



# הצעת פרויקט לעבודת גמר י"ד הנדסאי תוכנה (שאלון: 714918)

בנושא  
**הצפנה ושיתוף קבצים**



שם הסטודנט: אופיר בן שימול

ת"ז: 215964610

מנחה: אילן פרץ

תאריך: 13/1/2026

מכילה: כנפי רוח, קריית נוער ירושלים (סמל מוסד: 140129)

## תוכן עניינים

1.	תיאור הנושא .....	2
2.	רקע תיאורטי .....	3
3.	תיאור הפרויקט .....	6
4.	הגדרת הבעיה האלגוריתמית .....	7
5.	הליכים עיקריים בפתרון בעיה בטכנולוגיות הנדסה מתקדמות .....	9
6.	הליכים עיקריים בתחום למידת מכונה - לא רלוונטי .....	11
7.	הליכים עיקריים במע' הפעלה / רשותות / תקשורת נתונים / אבטחת מידע .....	11
8.	תיאור פרוטוקולי תקשורת .....	18
9.	פיתוחים עתידיים .....	19
10.	תיאור טכנולוגיה הנדסה .....	20
11.	מסד נתונים .....	20
12.	פרטים פורמליים .....	23

## 1. תיאור הנושא

תחום אבטחת המידע עוסק בהגנה על מידע דיגיטלי מפני גישה בלתי מורשית, שנייה, חשיפה או אובדן. בעידן הדיגיטלי המודרני, מידע אישי, עסקי וארגוני מואחסן וMOVUBER באופן מקוון חלק משגרה יומיומית, דבר ההופך אותו לנכס בעל ערך גבוה וליעד מרכזי לאזויים שונים. מידע זה כולל מסמכים, קבצים, נתוניים אישיים ונתוניים ארגוניים, אשר חשיפתם או פגיעתם עלולה לגרום לנזקים כלכליים, משפטיים ותdeximiyim.

מערכות מודרניות לאחסון ושיתוף קבצים מבוססות לרוב על מודל שרת/לקוח, שבו הקבצים נשמרים בשרת מרכזי ומונגים למשתמשים דרך רשת האינטרנט. מודל זה מאפשר נוחות, זמינות ועובדת משותפת, אך יוצר בעיה הנדרשת משמעותית בכל הנוגע לאבטחת מידע רגיש. במערכות רבות, השרת מחזיק בגישה מלאה לתוכן הקבצים, ולכן נזקמת תורפה במקרה של פריצה, דליפת מידע, או שימוש לרעה מצד בעלי הרשות לתשתיית עצמה. בנוסף, שיתוף קבצים בין משתמשים או ארגונים מגביר את משטח התקיפה ואת הסיכון לשינוי, העתקה או חשיפה של מידע ללא ידיעת בעלי. כתוצאה לכך, קיימں צורך הנדרש בפיתוח פתרונות המאפשרים אחסון ושיתוף קבצים תוך שימוש אמון בשרת, והבטחת סודיות המידע גם בסביבות מרובות משתמשים ובתנאי עבודה יומיומית.

כיום קיימות מערכות רבות לאחסון ושיתוף קבצים, הן לשימוש פרטי והן לשימוש ארגוני, כגון Google Drive, Dropbox, OneDrive, ועוד. מערכות אלו מאפשרות ה

עליה, הורדנה ושיתוף של קבצים בין משתמשים, ולעתים כוללות מנגנון הרשות וניהול משתמשים בסיסיים. עם זאת, ברוב המקרים המידע נשמר ומוניהל בשרת מרכזי בעל גישה מלאה לתוכן הקבצים, כך שספק השירות או בעלי גישה לתשתיית יכולם תאורתית לקרוא או לעבד את המידע. מצב זה יוצר תלות גבוהה באמינותה של הרשות ובאמצעי האבטחה שלו, ומדגיש את הפער הקיימן בין הנוחות התפעולית של מערכות אלו לבין רמת האבטחה הנדרשת בעת עבודה עם מידע רגיש.

כדי להתמודד עם האתגרים באחסון ושיתוף מידע רגיש, תחום אבטחת המידע מגדר מספר עקרונות יסוד המנחים את תכנון המערכות הנדרשות. העיקרי המרכזי הוא סודיות (Confidentiality), אשר מטרתו להבטיח כי רק גורמים מורשים יכולים לצפות בתוכן המידע. עיקנון נוסף הוא שלמות המידע (Integrity), המבטיח שהמידע לא שונה, נמחק או נפגע במהלך האחסון או העברה, ללא זיהוי. לצד אלו קיימים עיקנון אינטואציית הזרה (Authentication), המאפשר לוודא את זהות המשתמש במערכת, ועיקנון הרשותות (Authorization), הקובע אילו פעולות מותרות לכל משתמש ביחס למידע. עקרונות אלו מהווים בסיס תיאורטי הכרחי לכל מערכת העוסקת במידע רגיש, והם מגדרים את הדרישות הנדרשות של הפתרון לעמוד בהן, עוד לפני בחירת טכנולוגיות או אלגוריתמים ספציפיים.

