Technologic Papers

הצפנת VoIP למתחילים

מאת עידו קנר

מבוא

במאמר קצר זה, אני אנסה להסביר לאנשים הנוגעים ב-VoIP על סוגי הצפנות, ואף להסביר מדוע הן קיימות, כאשר הדגש הוא להבין מה לבחור, כיצד הצפנות אלו משפיעות על התהליך, ופחות לימוד על ההצפנות עצמן, אשר דורשות איש מקצוע בנושא.

המאמר הוא רק התחלה, ומנסה לגשת לאנשים ללא ניסיון קודם בהצפנות הקשורות לתקשורת, והוא מביא לקוראים את הידע בנושא מבחינת היכולת לבחור, למרות שאין הבנה מלאה אודות ההצפנה עצמה.

היסטוריה קצרה

עולם הטלפוניה התפתח יחסית לאט מאוד, ועם כניסת רשת האינטרנט לעולם אשר מחוץ לאקדמיה, עולם הטלפוניה תרגם את עצמו גם לעולם האינטרנט. למעשה עד שנות ה-70, ואף תחילת שנות ה-80, הגישה של עולם הטלפוניה היתה זהה, ומה שהתפתח הוא למעשה רק אוטומציה של אותה גישה.

למשל, בשביל להצליח לקבל הרבה שיחות, היה לוח קיר עם המון מתגים, אשר היו מחברים את השיחה שלך עם "ערוץ פתוח". כאשר "ערוץ פתוח", הוא למעשה שקע פנוי להעביר את השיחה לשם. זה נעשה על ידי מרכזניות - לרוב נשים אשר היו עונות לשיחה ומעבירות אותה הלאה למרכזיה אחרת העובדת באותה צורה, עד אשר הגעת ליעד אם בכלל.

מה קורה כאשר אין ערוץ פנוי? אתה מחכה בלופ אין סופי עד שיענו לך ויעבירו אותך הלאה. וכאן בשביל לעשות את זה נכון, יצרו את תורת התורות בעולם המתמטיקה, גישה הנמצאת בשימוש עד היום. החיוג עצמו, במכשיר הטלפון גם נעשה בצורה אנלוגית פשוטה מאוד, של מספר "מנופים" אשר היו מייצגים X ספרות, ולאחר מכן, היו מעבירים את הבקשה הלאה לפי מתחים שונים אל אותה מרכזיה.

העולם לקח את הגישה הזו ורק ביצע לה אוטומציה, בהתחלה אוטומציה אנלוגית, ומשנות ה-80 בערך, התחילו לקחת את הגישה גם בצורה דיגיטאלית, כדוגמת שימוש בצלילים עבור מקשים המוכרים בשם - DTMF או בשמם המלא Dial Tone Multi-Frequency Signaling.

עם כניסת האינטרנט (TCP/IP) לעולם העסקי בתחילת שנות ה-90, החלו לבצע פעולות דיגיטליות נוספות, וזה להמיר את עולם הטלפוניה לרשת האינטרנט, דבר שהמציא את העולם שאנחנו מדברים עליו עכשיו בשם VoIP.

המונחים נשארו אותם מונחים, הבעיות אותן בעיות, רק בצורה תוכנתית ופחות אנלוגית וחומרתית, וכמובן בעיות חדשות המגיעות מעולם ה-TCP/IP.

כיום, מרבית הטלפוניה בעולם היא טלפוניה מבוססת VoIP, גם כאשר ציוד הקצה הוא טלפון פשוט. זאת המפכה שכיום נמצאת בשיאה עם הכנסת רשת ה־NGN, אשר יצרה מענה דיגיטאלי מלא עם VoIP לעולם הטלפוניה.

מושגי יסוד

כאשר נוגעים בעולם הטלפוניה ברשת האינטרנט, נתקלים במספר מושגים אשר אסביר עליהם בקצרה:

- VoIP - Voice Over IP
- VoB - Voice over Broadband
- SIP - Session Initiation Protocol
- SDP - Session Description Protocol
- RTP - Real Time Protocol
- RTCP - Real-Time Control Protocol
- WebRTC - Web Real Time Communication

כאשר אנחנו מדברים על VoIP או VoB, אנחנו מדברים על מדיה העוברת תחת פרוטוקול TCP/IP, כאשר נהוג לרוב להשתמש ב-UDP לשם יצירת תעבורה, אך אין זה מחייב שזה יהיה תמיד המצב.

