תשובה 2:

הפרוטוקול תקשורת שרת - לקוח שמוצע מאפשר ללקוחות להעביר קבצים באופן מוצפן מאצלם אל השרת.

תהליך התקשורת ע"פ הפרוטוקול המוצע:

1. הלקוח יוצר קשר עם השרת, שולח בקשה לרישום.

2. השרת בתגובה שומר את נתוני הלקוח לData Base ומייצר עבור הלקוח מזהה יחודי, לאחר מכן שולח הודעת אישור הכוללת את המזהה היחודי.

3. הלקוח שולח מפתח ציבורי לשרת.

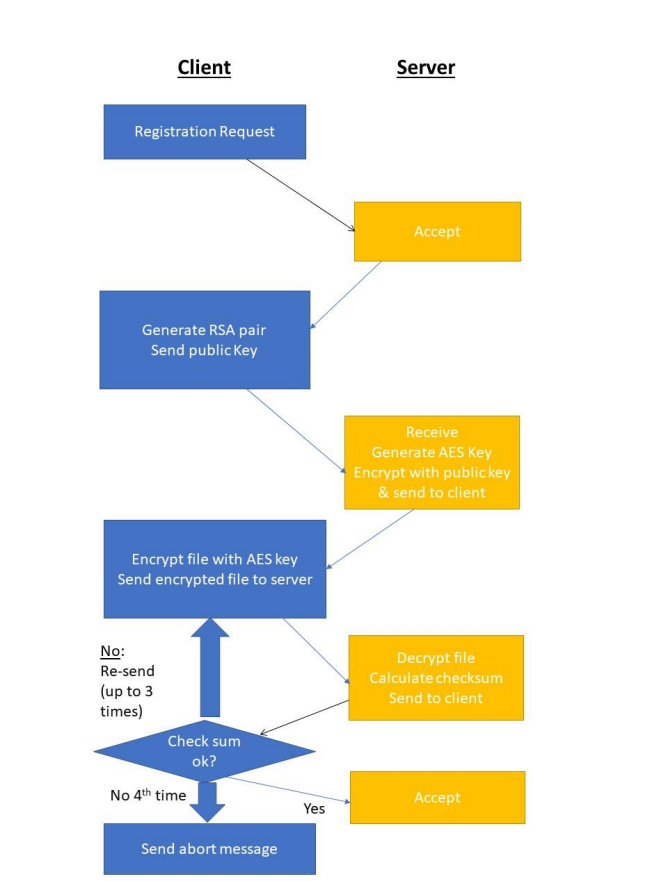
4. השרת יוצר מפתח AES, מצפין אותו בעזרת המפתח הציבורי שקיבל מהלקוח ושולח ללקוח.

5. הלקוח שולח קובץ מוצפן ע"י מפתח הAES שקיבל. ומחשב Check sum לקובץ (הלא מוצפן).

6. השרת מפענח את הקובץ ומחשב Check sum

7. הלקוח בודק האם תוצעות חישוב ה Check sum זהים אצלו ואצל השרת. ושולח עדכון לשרת

8. אם הCRC תקין - השרת שומר את הקובץ ושולח אישור ללקוח שהקובץ נקלט בהצלחה.

9.אם הCRC לא תקין – הלקוח מנסה לשלוח שוב את הקובץ (עד 3 פעמים).

חולשות אופציונאליות בפרוטוקול, והתקפות אפשריות:

\* הפרוטוקול לא מציע שלב וידוא שהשרת אינו שרת זדוני/ שהשרת הינו השרת הרצוי ולא גורם זדוני שהתערב בתקשורת. כלומר – כאשר הלקוח שולח לשרת את המפתח שלו, יכולה להיות מתקפה של **אדם בתווך (Man In The Middle)** – שבה גורם זדוני מתערב בתקשורת, מתמקם על תווך התקשורת שבין הלקוח לשרת ומשפיע עליה, קולט את המפתח מהלקוח ומחליף אותו במפתח ציבורי שהוא עצמו יצר, ואחכ יוכל לפענח את הקובץ המוצפן שהלקוח שולח.

כלומר – הגורם הזדוני יעביר את ההודעות של הלקוח לשרת באופן זהה, אך כאשר הלקוח שולח את המפתח הציבורי – הגורם הזדוני יחליף אותו במפתח ציבורי שלו, מזויף, ואז כשיקבל את התגובה מהשרת הוא יפענח את המפתח AES ע"י המפתח הפרטי המזויף, ויצפין את מפתח הAES מהשרת בעזרת המפתח הציבורי המקורי של הלקוח וישלח ללקוח. כך שכעת יש אצל השרת הזדוני את המפתח הסימטרי שבעזרתו יוכל לפענח את הקבצים המוצפנים שהלקוח שולח.

תיקון אפשרי – כדאי להחליף את פרוטוקול החלפת המפתחות בפרוטוקול אמין יותר כמו SSL. ולהוסיף מנגנון אימות סרטיפיקטים, שבו נבדקת אותנטיות ההודעה שהתקבלה, ויווצר מצב חריג במקרה שלא קיים סרטיפיקט או שהוא אינו מתאים לסרטיפיקט המצופה. במקרה כזה קשה לגורם זדוני להתערב באמצע כי עליו לדעת את המפתח הפרטי של הרשות המאשרת או לבקש סרטיפיקציה של הרשות המאשרת, אך הרשות המאשרת לא תנפיק סרטיפיקט עבור כתובת שאינה שייכת למבקש.