## 2. רקע תיאורטי

אחד המרכיבים המרכזיים בפתרונות אבטחת מידע הקיימים כיום הוא שימוש בהצפנה לצורך הגנה על מידע דיגיטלי. הצפנה מאפשרת להפוך מידע קריאלייצוג בלתי קריא, כך שרק גורמים מורשים יוכלו לשחזר את המידע למשמעותו המקורי. בתחום זה מקובל להבחין בין שתי שיטות עיקריות: **הצפנה סימטרית והצפנה א-סימטרית**, שלכל אחת מהן מאפיינים, יתרונות ושימושים שונים.

הצפנה סימטרית מבוססת על מפתח יחיד המשמש הן להצפנה והן לפענוח של המידע. שיטה זו מאופיינת בביצועים גבוהים ויעילות חישובית, ולכן מתאימה במיוחד להצפנה נפח' מידע גדולים, כגון קבצים ומסדי נתונים. עם זאת, האתגר המרכזי בהצפנה סימטרית הוא ניהול המפתח, שכן יש להעביר את המפתח בין הצדדים بصورة מאובטחת מבלתי חשוף אותו לגורמים בלתי מורשים. פתרונות קיימים נדרשם להמודד עם בעיה זו באמצעות מנגנונים משלימים לניהול והעברת מפתחות.

הצפנה א-סימטרית, לעומת זאת, מבוססת על זוג מפתחות: מפתח ציבורי ומפתח פרטי. המפתח הציבורי משמש להצפנה, בעוד שהמפתח פרטי משמש לפענוח. שיטה זו מאפשרת העברת מידע بصورة מאובטחת גם בין גורמים שלא שיתפו סוד מראש, ולכן נפוצה במיוחד בתחום אימות זהות והעברת מפתחות. עם זאת, הצפנה א-סימטרית דורשת חישובים מורכבים יותר, ולכן אינה מתאימה להצפנה כמוניות גדולות של מידע, אלא משמשת לרוב כחלק ממנגנון משולב עם הצפנה סימטרית.

בנוסף להצפנה, פתרונות אבטחת מידע עושים שימוש בפונקציות **Hash** קרייפטוגרפיות, אשר אין מיועדות להסתתרת המידע אלא לאימות שלמותו. פונקציית Hash מקבלת קלט באורך משתנה ומפיקה ערך תקצר באורך קבוע, כך שכל שינוי קטן בקלט גורם לשינוי משמעותי בתוצאה. באמצעות פונקציות אלו ניתן לzechות האם מידע עבר שינוי, פגעה או זיהוי במהלך האחסון או ההעברה. פונקציות Hash מהוות רכיב בסיסי בבדיקה שלמות נתונים, אימות זהויות וחתימות דיגיטליות, והן משלימות את תהליכי האבטחה הכלל של המידע.

להלן אלגוריתמים המהווים פתרון אשר הוזכר קודם בנושא זה:

AES הוא אחד האלגוריתמים המרכזיים המספקים פתרון ישיר לבועה של הגנה על תוכן הקובץ עצמו. זהו אלגוריתם הצפנה סימטרית, המשתמש באותו מפתח להצפנה ולפענוח, ומאפשר להפוך כל קובץ לקריאה בלתי אפשרית ללא מפתח מתאים. היתרון המרכזי שלו הוא יעילות גבוהה ומהירות ביצוע, גם כאשר עובדים עם קבצים גדולים או עם כמות גבוהה של פעולות הצפנה ביום עבודה שוטף. השימוש ב-AES מאפשר להצפין את הקובץ כבר הצד הלוקוט, לפני העלהתו לשרת, וכך אם הקובץ נחשף, יורט או נשמר בסביבה שאינה מאובטחת לחלווטין, לא ניתן יהיה להבין ממנו דבר. בכך האלגוריתם נותן מענה ברור לבועה שהוצגה בפרק הקודם: שמירה על סודיות הקובץ גם מחוץ למסגרת הארגונית.

שילוב של חזק ההצפנה יחד עם ביצועים יציבים הופך את AES לכל' משמעותי במערכות לשיתוף קבצים רגילים, ומספק שכבת הגנה בסיסית אך הכרחית בתהילך העבודה.