ההבדל היחיד בין VoB לבין VoIP הוא בכך ש-VoB מדבר על עבודה עם רוחב פס רחב, בעוד ש־VoIP מדבר על עבודה עם כל מהירות תקשורת באשר היא, וזה כאמור ההבדל היחיד בניהם.

SIP הוא פרוטוקול המאפשר לבצע איתות אודות מצב השיחה, למשל התחלה של שיחה, מצב של צלצול, מצב של מענה, מצב של תפוס, מצב של טלפון (למשל לשים בהמתנה את השיחה), ועוד מגוון רחב של אותות. בנוסף הפרטוקול מאפשר להרכיב על עצמו עוד פרוטוקולים שונים, ברמת ה־body, אשר הנפוץ בפרוטוקולים אלו הוא SDP.

התפקיד של SDP הוא למעשה לתאר בין היתר איך המדיה תעבור, והוא מבצע זאת על ידי מספר שדות, אשר חלקם מקבל מאפיינים (attributes) המאפשרים מידע נוסף או בחירה של דברים. המדיה שעוברת

בפועל, מתבצעת על ידי פרוטוקול בשם RTP, אשר כמעט תמיד (למעט מקרים יוצאי דופן), נעשה תחת UDP. השליטה ב RTP מתבצעת על ידי פרוטוקול נוסף, אשר נקרא RTCP.

הרעיון של RTCP הוא לשלוט במדיה העוברת תחת RTP, והוא סוג של צורת שליטה בפועל של התעבורה, כדוגמת שליטה בגודל באפר (Jitter Buffer), זמני תעבורה, סטטיסטיקות על התעבורה ואפילו שליטה של QoS.

בנוסף למה שנעשה עד היום, ישנם מעט התפתחויות בתחום.

למשל בשנתיים האחרונות נכנס פרוטוקול חדש לעולם המדיה, אשר קיבל את הכותרת Web Real Time Communication. הוא למעשה מאפשר לשלוח מדיה באמצעות תיאור של SDP (וכיום עם פרוטוקול חדש נוסף אותו אזכיר בקצרה עוד מעט), RTP־וRTCP ישירות מדפדפן, או כל מימוש אחר אל קצה כלשהו בצורה טבעית. WebRTC מחייב לפי התקן, תקשורת מאובטחת, אך הוא אינו נוגע באיתות ומשאיר זאת למי שמשתמש בפרוטוקול, אך הפרוטוקול מחייב כי גם האיתות יתבצע בצורה מאובטחת.

היות ו־WebRTC אינו מחייב דפדפן, יש לו מימושים למערכות שונות, וכל מימוש מקבל את תכונות אלו לפי התקנים בנושא. שימוש העיקרי של WebRTC הוא בדפדפנים שונים, כאשר ישנם שלושה דפדפנים עיקריים שתומכים בו (נכון לכתיבת מאמר זה): פיירפוקס, כרום ואופרה. מיקרוסופט הודיעה כי תתמוך אף היא בפרוטוקול בדפדפן "חדש" שהיא מפתחת.

גרסה 1.1 ל-WebRTC (הגרסה האחרונה נכון לכתיבת מאמר זה) קיבלה את השם ORTC - Object Real-Time Communication, ומספקת צורה אחרת לתאר מדיה במקום SDP.

במאמר זה, כאשר אני מדבר על VoIP, אני בהכרח מדבר על שימוש בפרוטוקולים SIP, SDP ו-RTP, אך עולם ה-VoIP עצמו מכיל פרוטוקולים נוספים הקשורים לאיתות אשר "מתחרים" ב-SIP, וכאמור WebRTC מספק "תחליף" ל SDP.

בעיות בעולם ה-VoIP

בניגוד למערכות כדוגמת HTTP, עם VoIP ישנם מספר בעיות מאוד קשות אשר חייבים לפתור. הבעיה העיקרית היא שVoIP מאוד פגיע להתקפת Man in The Middle או MiTM בקיצור ובנוסף גם למניעת שירות וגניבת שיחות בצורה ישירה.

