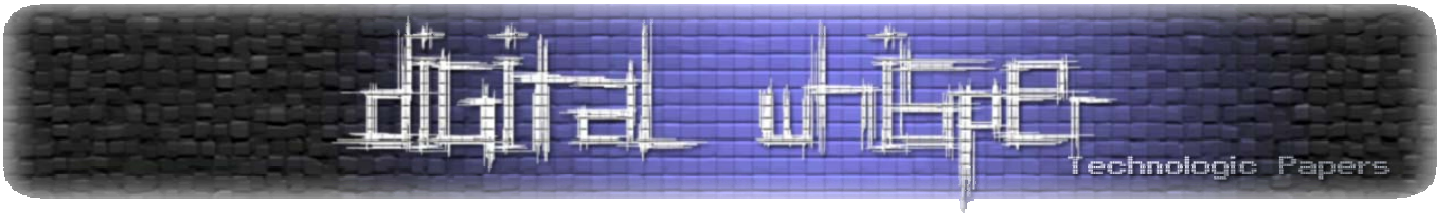


להבין את התכלס מאחורי האנונימיות המורכבת TOR

מאת ליאור ברש

הסיפור שלנו מתחיל בבצל, בצל די ישן, בצל מודל 98' יד ראשונה מרופא, שסי נקי, גיר ידני ומוח מטורף. רציתם אנונימיות, רציתם טכנולוגיה, תקבלו. בנתיים זה מתחיל מבצל.

ב- 98 רשם חיל הים האמריקאי פטנט, על בצל. כמה מוחות מרשימים שאפשר לקרא עליהם בוויקיפדיה פיתחו רעיון שנקרא Onion Routing שהוא בעצם טכניקה לשמירה על אנונימיות ברשת. המטרה היא לשמור בסוד "מי אמר למי", ומתוקף המבנה של המערכת יוצא שגם המידע שעובר בפקטים מוגן, אך מדובר בפועל יוצא של הרעיון המקורי וטכניקת העבודה.

לפי הרעיון, אם נשלח את המידע ישירות ליעד שלו, כולם יודעים מי אמר למי ומתי ולא עלינו גם מה נאמר שם. לעומת זאת, אם נצפין את המידע וגם את היעד בכמה שכבות של הצפנה ובכלל נשלח למישהו אחר, שהוא מבחינתו יכול רק להוריד את שכבת ההצפנה העליונה ולשלוח את החבילה לבא אחריו שיכול לעשות בדיוק את אותה הפעולה אז נצא בסדר ואף אחד לא ידע מה שלחתי ולמי.

אף על פי כן, משהו כאן מסריח וזה לא הבצל, נקודת הכשל הראשונה שעולה היא "מה קורה אחרי שמורידים את שכבת ההצפנה האחרונה והמידע חשוף?"

זו אכן בעיה וזה בסדר כי פתרון זה בכלל לא מתיימר להתמודד איתה אלא מצהיר מראש שעדיף וכדאי להשתמש בפתרון בתוך רשתות סגורות ומוגנות או לחילופין ליישם במקביל פתרונות כמו TLS\SSL וחברים.

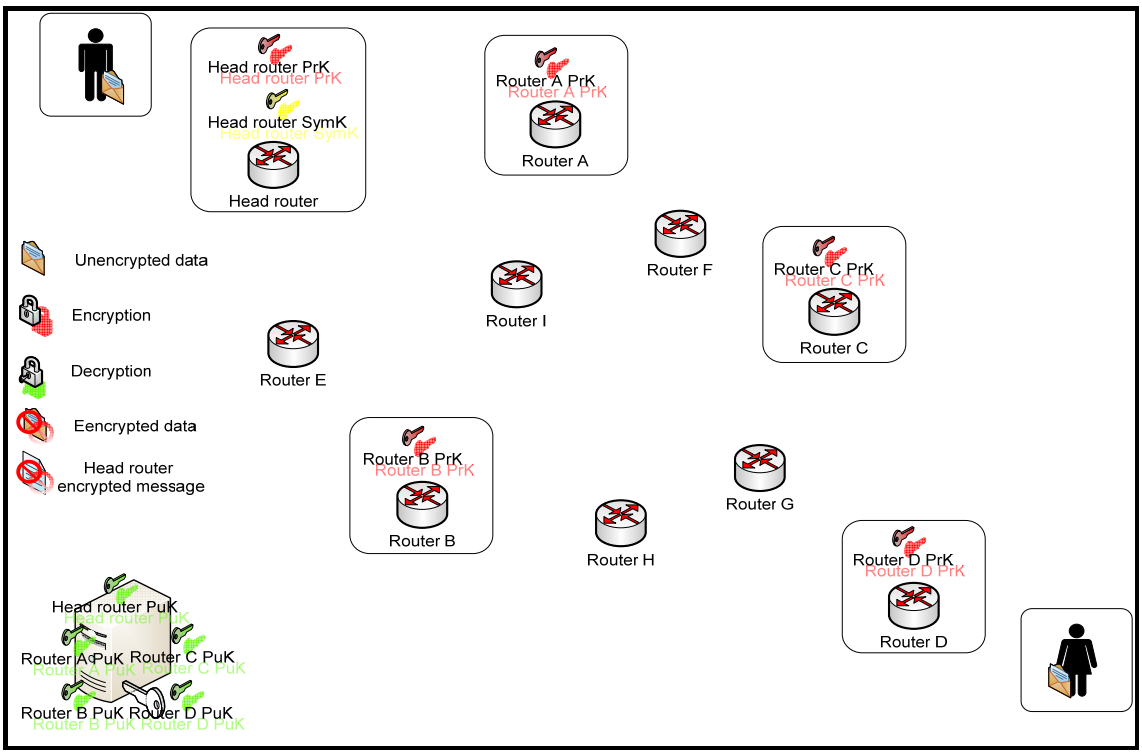
בואו נבין קודם איך עובד המנגנון.