Camellia הוא אלגוריתם הצפנה סימטרית נוסף. האלגוריתם מקבל מידע קריא והופך אותו לרצף בלתי מובן באמצעות מפתח ייחיד המשמש גם להצפנה וגם לפענוח. Camellia פועל בבלוקים של 128 ביט ומשלב מספר סבבים של החלפה, ערבוב ופיזור נתונים, באופן המבטיח שגם במקרה של "ירוט הקובץ לא ניתן יהיה להבין את תוכנו ללא המפתח המתאים. בזכות ייעילות גבוהה ויציבות בביטויים, Camellia מתאים להצפנה קבצים שלמים או נתונים בהיקף גדול, ומהווה שכבת הגנה מרכזית בתהילך אבטחת המידע.

RSA נותן מענה לבעה משלימה בתהילך, העברת מפתחות ההצפנה בצורה מאובטחת בין משתמשים. האלגוריתם מבוסס על זוג מפתחות, ציבורי ופרטי, ומאפשר ליצור ערוץ תקשורת שבו ניתן לשלוח את מפתח ההצפנה מבלי לחשוף אותו לגורם לא מורשה. הרעיון פשוט, המשתמש מצפין את מפתח הסימטרי באמצעות המפתח הציבורי של הנמען, ורק המפתח הפרטי של אותו נמען מסוגל לפענה אותו. בצורה זו, גם אם השרת או כל גורם אחר יירט את ההודעה בדרך, הוא לא יוכל להבין את המפתח. כך האלגוריתם פותר את אחת הביעיות הקritisיות במערכות הצפנה קבצים שהיא: כיצד לשתף מפתחות בצורה בטוחה לחלווטין, בלי לסמוך על השרת מבלי לסכן את סודיות הקובץ. השימוש באלגוריתם זה מפריד בין יכולת האחסון לבין יכולת קראאת המידע, ומחזיר את השליטה המלאה למשתמשים המורשים בלבד.

ECC (Elliptic Curve Cryptography) הוא אלגוריתם הצפנה אסימטרית המשמש להעברת המפתח האסימטרי בצורה מאובטחת בין משתמשים. האלגוריתם מבוסס על זוג מפתחות ציבורי ופרטי כאשר המפתח הציבורי משמש להצפנה והמפתח הפרטי משמש לפענוח. האלגוריתם מאפשר לשני הצדדים ליצור מפתח משותף מבלי לחשוף את המידע המתמטי העומד מאחוריו, ובכך מונע מגורמים לא מושרים לשחרר את המפתח גם אם הם מזינים לתקשורת. האלגוריתם מספק רמת אבטחה גבוהה יותר שימוש במפתחות קצרים ויעילים, ולכן מותאם במיוחד לנסיבות רשות שבahn יש צורך להניב מפתחות הצפנה בצורה בטוחה ומהירה.

SHA-256 פותר בעיה שונה לחלווטין מזה של הצפנה, הוא אינו מסתיר את תוכן הקובץ, אלא מבטיח שהקובץ לא שונה במהלך הדריך. האלגוריתם מייצר "טביעה אצבע" ייחודית לכל קובץ, כך שניינו של אפילו ביט יחיד יוצר טביעה אצבע שונה לחלווטין. בצורה זו ניתן לוודא שהקובץ שהורד או נשמר בשרת הוא בדיק אוטו קובץ שהועלה במקור, ללא זיופים, מניפולציות או פגיעה בשלמות המידע. שימוש באלגוריתם זה מאפשר גם אימיות זהויות ויצירת חתימות דיגיטליות, כך שימושים יכולים לבדוק שמא�, בקשה או פעולה מסוימת אכן הגיעו מקורה אמיתי. אלגוריתם זה מבטיח שהמידע נותר מדויק

ואמינ. שילוב של שלושת האלגוריתמים יחד יוצר מערכת מאובטחת שאינה מסתמכת רק על הצפנה התוכן, אלא גם על מגנון חזק המגן מפני שינויים וטעויות.

Whirlpool הוא אלגוריתם **Hash cryptographic** המשמש לייצור טביעת אצבע דיגיטלי של הקובץ. האלגוריתם מקבל את תוכן הקובץ ומפיק ממנו ערך ייחודי וחד כיווני באורך 512 ביט, המציג את מבנה הנתונים באופן מדויק. שניינி קטן ביותר בקובץ, אפילו ביט יחיד, יוצר ערך Hash שונה שונה לחלווטין, ולכן אלגוריתם זה מאפשר לאזחות האם הקובץ נשאר שלם או עבר שינוי במהלך העברה או האחסון. בשילוב עם חתימה דיגיטלית ניתן גם לאמת את מקור המידע וגם להבטיח שלא בוצעו בו מניפולציות, מה שהופך את האלגוריתם לכלי מרכזי לשמירה על אמינות ושלמות המידע במערכת.