הסיבה לכך די פשוטה - הפרוטוקול של SIP מאפשר לבצע פניות ישירות וניתובים שונים על ידי מספר גורמים "רשמיים":

- שימוש ברשומת DNS מסוג SRV אשר יכולה להכיל מספר של שרתים תחתה
- שימוש ב-SBC אשר משמש מעין Firewall לתעבורת VoIP
- שימוש ב-SIP Proxy אשר משמש לניתוב עבור מספר מערכות
- תמיכה בפרוטוקול לעבור ניתובים ונתבים שונים
- חובה לעשות שימוש בשרתים מרוחקים לביצוע NAT Traversal
- המערכת מתקשרת ב-UDP ולכן אין state לתקשורת
- שימוש ב"מכשיר" טלפון מבוסס TCP/IP (בין אם הוא רק תוכנה, או גם מכיל חומרה דמויית מכשיר "רגיל").

למעט הנקודה על UDP אשר בהכרח בעייתית, שאר הנקודות יכולות גם לא להתרחש מבחינת הבעייתיות שלהן, אך עצם כך שהפרוטוקול יודע ומודע לכך שניתן "לווסת" את התעבורה על ידי הנגשה לכתובות חדשות או לגשת מהרבה נקודות, קל מאוד להביא את התעבורה לשרת לא מורשה, או להוציא שיחה לא מורשת והנה יצרנו יחסית בקלות מספר סוגי התקפות שונות.

הצורה בעצם לבצע MiTM ב-SIP, או לנתב בכלל תעבורה היא די פשוטה - יש שדה בשם Route, ויש שדה בשם Max-Forward אשר ביחד יכולים לנתב בקשות ליותר משרת אחד כאשר השדה של Max-Forward מספק רק מידע מה כמות הניתובים המקסימלית לאפשר.

היות ו-SIP מתוכנן בראש ובראשונה לעבוד עם UDP, יש הכרח לממש דברים בתוך הפרוטוקול עצמו, ולא ברמת UDP, ולכן קל מידי להזריק ניתובים חדשים עם התערבות בתוכן של SIP, דבר שקשה יותר לביצוע אם היה מתבצע ברמת סוג ה-socket.

בשביל להוציא שיחה, אני צריך להזדהות כ"שלוחה" מבוססת SIP, ולפעמים אפילו לדעת את הסיסמה בנושא (כמו 12345) עבור challenge.

בנוסף לבעיות MiTM, חשיפה של מערכות VoIP לרשת בלתי מוגנת, גוררת אליה סריקת רשתות והתקפה מאסיביות במטרה לבצע פעולות fraud שונות - בהם גם גניבת שיחות כפי שתארתי בפסקה הקודמת, ואפילו "סתם" מניעת שירות.

בשל בעיות אלו, יש צורך בהצפנת המידע, אשר תהיה נגישה רק למי שמורשה לשלוח את התוכן הרצוי, בתקווה שזה רק היעד שרוצים להגיע אליו. אך הצפנות ללא הגנות נוספות, לעולם לא יגנו מפני כל הבעיות המתוארות כאן, אך השימוש בהצפנה מספק עוד דרך להגן על המידע בצורה משולבת, אשר תאפשר להגן טוב יותר על השיחות ותכניהם.

הבנת גישות הצפנה על רגל אחת

כאשר ניגשים להצפין מידע, צריך להבין כיצד יהיה בו שימוש. למשל בשביל טקסט שנשלח כל הזמן באמצעות זרימה, נוכל להשתמש בפרוטוקול בשם OTR - Off The Record, היות והוא בנוי להצפין את הטקסט בצורה סינכרונית (כלומר עם סיסמה אשר ידועה לשני הצדדים), ולפתוח את חלקי הטקסט השונים לצד המתקבל, גם אם לא מדובר בכל הטקסט עצמו.

ההבדל בין מצב זרימה למצב של מידע "סטטי", הוא בכך שיש צורך לדעת להתמודד עם חלקי מידע ולפתוח אותם בחלקים קטנים.

כאשר רוצים למשל להצפין קובץ zip, אין הוא מחזיק בתוכו הצפנה של מצב זרימה, ויש צורך לקבל את כל הקובץ עצמו בשביל להצליח ולדעת כיצד לשחזר את המידע המוצפן למידע של פורמט zip.

