית ממליץ לבדוק את האפליקציה הזו.

למידע נוסף, מומלץ להסתכל על האתר של החוקרים, ו/או על המצגת של Priya Chalakkal הבודקת את האבטחה של VoLTE ומראה פרצות נוספות.

SIP Buffer-Overflow Attack

בתהליך יצירת שיחת VoLTE, שני הצדדים קובעים ומאשרים הגדרות רבות כגון ה-codec, תיאום פרוטוקולים שונים ותיאום כתובות הרשת. פרוטוקול זה נקרא SIP (Session Description Protocol).

ה-SIP מוגדר בשיטה text-based בצורה הבאה:

```
<character>=<value><CR><LF>
```

כאשר <character> מתייחס לאות אנגלית גדולה או קטנה, <value> מתייחס לערך שיוגדר ב-<character>, ו-<CR><LF> הוא ייצוג ASCII של ירידת שורה (<CR> מזיז את האותיות שמאלה ו<LF> מוריד את האותיות למטה).

דוגמה לשורת SIP:

```
o=JohnDoe 2890844526 2890842807 IN IP4 10.47.16.5
```

במקרה הזה, נראה שה-<character> הוא o (יוזם השיחה - Originator). ה-<character> הזה מבקש את הדברים הבאים:

o=Username, ID, Version Number, Network Address

האקרים יכולים לנצל את העובדה ש-SIP מוגדר בשיטה text-based. הם יכולים לשלוח SIP תקין, אך עם <value>-ים מאוד גדולים. לדוגמה, להזין ב-o שם משתמש מזויף בגודל העובר את ה-buffer, לדוגמה 1MB (מכיוון ש-SIP משתמש ב-UTF-8, אורך השם בדוגמה שלי יהיה בין 262,144 אותיות מהרשימה הבאה לבין 1,048,576 אותיות אנגליות).

לתופעה שבה כמות המידע הנשלח גדול מכמות המידע שה-buffer יכול להכיל קוראים Buffer Overflow. הבעיה ב-Buffer Overflow היא שאם לא מגבילים את הגודל של שליחת ה-buffer / תוחמים מקום ל-buffer בזמן העיבוד, הוא יכול לדרוס ולהחליף מידע אחר בקוד של המכשיר המקבל את ה-buffer.

כשיש להאקרים את היכולת לכתוב מידע משלהם מחוץ לקטע המוגדר להם, הם יכולים להשתיל קוד משלהם במכשיר המקבל את ה-SIP ששלחו.

בעזרת היכולת הזאת, הם יכולים להשתיל קוד למכשיר ולקבל שליטה מלאה על המערכת.

למזלנו, רוב ספקי האינטרנט כבר פתרו בעיה פשוטה זו. בשביל ליישם Buffer Overflow Attack, התוקף נדרש לשלוח הודעה ארוכה מה-buffer. לכן, ספקי האינטרנט מונעים שליחת SIP הארוך מגודלו של ה-buffer. כך יימנע כל סוג של Buffer Overflow ומתקפות כאלו לא יוכלו להתרחש.

פרצות לא פתורות

SIP Spoofing - גם בגבול החדש של ה-buffer, עיוות של ה-SIP מאפשר להאקרים לזייף את הזהות שלהם. הם יכולים לגרום למכשיר שלהם להתחזות לכל מכשיר אחר, בין אם זה לבנק מפורסם, למוקד בית חולים או אפילו למשטרה.