בסכמה הראשונה (סכמה 1.) מתוארת הסביבה בה מתרחש הסיפור שלנו, בצד שמאל למעלה יש "בן" שרוצה לשלוח מכתב ל "בת" שנמצאת בצד ימין למטה. בדרך מבן לבת יש נתבים, בצד שמאל למטה יש שרת מפתחות שמחזיק את המפתחות הציבוריים הזמינים לתרחיש ובין שרת המפתחות ל "בן" יש מקרא מפה.

לכל נתב בדרך יש מפתח פרטי תואם.

מקרא המפה:

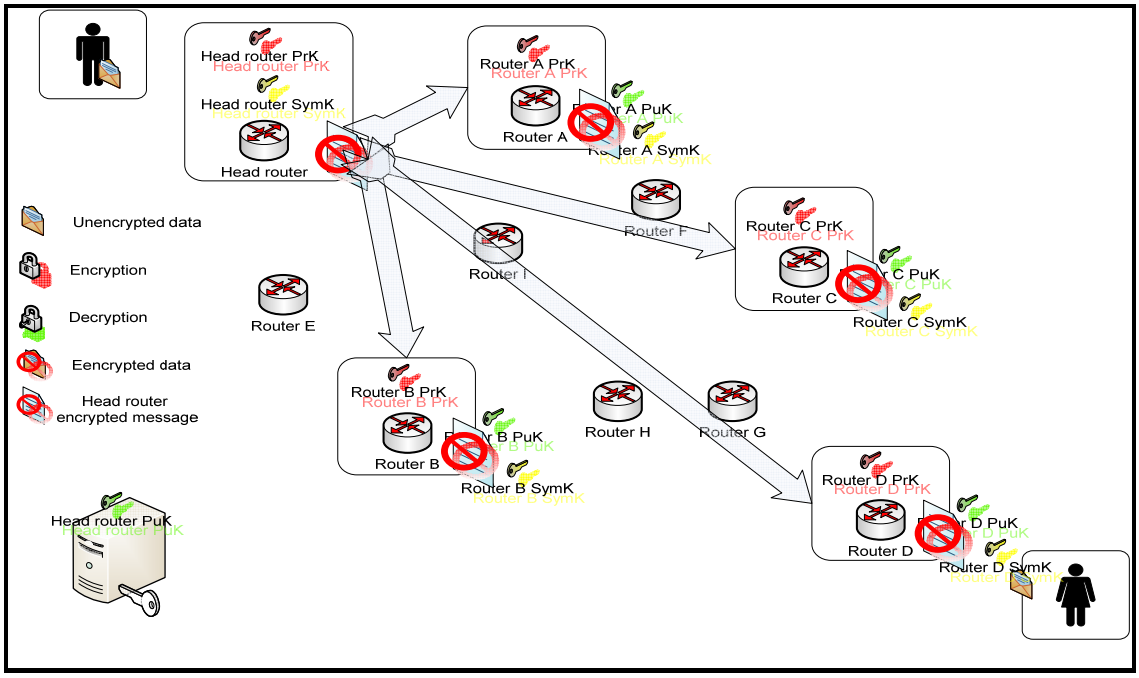
- האייקון של המעטפה מייצג מידע לא מוצפן.
- המנעול האדום בלי המפתח מייצג תהליך של הצפנה.
- המנעול הירוק עם המפתח מייצג תהליך של פענוח.
- המעטפה עם ה- "אין כניסה" מייצגת מידע מוצפן.
- הדף עם ה- "אין כניסה" מייצג את ההודעה המוצפנת שמתחילה את כל החגיגה.



סכמה 1.

התהליך מתחיל (סכמה 2.) כאשר הנתב הראשון בוחר באופן אקראי את הנתבים דרכם ינותב המסלול ושולח לכל אחד מהם הודעה נפרדת המכילה את הפרטים הבאים:

1. מפתח הצפנה סימטרי.
2. מי הנתב הבא בדרך-, יענו: Next hop.



סכמה 2.