היות ועולם ה-VoIP בנוי על זרימה, חייבים להשתמש בפרוטוקולים אשר מסוגלים לספק חלקי מידע בצורה מוצפנת, ועדיין להצליח לפתוח אותם בצד השני. לשם כך, משתמשים בהצפנות שהן א-סימטריות. הצפנה א-סימטרית אינה עובדת על סיסמה ידועה מראש, אלא על ידי שימוש בחלק מידע הידוע וזמין לכולם אשר הוא חלק ממידע סודי ומסווג, אשר שני החלקים בייחד יוצרים צורת הצפנה בין שני צדדים, כאשר ההצפנה היא יחודית בין שני הצצדים.

לרוב בהצפנה שכזו, ישתמשו בתעודות הצפנה, ויממשו את פרוטוקול Diffie Hellman.

סוגי הצפנות ל-VoIP

היות ו-VoIP משתמש במספר פרוטוקולים שונים במקביל, כפי שהזכרתי בהקדמה, כל פרוטוקול מקבל הצפנה משל עצמו.

הצפנת איתות

עבור SIP יש מקביל אשר קיבל את השם SIPS. ה-S בסוף מייצג "secure" כפי שב-HTTPS מייצג ה-S מצב "secure". בעוד ש-SIP רגיל לרוב נמצא תחת פורט 5060, לרוב SIPS יהיה תחת פורט 5061.

SIP תומך בהצפנה של SSL, אך היות והיא ידועה כלא מאובטחת, לא אכנס אליה בכלל.

מה שמשאיר לנו את TLS.

TLS הוא פרוטוקול הצפנה א-סימטרי המממש את תפיסת Diffie-Hellman, ו"מתלבש" מעל רמת ה socket ב-TCP/IP. הסיבה לשימוש בו היא בגלל שההצפנה נועדה לתקשורת מחשבים, ולוקחת את התוכן שנמצא על גבי פרוטוקלים UDP ו TCP בשכבה 5 (לפי מודל 7 השכבות) שהיא שכבת ה"שיחה", ומצפינה אותם. כלומר זה מצב בו מחליטים מה קורה בתוך הבקשה, לאחר שהוחלט כי היא למשל UDP.

הסיבה להצפנת פרוטוקול האותות, היא בעצם לאפשר שליטה טובה יותר למי מותר לבצע פעולות, אך חשוב מזה, להצליח להצפין את המדיה, בלי להדליף מידע בנושא. אם לא תתבצע הצפנה ל SIP שנושא בתוכו SDP, אשר יתאר מדיה מוצפנת, יהיה ניתן לקבל מידע אודות ההצפנה של המדיה, ובכך ניתן להאזין ל RTP גם כאשר הוא מוצפן.

SDP "רגיל" יכול להראות כך:

```
v=0
o=- 25678 753849 IN IP4 10.0.0.4
s=
c=IN IP4 10.0.0.4
t=0 0
m=audio 3456 RTP/AVP 18 96
a=rtpmap:96 telephone-event
a=fmtp:96 0-15,32-35
a=sqn: 0
a=cdsc: 1 audio RTP/AVP 0 18 96
a=cpar: a=fmtp:96 0-16,32-35
a=cdsc: 4 image udptl t38
a=cdsc: 5 image tcp t38
```

אך SDP המתאר גם הצפנה יכול להראות כך (לפי RFC4568):

```
v=0
o=jdoe 2890844526 2890842807 IN IP4 10.47.16.5
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.example.com/seminars/sdp.pdf
e=j.doe@example.com (Jane Doe)
c=IN IP4 161.44.17.12/127
t=2873397496 2873404696
m=video 51372 RTP/SAVP 31
a=crypto:1 AES_CM_128_HMAC_SHA1_80    \
    inline:d0RmdmcmVCspeEc3QGZiNWpVLFJhQX1cfHAwJSoj|2^20|1:32
m=audio 49170 RTP/SAVP 0
a=crypto:1 AES_CM_128_HMAC_SHA1_32    \
    inline:NzB4d1BINUAvLEw6UzF3WSJ+PSdFcGdUJShpX1Zj|2^20|1:32
m=application 32416 udp wb
a=orient:portrait
```

כפי שניתן לראות בקוד המודגש, יש תאור של צורת החתימות וניתן לדעת כיצד להאזין לאודיו מול מה שנבחר. יותרה מזאת, ניתן לדעת גם מה סוג המידע שיהיה ב-RTP באמצעות SAVP.