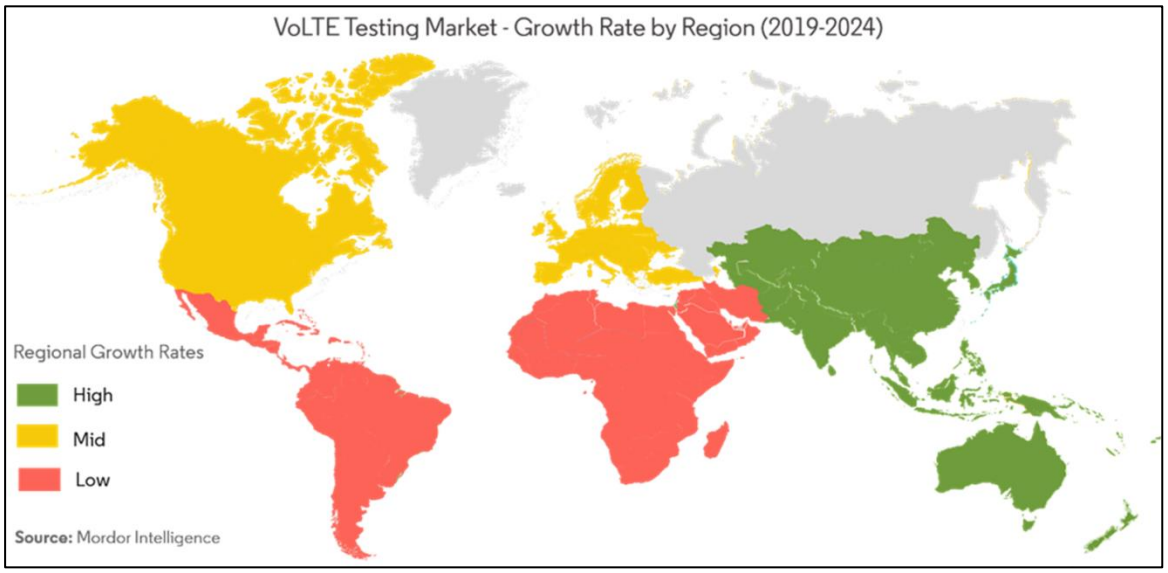
SIP Sniffing - האקרים יכולים להסניף מהמכשיר הנייד שלהם את ה-SIP של הצד השני, אם הוא ענה להם לשיחה. בעזרת מידע זה הם יכולים להשיג את ה-IP של הצד השני (נותן מיקום עם רמת דיוק של מגדל התקשורת / הראוטר הקרוב ביותר) ואת מספר ה-IMEI שלו (שונה לכל מכשיר, זה מספר שניתן לפענח ממנו בדיוק איזה מכשיר בצד השני של השיחה).

שילוב של שתי הבעיות הללו נותן להאקרים את האופציה להתחזות בקלות לגורם שאחרים לא יהסו לענות לו, וכשעונים להם, לשאוב מידע על המכשיר למען מציאת פרצות ספציפיות אחרות למכשיר מסוים.

קיימות פרצות רבות נוספות ב-VoLTE וממשיכות להתגלות פרצות חדשות ככל שהזמן עובר, אך גם פרצות רבות נפתרות והפרוטוקולים ממשיכים להתעדכן.

שימוש כיום

כיום, כמעט כל הטלפונים משתמשים ב-VoLTE, וספקי אינטרנט רבים עוברים לתמוך ב-VoLTE - עוברים מרשת Circuit Switched לרשת IP-Centric (העברת פקטות ממקום למקום בעזרת IP בעזרת טכנולוגיית 4G/5G). הינה מפה מ-2019 שאומדת כמה התמיכה ב-VoLTE תגדל עד 2024:



[מקור: https://www.mordorintelligence.com/industry-reports/volte-testing-market]

בסך הכל, VoLTE חולל מהפכה בתקשורת הקולית על ידי שדרוג היכולות של רשתות LTE, איכות קול מעולה יחסית לקודמים לו וניהול שיחות ושימוש בנתונים ניידים בו זמנית. VoLTE הפך לסטנדרט לשיחות קוליות ברשתות 4G וככל שרוחב הפס יתרחב, ו-5G יתנרמל, היכולות שלו יורחבו ל-5G ושיחות ברשת יהיו מהירות וברורות יותר מתמיד.