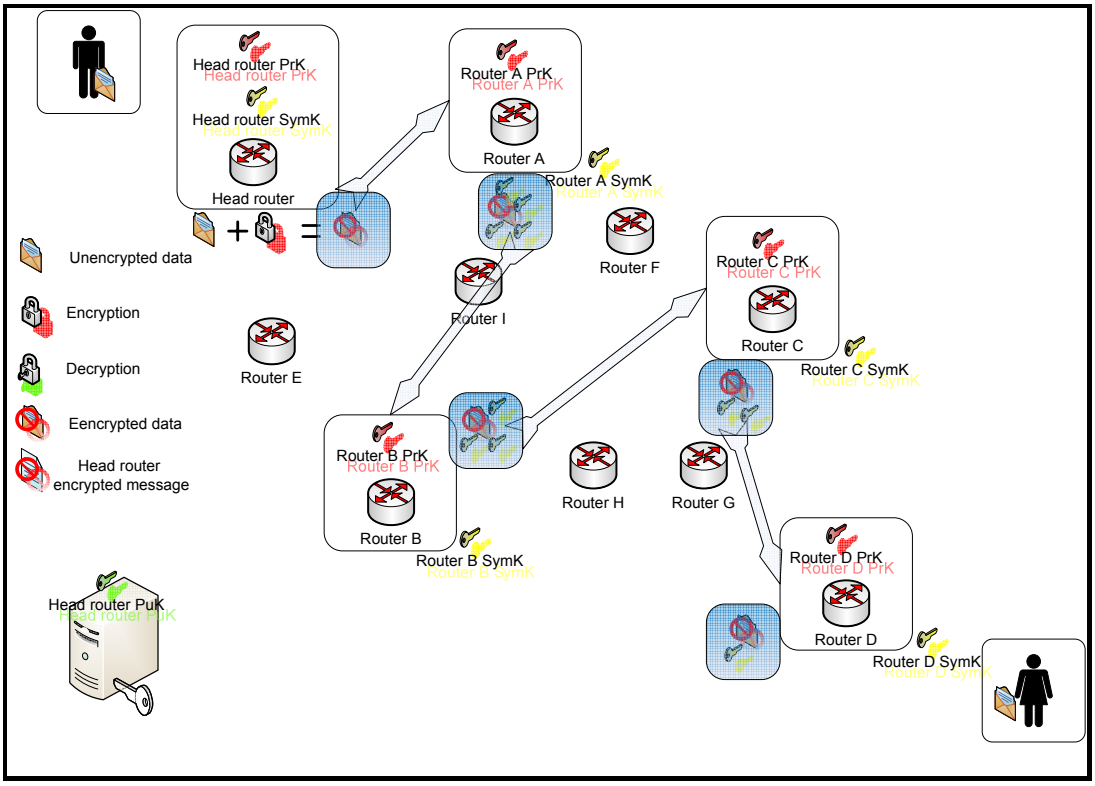
במקרה שלנו, הנתב הראשי בחר את המסלול דרך הנתבים הבאים :

1. Router A
2. Router B
3. Router C
4. Router D

כאמור לכל נתב נשלחה הודעה המכילה את המידע הרלוונטי לגביו (נתב הבא ומפתח סימטרי) כאשר כל הודעה מוצפנת בעזרת המפתח הציבורי של כל נתב בהתאמה.
ההודעה לנתב A הוצפנה עם המפתח הציבורי של נתב A, ההודעה לנתב B הוצפנה עם המפתח הציבורי של נתב B וכן הלאה.
מכאן, "בן" יכול לשלוח את המידע שרצה ל "בת" מבלי שידעו שהוא שלח את ההודעה אליה ו/או מה כתוב שם.

שוב יש בעיות? מריחים בצל? , נכון, אם מישהו השתלט על הנתב הראשי אנחנו בבעיה.

בשלב הבא (סכמה 3.) ההודעה יוצאת לדרך כאשר היא מוצפנת ארבע פעמים, נכון, בדיוק לפי כמות הנתבים שבדרך כאשר כל נתב מוריד שכבת הצפנה אחת בעזרת המפתח הסימטרי שברשותו ומעביר את מה שיצא לנתב הבא שהוגדר לו מראש, הוא מצידו עושה את אותו הדבר וחוזר חלילה עד הנתב האחרון שמפשיט את ההודעה לחלוטין ומעביר את התוכן הלא מוצפן לידיה של "בת".


Head router PrK
Head router SymK
Head router
Router A PrK
Router A
Router A SymK
Router F
Router C PrK
Router C
Router C SymK
Router I
Router E
Router B PrK
Router B
Router B SymK
Router H
Router G
Router D PrK
Router D
Router D SymK
Head router PuK
Unencrypted data
Encryption
Decryption
Eencrypted data
Head router
encrypted message

סכמה 3.

בצל זה טעים אבל לא בשביל זה אנחנו כאן, אז לבנתיים זה מספיק ועכשיו נדבר על TOR.

מערכת TOR (The Onion Routing) לוקחת את העניינים צעד קדימה ופועלת על עיקרון דומה למה שלמדנו עד עכשיו.

ראשית, נגדיר שמטרתה העיקרית של המערכת לאפשר אנונימיות למי שרוצה לצאת לרשת מבלי שידעו מי הוא ולאן הוא הולך ובאותה המידה לאפשר אנונימיות למערכת שרוצה שיגיעו אליה מבלי שידעו היכן היא ממוקמת. לפני שנכנס למעמקי הטכנולוגיה וכיצד היא פועלת חשוב להבין מה הם השימושים האפשריים במערכת, ננסה להגדיר אותם לפי נקודות מבט שונות בחלוקה מגזרית.