AVP הם ראשי תיבות ל Audio Visual Profile ולמעשה SAVP כפי שניתן לנחש מדבר על Secure Audio Visual Profile. יש גם עוד גישה ל-AVP אשר נקראת Profile with Feedback Audio Visual או AVPF בקיצור, אשר גם היא מכילה גרסה מאובטחת בשם SAVPF.

בנוסף ל-RTP שהסברתי בקצרה, גם RTCP מושפע כאן ישירות. וזו למעשה הסיבה שבמידה ורוצים להצפין את המדיה, יש צורך גם להצפין גם את מערכת האותות שבאה איתה.

סוגי הצפנות למדיה

ישנם כ-שתי תצורות להצפנות נפוצות עבור המדיה עצמה:

1. SRTP
2. zRTP

SRTP

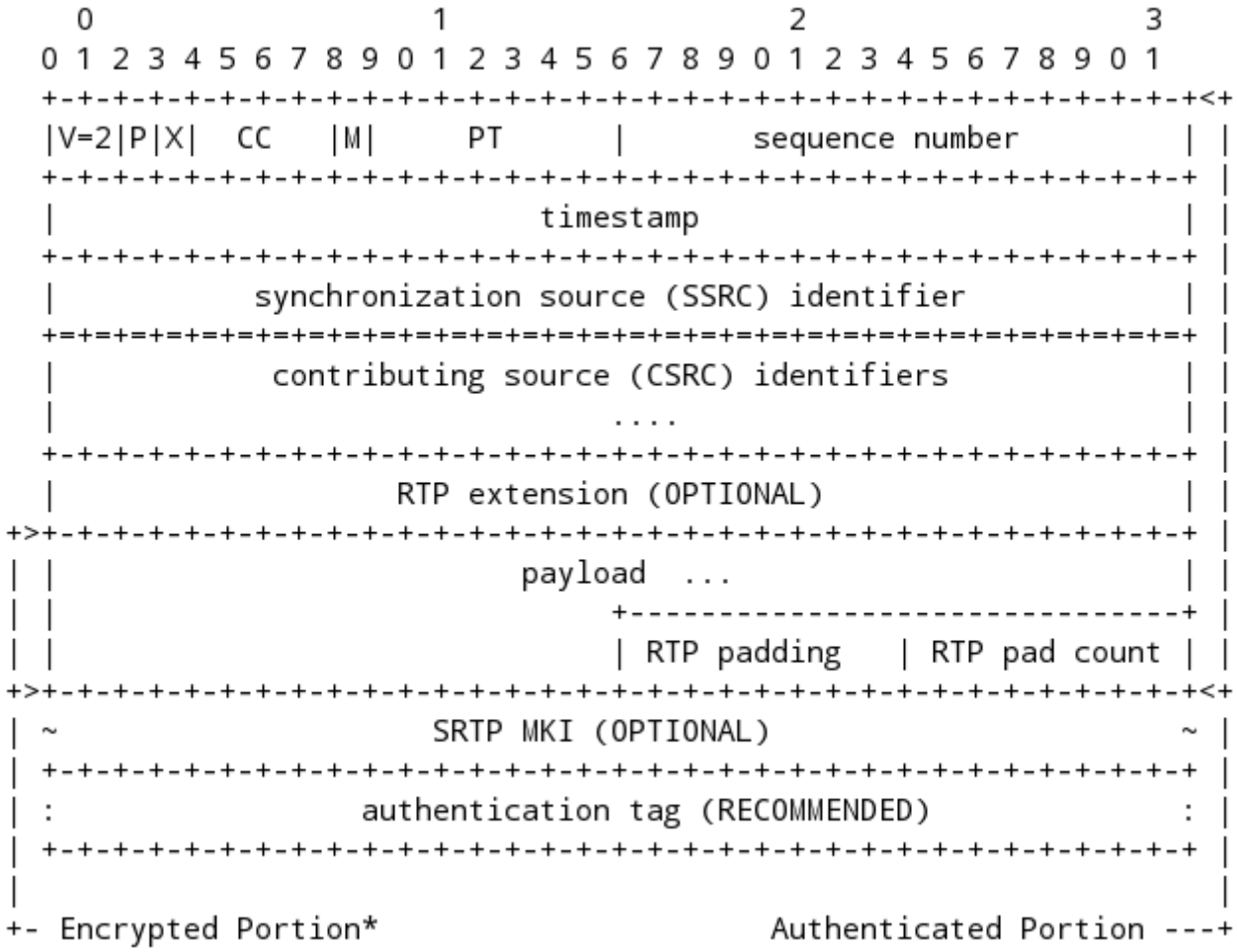
זוהי שיטה נפוצה מאוד אשר מכילה את ראשי התיבות של Secure Real-Time Protocol. הרעיון הוא לבצע הגנה לשידור המדיה במצבים של unicast - כלומר שידור לצד אחד, או multicast - שידור למספר גורמים ידועים במקביל (כדוגמת שיחת ועידה). הפרוטוקול נוצר על ידי חברות סיסקו ואריקסון בשביל להצפין את המדיה העוברת בניהם.
הרעיון של SRTP הוא הצפנה בשני חלקים:

1. הצפנה של הפקטה עצמה
2. הצפנה של חלקי הזרימה

הצפנת הפקטה עצמה היא סוג של Block cipher, אשר למעשה לוקחת את כל הפקטה ומצפינה אותה כמו שהיא. הסיבה לכך היא שפקטה תמיד מגיעה באותו הגודל, והיא תמיד מגיעה או לא מגיעה בצורה אטומית, ובכך כל המידע מגיע ועליו ניתן להפעיל ולפתוח את ההצפנה.

הצפנת חלקי זרימה, הוא הצפנת המידע (payload) עצמו.

לפי RFC3711, מבנה של SRTP נראה כך:



ה-SDP שמתואר ב-RFC4568 אשר הדבקתי בחלק של הצפנות VoIP למעשה מתאר שימוש ב-SRTP. מבנה הפקטה מתואר בתרשים כאן למעלה, וניתן לראות כי הצפנת ה-block מתוארת בצד ימין, והצפנת המידע (payload) עצמו מתוארת בצד שמאל.

בנוסף ל-SRTP, גם RTCP מכיל פרוטוקול הצפנה אשר קיבל את השם SRTCP. הבעיה היא שלא חובה לבחור בשימוש של SRTP ושל SRTCP. גם אם אחד מהפרוטוקולים נכנס להצפנה, זה לא אומר שגם הפרוטוקול השני מוצפן.

הפרוטוקול אפילו מאפשר לשים הצפנה בשם null cipher, שהיא למעשה רק מתנהגת כאילו יש הצפנה, אך המידע אינו באמת מוצפן. כאשר רוצים מידע מוצפן כמו שצריך, חייבים לקחת בחשבון את כל הנקודות האלו ולדאוג להן לפני שניתן לדעת שיש הצפנה בפועל של המדיה.

zRTP

זוהי גישה מבוססת על Diffie-Hellman, בה כאמור יש החלפת מפתחות הצפנה ושימוש ב-SRTP להצפנה עצמה ופותחה על ידי IETF. האות z מייצגת את אחד המתכננים העיקריים של הפרוטוקול בשם פיל צימרמן (Phil Zimmermann).

zRTP עובד רק במצב של אחד לאחד (כלומר unicast), ואינו מתוכנן לעבוד עם מצב של שיחות בין קבוצות. אך המימוש חייב לתמוך ברעיון של multiplexing - רעיון בו ניתן להעביר יותר ממידע בודד באותו הצינור בדיוק, אותו רעיון של "קו טלפון" מסורתי, אשר כבל אחד יכול "להחזיק" מספר שיחות במקביל.

הרעיון של zRTP הוא שאין אמת צורך "לתאר" אותו בזמן איתות, אך כן ניתן להכריז עליו בSDP עם שדה attribute כלשהו ובכך לנסות ולהציג את יכולת התמיכה כבר בשלב האיתות, אך הוא אינו חייב לבצע זאת ברמת ה-SDP.

הסיבה לכך היא בגלל שהפרוטוקול מנסה לבדוק האם הצד השני מסוגל לענות בפרוטוקול, ובמידה וכן, מצפין את המידע, ובמידה ולא, המידע עובר בצורה לא מוצפנת. גישה זו קיבלה את השם "הצפנה מזדמנת" (Opportunistic encryption).