להלן טבלה אשר מציגה השוואה בין אלגוריתמים קריפטוגרפיים שונים, תוך פירוט יתרונות והחסרונות של כל אלגוריתם בהיבטים של אבטחה,יעילות ו שימושיות.

חסרונות	יתרונות	סוג	אלגוריתם
<ul style="list-style-type: none"> <li>• תלות מלאה בשמירה על סודיות המפתח העברת המפתח דורשת מגנון אבטחה נפרד</li> </ul>	<ul style="list-style-type: none"> <li>• ביצועים גבוהים בהצפנה</li> <li>• נפח מידע גדולים</li> <li>• מבנה פשוט ליישום</li> <li>• עמידות גבוהה בפני התקפות ידועות</li> </ul>	סימטרי	AES
<ul style="list-style-type: none"> <li>• מרכיבות פנימית גבוהה יותר של שלבי ההצפנה דורש ניהול מפתח זהה להצפנה ולפענוח</li> </ul>	<ul style="list-style-type: none"> <li>• רמת אבטחה גבוהה עם פיזור ובלבול חזקים</li> <li>• מתאים להצפנה קבצים שלמים</li> <li>• יציב גם בעומס עבודה גבוההים</li> </ul>	סימטרי	Camellia
<ul style="list-style-type: none"> <li>• חישובים כבדים</li> <li>• אינם מתאימים להצפנה קבצים גדולים</li> </ul>	<ul style="list-style-type: none"> <li>• מאפשר העברת מפתחות ללא שיתוף סוד מוקדם</li> <li>• הפרדה ברורה בין הצפנה לפענוח</li> </ul>	א-סימטרי	RSA
<ul style="list-style-type: none"> <li>• שימוש מרכיב יותר מבחינה מתמטית</li> <li>• ר蓋ש לטעויות בהגדרת פרמטרי העקום</li> </ul>	<ul style="list-style-type: none"> <li>• רמת אבטחה גבוהה עם מפתחות קצרים</li> <li>•יעילות חישובית טובה</li> <li>• בתקשות רשות</li> </ul>	א-סימטרי	ECC

<ul style="list-style-type: none"> <li>• אין מספק הצפנה או הסתרת מידע</li> <li>• אין אפשרות שחזור נתונים</li> </ul>	<ul style="list-style-type: none"> <li>• מזהה כל שינוי קטן בנתונים</li> <li>• פلت קבוע באורך ידוע</li> </ul>	<ul style="list-style-type: none"> <li>• בדיקת שלמות ואימות מידע</li> </ul>	SHA-256
<ul style="list-style-type: none"> <li>• דרוש משאבי חישוב גבוהים יותר</li> <li>• אין מספק הצפנה או ניהול מפתחות</li> </ul>	<ul style="list-style-type: none"> <li>• פلت ארוך במילוי המקטין סיכוי להתקנשות רגישות גבוהה לשינויים בקלט</li> </ul>	<ul style="list-style-type: none"> <li>• בדיקות שלמות ואימות מידע</li> </ul>	Whirlpool

### 3. תיאור הפרויקט

המערכת היא מערכת לניהול ושיתוף קבצים רגילים, המיועדת למשתמשים וארגוני הזרים לרמת אבטחה גבוהה בעת עבודה עם מידע דיגיטלי. מטרת המערכת היא לאפשר אחסון, שיתוף וגישה לקבצים בצורה מבוקרת, תוך שמירה על סודיות המידע ושלמותו גם כאשר הקבצים נשמרים וmovedרים דרך תשתיות חיצונית. המערכת נועדה לתת מענה לצורכי של משתמשים המעוניינים לשלוט בגישה למידע שלהם, מבלי לחסוך את תוכן הקבצים לגורמים שאינם מורשים.