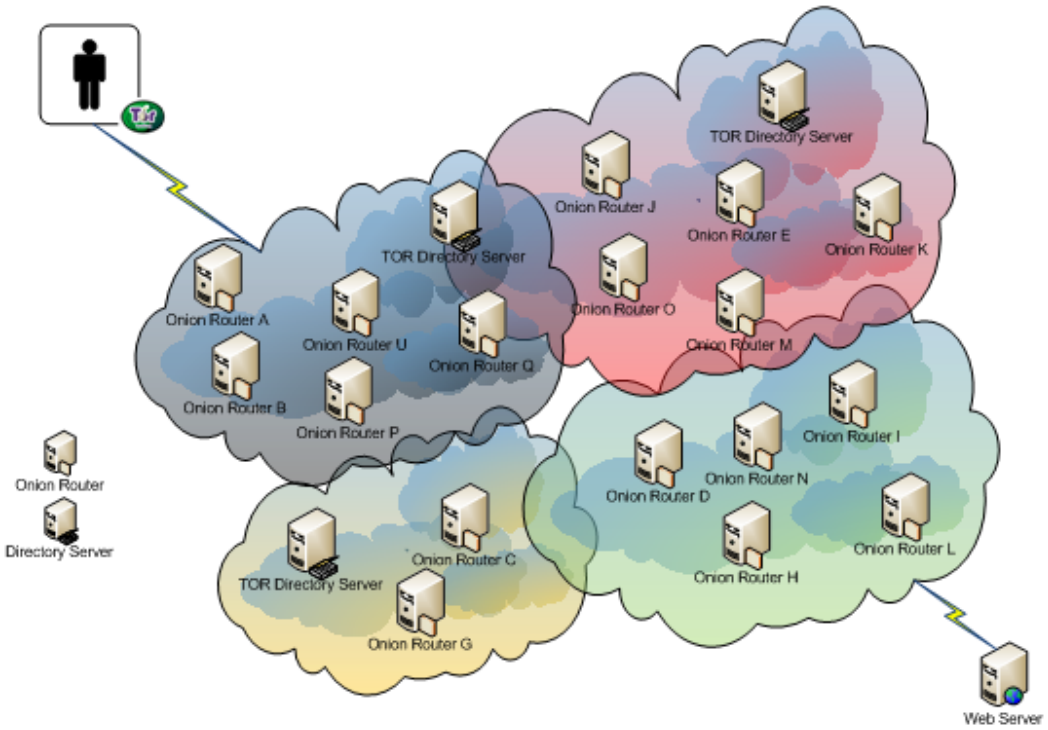
- עיתונאים, בלוגרים וכל מי שמפרסם מידע שיש מי שלא ירצה אותו מפורסם, ישמחו לעשות את עבודתם באופן כזה בו הם יודעים שזהותם ומיקומם הגאוגרפי נשמרים בסוד ובפרט אם הם רוצים להתגבר על מגבלות צנזורה.
- אנשים פרטיים שרוצים לפנות לקבלת מידע באינטרנט מבלי לחסוף את זהותם ולאפשר איסוף מידע לגבי פעילותם ברשת, או לחילופין מעוניינים להחליף מידע בחדרי שיחה או פורומים, למסור מידע לרשויות מבלי להחשף ולהשאר אנונימיים גם ברמת התקשורת.
- ארגונים באשר הם המעוניינים למנוע דליפה של מידע המאפשר ניתוח וריגול עסקי, כמו למשל אילו מחלקות מדברות עם אילו מחלקות, לאילו אתרים משתמשי החברה גולשים ומי הם הספקים איתם עובדים הכי הרבה.
גופים מדיניים המעוניינים להסתיר את הפעילויות המקוונות שלהם כך שלא יתאפשר ניתוח המספק מידע קריטי. הרשת הצבאית אמנם סגורה, אבל מה קורה אם מישהו ינתח פעילות בתוך הרשת, האם נוכל לקבל מידע חיוני לגבי מיקומים וקשרים פנים ארגוניים על ידי ניתוח התעבורה?

שמתם לב שבאתרים רבים שאתם גולשים בהם וממוקמים מחוץ לגבולות המדינה בה אתם חיים או כתובים בשפה שונה, אתם מקבלים פרסומות מותאמות לשפה שלכם, למקצוע שלכם וכן הלאה?

עוד נקודה שחשוב להבהיר היא ש-TOR מתוקף היותו כלי המיועד לשמור על אנונימיות, מבוסס על מתנדבים ואינו מנוהל מרכזית, לא אוהב וטכנית גם קשה לו להתמודד עם העברת כמויות גדולות של מידע. את הסיבה המדוייקת נסביר כשנדבר על המדיניות שניתן להגדיר לשרתי TOR.

מכאן, טכנולוגיה.

סביבת העבודה שלנו (סכמה 4.) כוללת מעט מרכיבים עם הרבה טכנולוגיה שמיושמת ביניהם.



סכמה 4.

בצד שמאל למעלה נמצא שוב "בן" כשעל המחשב שלו מותקן TOR שהוא בעצם פרוקסי קטן, במקרא המפה מתחת ל "בן" ישנם שני שרתים, האחד הוא Onion Router שהוא בעצם שרת המריץ TOR והשני הוא TOR Directory Server.בעננים השונים מפוזרים שרתי TOR ושרתי TOR Directory שכמובן פזורים על גבי רשת האינטרנט באופן חופשי.