אחת הדכים ש-zRTP מבצע זאת, היא על ידי ביצוע סוג של ping, אשר כל מי שתומך ב-zRTP חייב לענות עליו. במידה ואין מענה, אז הוא יודע כי לא ניתן "להיכנס" למצב הצפנה.

למרות השימוש ב-Diffie-Hellman, אין צורך בהכרזה מוקדמת על מפתח ההצפנה בפרוטוקול, היות והמפתחות נוצרים בזמן ריצה בין שני הצדדים. גישה זו מאפשרת להגן מפני MiTM במידה והמחשב שקיבל את הבקשה הראשונה אינו שייך לתוקף, ובכך קשה מאוד "לזהם" את ההצפנה כאשר זו התרחשה בין שני צדדים.

הניסיון למנוע התקפת MiTM מתבצע על ידי שימוש ב-SAS, אשר גורמת לשני הצדדים לחשוב על טקסט או חתימה דיגיטלית קצרים שהם משמשים כזיהוי הראשוני, ובמידה ואותו הדבר קיים אצל שני הצדדים, אז בעצם נכנס ה-PKI לתמונה וכך נמנע MiTM וההצפנה מתחילה עם תעודת הצפנה.

למעשה שדה ה-attribute תחת ה-SDP אמור להכיל את הודעת ה-SAS, אך אין זה חייב להתרחש שם (לפי RFC6189):

```
v=0
o=bob 2890844527 2890844527 IN IP4 client.biloxi.example.com
s=
c=IN IP4 client.biloxi.example.com
t=0 0
m=audio 3456 RTP/AVP 97 33
a=rtpmap:97 iLBC/8000
a=rtpmap:33 no-op/8000
a=zrtp-hash:1.10 fe30efd02423cb054e50efd0248742ac7a52c8f91bc2 \
   df881ae642c371ba46df
```

בקשות zRTP מכילות פקודות שונות ולמעשה מכילות תת פרוטוקול תקשורת, כאשר כל תת פקודה שכזו מתועדת ב-RFC ומכילה מבנה משל עצמה.

לכן הפרוטוקול נחשב ל"בזבזני" יותר, אך מצד שני מספק יכולות רבות יותר, והגנה טובה יותר, והצפנת המדיה עצמה עדיין מתבצע עם SRTP, אשר zRTP מתערב בו.

WebRTC

כפי שציינתי בתחילת המאמר, ישנו פרוטוקול חדש יחסית בשם WebRTC. הוא כאמור מכיל SDP, אך גם מחייב להצפין את כל המידע העובר דרכו. WebRTC מכיל עולם שלם של דברים, אשר נכונים למימושים שלו, ואכנס כאן רק לחלק ההצפנה בפועל שנדרש.

ב-WebRTC צריך להצפין את היכולת לקבל אותות, אשר אינה ממומשת על ידי הפרוטוקול עצמו, אלא ניתנת לבחירה בזמן השימוש, וזה נעשה באמצעות WSS שהם ראשי תיבות של Web Sockets Secure.

אחד הרעיונות המקוריים היו גם כיצד לתאר את ההצפנות תחת ה SDP, ולשם כך הציעו את - SDES Session Description Protocol Security Description, אשר מסביר כיצד ה-SDP מציג הצפנה. אך ההצעה ירדה מהפרק (נכון לכתיבת מאמר זה), והמימוש היחיד של SDES שאני מכיר בעולם של WebRTC הוא של חברת גוגל.

עבור WebRTC יש שימוש בתת גרסה של SRTP בשם DTLS-SRTP. ראשי התיבות של DTLS הם Datagram Transport Layer Security. התפקיד שלו הוא לאפשר הגנה על UDP באמצעות TLS, ועליו יש מימוש של SRTP ו-SRTCP כפי ש-RFC 5763 ו-RFC 4764 מתארים. הדגמה של SIP עם SDP עבור DTLS-SRTP נמצאת ב-RFC5763.