סיכום

לאחר חקירה ממושכת והתעמקות בפרוטוקול VoLTE, נראה של-VoLTE יש יתרונות רבים על פני פרוטוקולים אחרים בתחומים רבים. במהלך כתיבת המאמר וככל שבדקתי במאמרים שונים בכלי תקשורת מגוונים, נותרה עדיין השאלה - האם מדובר בפרדוקס? מצד אחד, אנשים רבים ברחבי העולם משתמשים בפרוטוקול, ואילו מצד שני, אם מדובר בפרוטוקול שכיח מדוע מציינים בכלי התקשורת ובמאמרים רבים רק את חסרונותיו?

בעת סיום כתיבת המאמר ולאחר קריאתו מחדש, הבנתי עד כמה ה"פיל" הוא בעצם "עכבר". במאמרים השונים ל-VoLTE יש "סטיגמה" שלילית, ובמאמר שלי הצלחתי להדגיש את היתרונות הבולטים שלו, בין אם בדחיסה עצמה לבין ה-codec המיוחד שנבחר וכך יכולתי להתנגח ב"סטיגמה".

על המחבר

שמי **תומר גל נצר**, בן 16 מתל אביב, תלמיד כיתה י"א בכיתת מופ"ת בתיכון "הכפר הירוק" ברמת השרון. כמו כן אני תלמיד שנה ב' באוניברסיטת "תל אביב" ב"מרכז מדעני העתיד" בתכנית "אודיסאה".

הכתיבה תרמה לי בהכרת הנושא באופן הוליסטי וההתעמקות בו הפכה למשמעותית ביותר בהבנה אמיתית, בצורך לחקור וברצון למקסם ולרכז את כל המידע בצורה מיטבית.

בחרתי לכתוב בנושא שתמיד עניין אותי - VoLTE. מטרתי העיקרית בכתיבת המאמר היא להעלות לתודעה הציבורית את הבעיות והפרצות השונות שכרוכות בשיחות טלפוניות, ולחפש האם יש יתרונות בשימוש ב-VoLTE על פני פרוטוקולים אחרים.

בעיקר התעמקתי ב-VoLTE בפירצה בשם ReVoLTE Attack. בכל בית אב בישראל בפרט ובעולם בכלל מומלץ ואפילו הכרחי להיות מודע לסיכונים הכרוכים ב-ReVoLTE וכיצד להימנע מהם.

ברצוני להודות ל**ד"ר שלומי בוטנרו** ו**אורן רנרד** על הליווי המקצועי והעצות המועילות במהלך כתיבת המאמר.

מושגי יסוד

Packet Switching - טכנולוגיה בה מקורות תקשורת מתקשרים בעזרת חלוקה של המידע שלהם לרצפי ביטים קצרים בשם "פקטות". בפקטה שמור המקום שהיא צריכה להגיע אליו בעזרת שיטת מספור של ראוטרים ברשת שנקראת IP. בפקטה שמורים שני מספרים: הראשון מיועד להגיע למקור תקשורת הנכון והשני מיועד להגיע לתוכנה המתאימה במקור התקשורת. ניתן להבין את המושג בהשוואה לשליחת חבילה למען מסוים - החבילה נשלחת לנמען בכתובת מגורים כלשהי, ועל המשלוח מצוין תיאור החבילה. בנוסף למידע הזה היא שומרת חלק מהמידע שהלקוח ביקש - טקסט / תמונה / קובץ להורדה וכו'.

ADC - ראשי תיבות של "Analog-to-Digital". מכשיר שקולט ויברציות קול ושומר אותן בתור רצף ביטים, המתארים בצורה הכי מדויקת (לפי רעש) את הויברציות בפרקי זמן קבועים.

Codec - המילה נוצרה משילוב המילים decoder + coder. זהו מכשיר פיזי או תוכנת מחשב, שנועדה לקודד ולפענח זרם של ביטים. הסיבה שמשתמשים בסוגי codec שונים, היא שלכל מטרה שונה, הצורכת שליחת זרם ביטים, נדרש סוג דחיסה שונה - כזו שתוכל לדחוס את זרם הביטים בצורה הטובה ביותר.