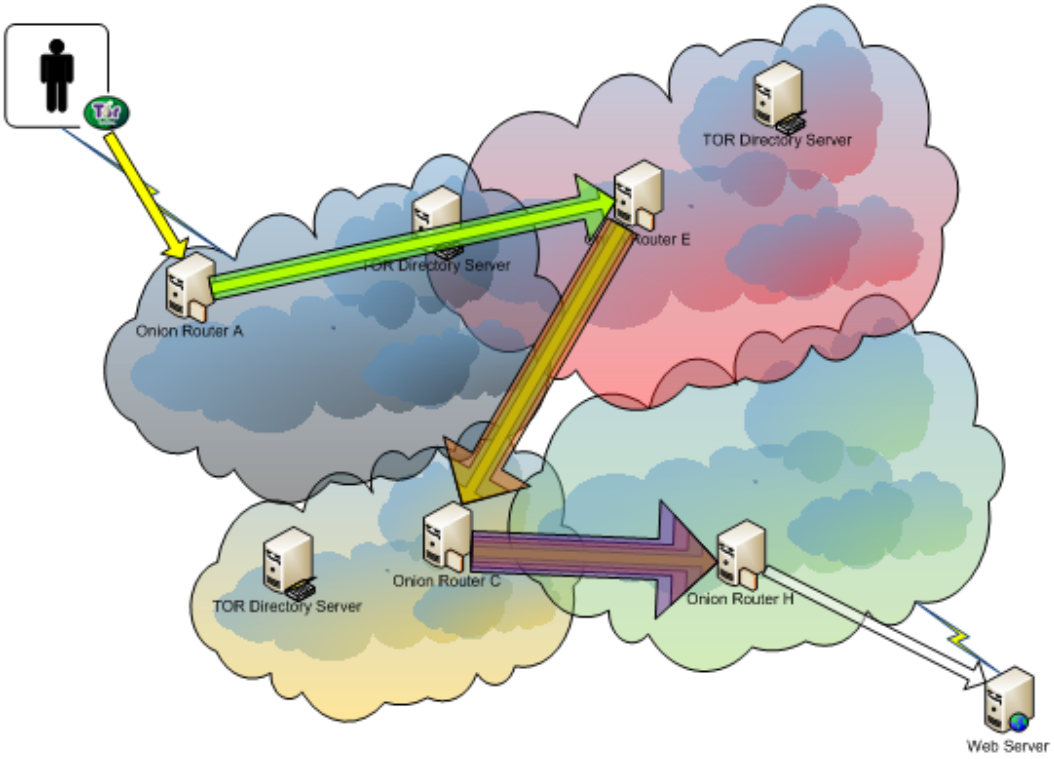
נתחיל מלהבין את אופן הפעולה הבסיסי של המערכת (סכמה 5.) ומשם נרד לפרטים.

"בן" שולח את בקשת החיבור שלו לאתר דרך הפרוקסי של TOR, שנקרא Onion proxy, ולצורך העניין יודע לעבוד עם כל אפליקציה שתומכת ב-SOCKS. הניתוב הקובע דרך אילו Onion Routers תעבור הבקשה, שמורכב מ-Circuits נקבע מראש בצד המשתמש שיודע אילו Onion Routers זמינים, בעזרת שרת ה--TOR Directory שהוא שבעצם Onion Router מוכר ואמין שקיבל את האפשרות להיות שרת כזה, עם תעודה דיגיטלית שתחתום את הנתונים שהוא יודע להעביר כדי שנהיה שמחים ורגועים.

דבר חשוב נוסף שכדאי לדעת הוא ש-TOR משתמש ב Telescoping circuits, מה שאומר ש "בן" מכיר את כל הדרך ובעצם מתבסס על מבנה שנקרא Leaky-pipe circuit topology, המאפשר לו לקבוע מאיזה Circuit המידע יוצא ליעד הסופי. מבנה זה עוזר בהתמודדות עם התקפות שונות המבוססות על Traffic observation.

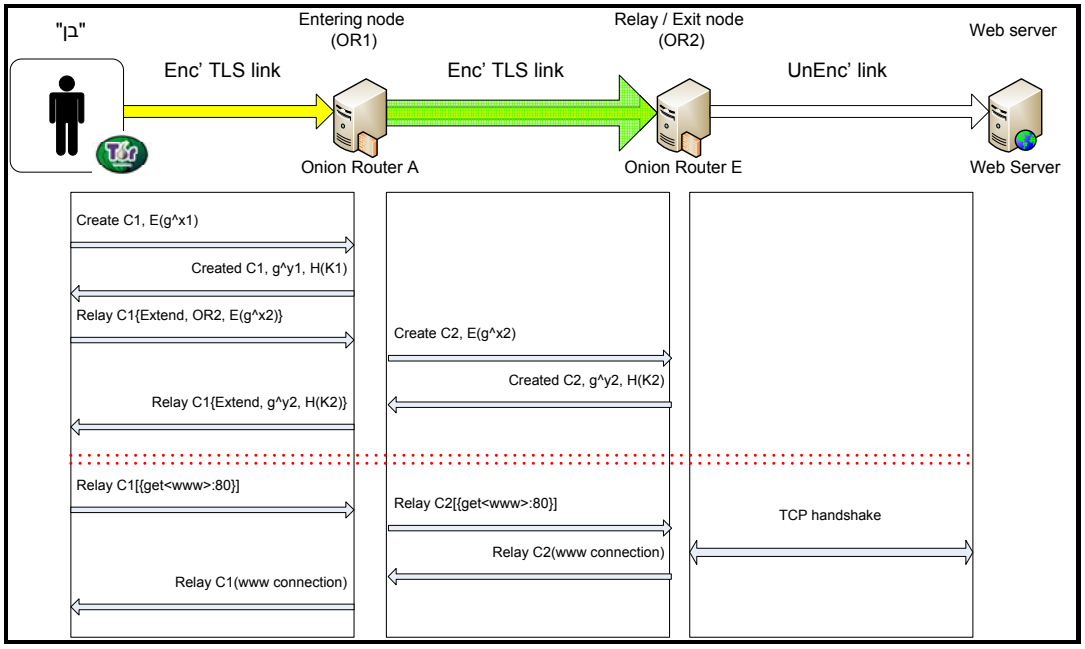
הסשן שנפתח מול ה-Onion Routers מבוסס TLS מטעמי יעילות ואבטחה. עבודה עם הצפנה א-סימטרית לאורך כל ההתנהלות היא לא ריאלית מהיבט של ביצועים, ובעייתית מאוד ברמת תיכנות היישום בפועל. "בן" מתחיל סשן TLS מול ה-Onion router הראשון בדרך שנקרא גם Entering node, לאחר מכן "בן" עושה relay דרך ה-Entering node לכיוון ה-Onion router הבא איתו יצור סשן TLS. במידה וה-Onion router הנוכחי ממשיך ומשרת את "בן" כדי לעשות Relay לעוד Onion Router הוא יקרא Relay node וכך ימשך התהליך עד ה-Onion router האחרון, שנקרא Exit node ותפקידו להעביר את המידע ליעד הסופי. בסופו של דבר "בן" יוצר n סשנים של TLS כאשר n מייצג את מספר ה Onion routers דרכם הוא עובר. שימו לב שבמתודולוגיה הזו כל Onion router מכיר רק את שכניו הקרובים ולא את כל ה-Circuits.