אליס שולחת בקשה לבוב:

```
INVITE sip:bob@example.com SIP/2.0
To: <sip:bob@example.com>
From: "Alice"<sip:alice@example.com>;tag=843c7b0b
Via: SIP/2.0/TLS ua1.example.com;branch=z9hG4bK-0e53sadfkasldkfj
Contact: <sip:alice@ua1.example.com>
Call-ID: 6076913b1c39c212@REVMTEpG
CSeq: 1 INVITE
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, UPDATE
Max-Forwards: 70
Content-Type: application/sdp
Content-Length: xxxx
Supported: from-change
v=0
o=- 1181923068 1181923196 IN IP4 ua1.example.com
s=example1
c=IN IP4 ua1.example.com
a=setup:actpass
a=fingerprint: SHA-1 \
  4A:AD:B9:B1:3F:82:18:3B:54:02:12:DF:3E:5D:49:6B:19:E5:7C:AB
t=0 0
m=audio 6056 RTP/AVP 0
a=sendrecv
a=tcap:1 UDP/TLS/RTP/SAVP RTP/AVP
a=pcfg:1 t=1
```

החלק המודגש מראה היכן יש הגדרה לבקשה שכזו, כאשר כאן זו הצעה לצד השני להשתמש ב-DTLS SRTP.

בוב יגיב לכך באופן הבא:

```
INVITE sip:bob@ua2.example.com SIP/2.0
To: <sip:bob@example.com>
From: "Alice"<sip:alice@example.com>;tag=843c7b0b
Via: SIP/2.0/TLS proxy.example.com;branch=z9hG4bK-0e53sadfkasldk
Via: SIP/2.0/TLS ua1.example.com;branch=z9hG4bK-0e53sadfkasldkfj
Record-Route: <sip:proxy.example.com;lr>
Contact: <sip:alice@ua1.example.com>
Call-ID: 6076913b1c39c212@REVMTEpG
CSeq: 1 INVITE
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, UPDATE
Max-Forwards: 69
Identity: CyI4+nAkHrH3ntmaxgr01TMxTmtjP7MASwliNRdupRI1vpkXRvZXx1ja9k
          3W+v1PDsy32MaqZi0M5WfEkXxbgTnPYW0jIoK8HMyY1VT7egt0kk4XrKFC
          HYWGCl0nB2sNsM9CG4hq+YJZTMaSROoMUBhikVIjnQ8ykeD6UXNOyfI=
Identity-Info: https://example.com/cert
Content-Type: application/sdp
Content-Length: xxxx
Supported: from-change

v=0
o=- 1181923068 1181923196 IN IP4 ua1.example.com
s=example1
c=IN IP4 ua1.example.com
a=setup:actpass
```

```
a=fingerprint: SHA-1 \
  4A:AD:B9:B1:3F:82:18:3B:54:02:12:DF:3E:5D:49:6B:19:E5:7C:AB
t=0 0
m=audio 6056 RTP/AVP 0
a=sendrecv
a=tcap:1 UDP/TLS/RTP/SAVP RTP/AVP
a=pcfg:1 t=1
```

בחזרה, ניתן לראות למשל בבקשת ה-SIP של בוב כי היה ניתוב ל-2 שרתים עד שהבקשה הגיעה לבוב מאליס. אנו רואים זאת על ידי השדות של Via.

כך שניתן לראות עד כמה קל להיות פגיע לבעיות ניתוב שונות ושוב פעם להבין את החשיבות של הצפנת האותות עוד לפני שמצפינים את המדיה עצמה.

סיכום

במאמר זה סיפקתי ראיה על בעיות שיכולות להיות בעולם ה-VoIP וכיצד שימוש בהצפנה יכול לסייע בנושא. הצגתי מה הם המרכיבים העיקריים של עולם ה-VoIP, כדוגמת איתות באמצעות SIP, תאור המדיה באמצעות SDP, המדיה עצמה עם RTP ושליטה ב-RTP באמצעות RTCP.

הצגתי מה הם הגישות שניתן לקחת עבור הגנה על תקשורת שכזו, והסברתי את השוני בין התפיסות השונות.

למשל שימוש ב-TLS בשביל להצפין את בקשת ה-SIP ובקשת ה-SDP, ובכך למנוע יכולת לחשוף את צורת ההצפנה שתתבצע עבור המדיה. הצגתי שני שיטות עיקריות עבור הצפנת המדיה באמצעות SRTP ו-zRTP. וסיפקתי תאור קצר כיצד SRTP בנוי.

אני מקווה כי עכשיו יהיה קל יותר לתרגם את ההגדרות של המערכות בהם אתם נוגעים כאשר נדרשת הצפנה, לצרכים שאתם מנסים לפתור.