ההיבריד של OFDMA ושל MIMO - OFDMA היא טכניקה ייעודית למספר משתמשים שיוכלו להתחלק באותה רצועת תדרים באותו זמן. היא עובדת על ידי חלוקת רצועת התדרים לתת-תדרים, הניתנים למשתמשים שונים. כל תת-תדר יכול להפעיל נתונים עבור משתמש אחד ומשתמשים מרובים יכולים להיות משוייכים לאותו תת-תדר באותו זמן, כל עוד הם מופרדים בזמן או בתדר. הדבר מאפשר למספר משתמשים לשתף את אותה רצועת תדרים ללא הפרעה זה לזה. MIMO היא טכנולוגיה המשתמשת במספר אנטנות כדי לשפר את השידור והקליטה של הנתונים. היא יוצרת מספר זרמי נתונים בין תחנת הבסיס והמכשיר הנייד, ובכך מאפשרת שימוש יעיל יותר בטווח התדרים הזמין. ההיבריידיות של שניהם מאפשרת למשתמשים רבים להיות שייכים לאותה רצועת תדרים, שיעברו לאנטנות שונות בהתאם לצורך המיקום של הפקטה. ההיבריידיות מאפשרת ל-LTE שטח רחב יותר, קצבי נתונים גבוהים יותר, אמינות משופרת וייעול של מעבר הפקטות.

Single-Carrier Frequency-Division Multiple Access - SC-FDMA הוא סוג של Frequency Division Multiple Access. בצורה פשוטה זוהי דרך לחלק גלי רדיו ללקוחות שונים. בניגוד ל-FDMA בסיסי אשר מחלק את הרדיו לפי תדרים שונים, SC-FDMA עובד על אותו התדר בעזרת טכניקה שנקראת Single Carrier. בתחילת כל פקטה שעוברת מועבר Carrier ייחודי לכל לקוח, אשר מסמל למי הפקטה שייכת. VoLTE משתמשת ב-SC-FDMA ולא ב-FDMA בסיסי מכמה סיבות. הסיבה העיקרית היא ש-SC-FDMA מאפשר טווח רחב יותר מאשר FDMA בסיסי, ובכך ניתן יהיה להשתמש באותו תדר ללא גריעה באיכות השיחה.

בנוסף, השימוש ב-SC-FDMA חוסך בחשמל - היחס בין הכמות המקסימלית של חשמל ש-SC-FDMA צורך לבין הכמות הממוצעת, הוא מאוד נמוך ביחס לשאר הפרוטוקולים של FDMA. הסיבה ליחס הנמוך מתרחשת בגלל שימוש בטכניקה בשם Discrete Fourier Transform. הטכניקה הזו מפזרת מידע בצורה שווה על גבי הרבה Carriers שונים למען נרמול צריכת החשמל ביניהם. כך כמות התנודות בצריכת החשמל יהיו מינימליות באופן שמייעל אותה. כך משתמשים לא צריכים לחשוש מצריכה רבה בבטרייה בזמן ניהול שיחה.

Quality of Service - ערך שניתן להשתמש בו כדי להקצות משאבי רשת ותעדוף ברשת לפי דרישות מסוימות. כשמשתמש מבצע שיחת VoLTE, ה-QoS מבטיח שהשיחה תקבל את העדיפות הגבוהה ביותר על פני נתונים אחרים שעשויים לעבור ברשת באותו זמן. הוא עושה זאת על מנת להבטיח שהשיחה תישאר עם זמן תגובה נמוך וכמות מינימלית של הפרעות.

מתקפות "Stream Cipher" - Stream Cipher: זוהי הצפנה מאוד מהירה, שנעשית באמצעות הפעלת xor על כל ביט בהודעה המקורית לפי מפתח שנקבע על ידי סדרה בינארית פסאודו אקראית. כלומר, לשני הצדדים קיים אלגוריתם שיוצר בזמן אמת אפסים ואחדים בצורה שנראית אקראית, אבל היא קבועה. בהנחה ששני הצדדים יפעילו אותו אלגוריתם, הם יקבלו את אותו מפתח.