כל ה- Circuits יחדיו מוגבלים באופן אוטומטי למשך זמן של 10 דקות, שלאחריו הם משתנים. TOR מבצע גם End-to-end integrity checking בתוך סביבתו, כמה יופי יכול להיות בפתרון אחד?.



סכמה 5.

ובתצורה הסכמטית:



סכמה 6.

TOR מאפשר לקבוע מדיניות עבודה לכל Onion router שתגדיר מה כמות המידע שתעבור דרכו, רוחב הפס שהוא מאפשר, הגבלות ברמת כתובות IP וכן הגבלות ברמת פורטים. כך למשל, אם נגדיר מדיניות שבה אנו מאפשרים את כל כתובות ה IP בכל הפורטים, ה-Onion router שלנו ישמש את כל התפקידים האפשריים, Entering node; Relay node; Exit nod.
לחילופין אם נגדיר מדיניות שלא מאפשרת אף IP באף פורט, ה-Onion router שלנו יוכל לשמש רק כ-Entering node; Relay node ולא ישמש כ-Exit node המעביר את התעבורה ליעד הסופי. כל המידע שמופיע ב- Description חתום בעזרת המפתח הפרטי של אותו ה- Onion router.

חשוב להבין כי כל התקשורת בין ה-Onion routers לבין עצמם ובינם לבין ה-TOR directory מבוססת גם היא על TLS. רגע לפני שנעבור לכיצד מתבצע התהליך של פרסום שרתים מוסתרים נסכם כמה נקודות

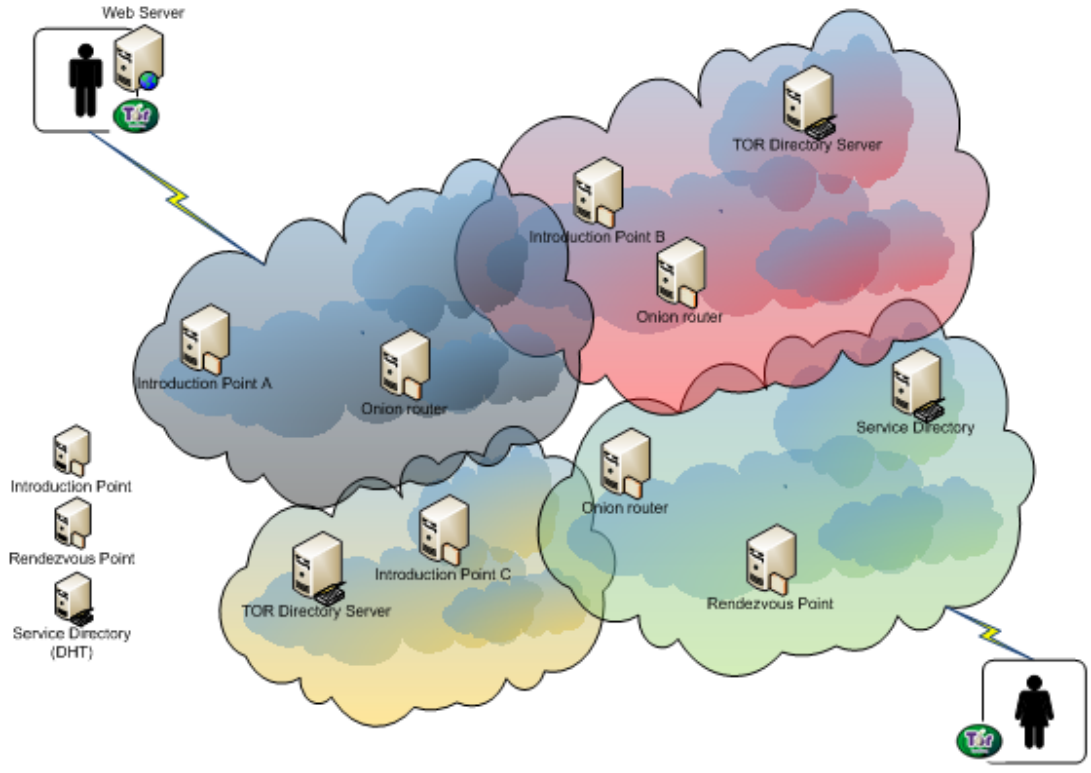
TOR מספק:

- Forward secrecy
- End-to-end integrity check
- Multi TCP streams per circuit
- Leaky-pipe topology
- Distributed authority
- Directory server
- Inter-TORnetwork TLS

- Decentralized congestion control
- Protocol cleaning via SOCKS support

בחלק האחרון של סקירה זו נבין איך אפשר לפרסם שרות מוסתר. תחת הנושא הזה TOR מגדיר מספר מושגים עיקריים, וביניהם Hidden service (שהוא אותו שרות מוסתר אותו אנו רוצים לפרסם), Rendezvous points (שכשמן כן הן- נקודות מפגש) ו- Introduction point ו Service directory- מונח אותו נסביר בקרוב.

סביבת העבודה בתרחיש הזה (סכמה 7.) מושתת כמובן על אותה המערכת, אולם בשל הכיוון ההפוך והדרישות שעולות מכך התהליך מעט שונה. אם הגענו למצב שבו אנו רוצים לפרסם שרות מבלי לחשוף את הכתובת שלו, בוודאי נרצה לוודא כי בידנו האפשרות לבצע Access control filtering, כך שלא כל אחד יוכל להגיע לשרות. בנוסף, נרצה להסדיר כי גם לאחר שהגיעו אלינו נוכל לקבוע מדיניות עבודה ולהבטיח זמינות, בפרט כאשר מדובר ברשת שאין לה "אמא ואבא" וכולה מושתת על מתנדבים. כמו כן, נרצה גם לוודא כי אף על פי שהגישה לשרות שלנו מחייבת עבודה עם TOR, המשתמש שיגיע אלינו יוכל להמשיך לעבוד עם הכלים הסטנדרטיים איתם הוא עובד או עם הדפדפן הרגיל שלו. זכרו כי בנוסף לכל קביעות אלו, הדבר החשוב מכל הוא להבטיח כי תוקף לא יוכל להציג נקודת גישה משלו לאתר שלנו, או בשפה המקצועית phishing.



סכמה 7.

בצד שמאל למעלה שוב נמצא ידידנו "בן" שמפרסם שרות Web מוסתר ומצד ימין למטה נמצאת "בת" המעוניינת להגיע אל שרות ה Web שלנו. את השרות אנחנו יכולים לפרסם דרך פורטל סגור באתר האינטרנט הראשי שלנו, או פשוט ליידע בצורה כזו או אחרת את "בת" על המצאותו של השרות.

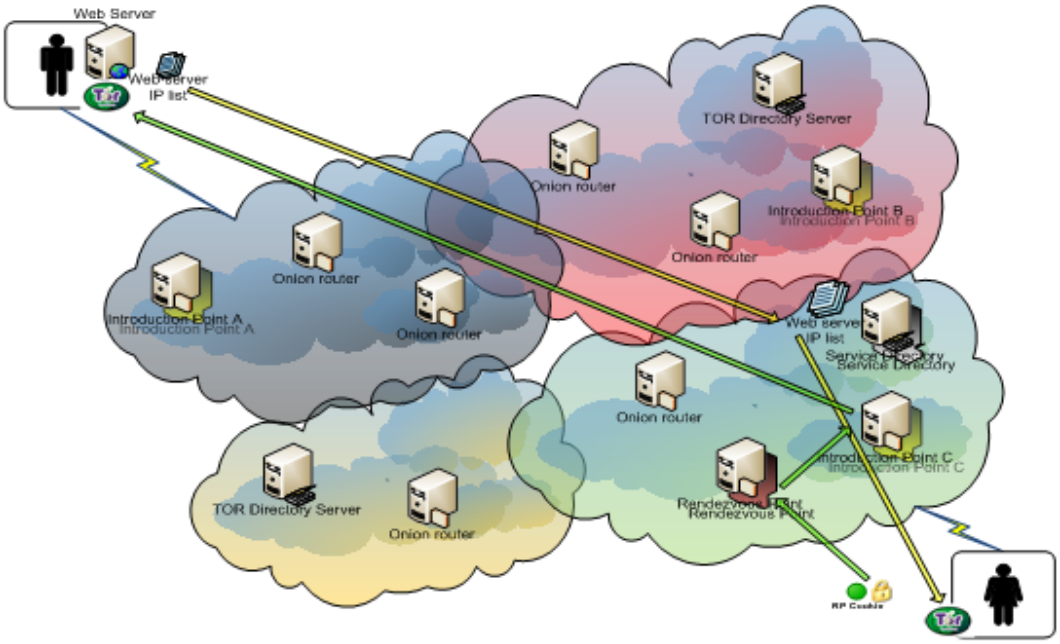
במקרא המפה מתחת ל "בן" מופיעים לפי הסדר:

- Introduction point- נקודת הקישור בה מפורסם השירות, הזהות הבדויה.
- Rendezvous point -Onion router שהוגדר על ידי "בת" כזהות הבדויה שלה.
- Service Directory (DHT)- שרות מבוזר המספק שירותי Lookup.

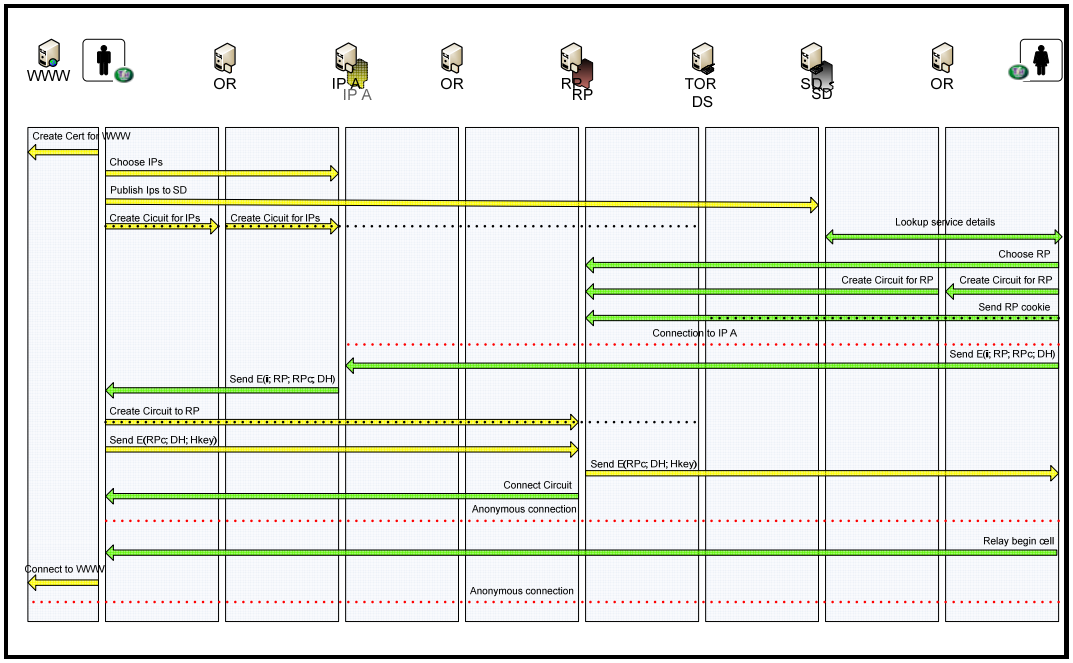
התהליך הכללי (סכמה 8) מתחיל כאשר כאשר "בן" מייצר Public key pair, שישמש כמזהה של השרות אותו הוא מעוניין לפרסם. לאחר מכן, הוא בוחר את ה-Introduction points דרכן יפרסם את השרות ושולח את רשימת ה- Introduction points אל ה-Service directory. לבסוף, מגדיר "בן" Circuits לכל ה-Introduction points שבחר, ומורה להן להמתין לפניות.

עכשיו תורה של "בת" לקבוע Rendezvous point, שהיא בעצם Onion router שנבחר על ידה, ודרכו היא תיגש לשרות. בנוסף, מייצרת "בת" Rendezvous point cookie, אותו היא שולחת אל ה-Rendezvous point שבחרה כדי שיכיר בשרות של "בן" אליו היא רוצה לפנות.
לאחר שבחרה ב-Circuits אל ה-Rendezvous point שלה, היא פותחת סשן אנונימי אל אחד מה-Introduction points שברשותה ומעבירה לבן הודעה שמכילה את הבקשה לחיבור, את זהותו של ה-Rendezvous point שלה, Rendezvous point cookie והתחלה של DH handshake. כל המידע הוצפן בעזרת המפתח הציבורי שהונפק לשרות של "בן".

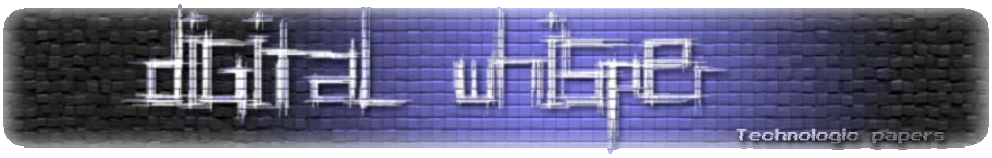
לאחר ש"בן" מקבל את ההודעה, במידה והוא מעוניין לאפשר ל"בת" להתחבר, הוא יוצר Circuit אל ה-Rendezvous point של "בת" ושולח אליה תגובה המכילה את ה-Rendezvous point cookie, את תגובתו ל DH handshake וכן Hash של ה-Session key שיצרו זה עתה.
בשלב זה ה-Rendezvous point של "בת" מחבר אותה ל"בן". הדבר האחרון שנותר לעשות הוא ש"בת" תשלח Relay begin cell אל ה-Onion proxy של "בן", שיחבר אותה ישירות לשרות שלו.



סכמה 8.



ואיך אפשר לסיים בלי ציור זמן (סכמה 9.) מפורט?, אז בבקשה.



מידע נוסף

1. האתר של TOR.

2. מומלץ להציץ על Vidalia, שהיא מערכת גרפית המאפשרת לדעת:
- אם ה- OP שלכם פעיל, להפעיל או לעצור אותו.
- להגדיר את ה- relayים שלכם.
- להביט על פריסת הרשת של TOR.
- להחליף זהות.
- לראות גרף של ניצול רוחב פס.
- להציץ בלוג.
- לגשת למאפיינים של המערכת.
- Chord הוא אתר מעניין שכדאי להכיר. באתר תוכלו למצוא מידע על CFS, שהיא מערכת מבוססת DHT (Distributed Hash Table).

שאלה אחרונה למחשבה, האם יש משמעות להיכן מתבצעת שאילתת ה- DNS של המשתמש?