נניח שהמידע המקורי הוא A, והמפתח הוא K. לאחר הצפנת Stream Cipher נקבל $A \oplus K$. הדרך שבה הצד השני ממיר בחזרה את המידע המוצפן הוא בעזרת הפעלה נוספת של xor. בהנחה שהצד השני יקבל $A \oplus K$, אם הוא יודע את המפתח, הוא יוכל לעשות את הפעולה הבאה:

$$(A \oplus K) \oplus K = A \oplus (K \oplus K) = A \oplus 0 = A$$

החסרון היחיד הוא שהמפתח חייב להיות לא צפוי. אחרת כל אחד יכול להפעיל פעולת ה-xor ולקבל את ההודעה המקורית. מתקפת "Stream Cipher" יכולה להתרחש כאשר ניתן לגלות בצורה כלשהי את K או חלק ממנו. אם ניתן לגלות רק *n* ביטים מ-K, נוכל להפעיל על אותם ביטים את פעולת ה-xor עם המידע המוצפן ונקבל *n* ביטים של המידע המקורי.

הסבר אינטואיטיבי על Buffer Overflow - אפשר לחשוב על כך בצורה הבאה: נסתכל על הזיכרון של המכשיר בתור רחוב, ועל כל רצף ביטים בתור בניין עם כתובת. כאשר המכשיר מקבל SIP buffer, נוכל להתייחס לכך בתור הנחייה לבנות בניין חדש בכתובת x בהתאם לתוכניות הבנייה שנשלחו. הבעיה מתחילה כאשר אנחנו מקבלים הנחייה לבנות בניין הגדול מהשטח הנתון לנו. הפועלים יתחילו לבנות את הבניין בכתובת x משמאל לימין. הם יגיעו לקצה הימני של השטח המוקצב לבניין, אבל יראו שחלק מהבניין עדיין לא נבנה - אז הם יגלשו לשטח הבית של השכן, יבנו בו וכתוצאה מכך יהרסו לו חלק מהבית או את כל הבית. ככל שתוכניות הבנייה יותר גדולות, הפועלים יהרסו יותר בתים מימין ויחליפו אותם בבניין המשורטט בתוכנית.

ביבליוגרפיה

- https://en.wikipedia.org/wiki/Voice_over_IP
- https://en.wikipedia.org/wiki/Session_Description_Protocol
- https://en.wikipedia.org/wiki/Packet_switching
- https://www.geeksforgeeks.org/difference-between-message-and-packet-switching/
- https://www.lifewire.com/circuit-switching-vs-packet-switching-3426726
- https://en.wikipedia.org/wiki/Analog-to-digital_converter
- https://en.wikipedia.org/wiki/Codec
- https://en.wikipedia.org/wiki/LTE_(telecommunication)
- https://www.techtarget.com/searchmobilecomputing/definition/Long-Term-Evolution-LTE
- https://en.wikipedia.org/wiki/MIMO
- https://en.wikipedia.org/wiki/Single-carrier_FDMA
- https://en.wikipedia.org/wiki/Robust_Header_Compression
- https://www.ietf.org/rfc/rfc5225.html#page-121
- https://en.wikipedia.org/wiki/Linear_predictive_coding
- https://en.wikipedia.org/wiki/Linear_prediction
- https://en.wikipedia.org/wiki/Code-excited_linear_prediction
- https://en.wikipedia.org/wiki/Algebraic_code-excited_linear_prediction
- https://en.wikipedia.org/wiki/Adaptive_Multi-Rate_Wideband
- https://en.wikipedia.org/wiki/Adaptive_Multi-Rate_audio_codec
- https://en.wikipedia.org/wiki/Quality_of_service
- https://en.wikipedia.org/wiki/Stream_cipher
- https://en.wikipedia.org/wiki/Buffer_overflow
- https://www.kaspersky.com/blog/volte-insecurity/10463/
- https://www.sharetechnote.com/html/IMS_SIP_SDP.html
- https://web.archive.org/web/20201127132600/https://www.metaswitch.com/blog/evaluating-volte-security
- https://www.itnews.com.au/news/volte-can-be-abused-to-track-and-spoof-users-464943