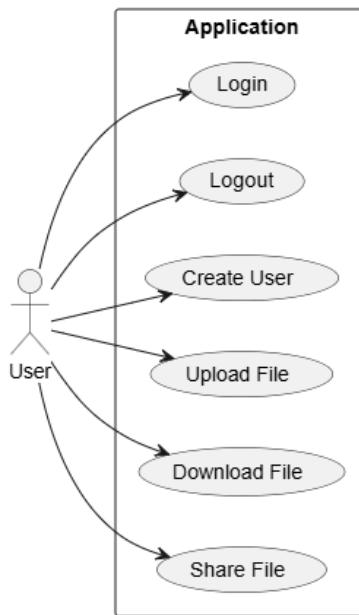
המערכת מיועדת לשימוש של משתמשים פרטיים וארגוני, כגון עובדים בארגון, צוותים מקצועיים או גופים המטפלים במידע רגיש. כל משתמש פועל מתוך חשבון אישי, יכול להיות חלק מארגון או קבוצה מוגדרת. המשתמשים יכולים להעלות קבצים, לשמר אותם במערכת, ולשתף אותם עם משתמשים אחרים בהתאם להרשות שנקבעו. השימוש במערכת מתבצע דרך ממשק Web פשוט וברור, המאפשר עבודה נוחה ללא צורך בידע טכני מוקדם.

מנקודת מבטו של המשתמש, תהליך העבודה במערכת כולל מספר פעולות עיקריות. תחילת המשתמש נרשם למערכת ומתחבר לחשבונו. לאחר ההתחברות הוא יכול להעלות קבצים מהמחשב האישי, לצפות ברשימת הקבצים השמורים, ולהגדיר עם מי הקבצים ישוטפו. המשתמש יכול לבחור אילו משתמשים אחרים יכולים לגשת לקובץ מסוים, ולהגדיל את הגישה בהתאם לצרכים שלו. כאשר משתמש מקבל גישה לקובץ, הוא יכול להוריד אותו או לצפות בו, בהתאם להרשות שניתנו לו.

הבעיה המרכזייה שהמערכת פותרת עבור המשתמש היא חוסר האמון במערכות אחסון רגילות, שבן ספק השירות או גורמים בעלי גישה לתשתיות יכולים להיחשף לתוכן הקבצים. במערכות רגילות, המשתמש נדרש לשמור על הרשות שישמור על סודיות המידע, דבר שאינו תמיד מספק בעת עבודה עם מידע רגיש. המערכת מצמצמת תלות זו ומאפשרת למשתמשים לעבוד עם קבצים בצורה מאובטחת יותר, תוך שליטה טובה יותר בהרשות ובגישה למידע.

באופן זה המערכת מספקת פתרון שימושי וברור למשתמשים הקיימים לשילוב בין נוחות העבודה לבין רמת אבטחה גבוהה, ומאפשרת עבודה יומיומית עם קבצים רגילים בסביבה דיגיטלית מובוקרת.

להלן תרשימים מסווג **UseCase** אשר מסביר אילו פעולות המשתמש יכול לבצע.



המשתמש הוא הגורם החיצוני היחיד הפועל מול המערכת, והוא יכול לבצע פעולות של ייצור משתמש (Create User), התחברות (Login) לצורך זיהוי ואיומות, וניתוק (Logout) לסיום עבודה מאובטח. לאחר ההתחברות המשתמש רשאי להעלות קבצים למערכת (Upload File), להוריד קבצים אל המחשב האישי (Download File) ולשתף קבצים עם משתמשים אחרים בהתאם להרשות שניתנו (Share File). התרשימים ממחישים בצורה פשוטה וברורה את תחומי האחריות של המערכת ואת הפעולות הזמיןות למשתמש, מלבד להיכנס לאופן המימוש הפנימי של כל פעולה.

## 4. הגדרת הבעה האלגוריתמית

הבעה האלגוריתמית שבה עוסקת מערכת זו נוגעת לאבטחת מידע דיגיטלי בעט אחסון ושיתוף קבצים בין משתמשים שונים, בסביבה שאינה בהכרח מאובטחת. מדובר בבעיה מורכבת, הנובעת מה הצורך לאפשר זרימה חופשית ונוחה של מידע בין גורמים מורשיים, תוך מניעה מוחלטת של גישה, שינוי או ניצול של המידע על ידי גורמים בלתי מורשיים. הבעיה אינה עוסקת רק בהגנה על המידע עצמו, אלא גם בשמירה על אמינותו, שלמותו והיכולת לשנות בಗישה אליו לאורך זמן.

באופן פורמלי, ניתן להגדיר את הבעיה כך: נתון קובץ דיגיטלי המיועד לאחסן או לשיתוף בין משתמשים, וכן אוסף של משתמשים בעלי רמות הרשאה שונות. נדרש תהליך אלגוריתמי אשר יפיק יציג מאובטח

של הקובץ, כך שתokin הקובץ לא יהיה קרייא או ניתן לשחזר על ידי גורם שאינו מורה, גם אם יש לו גישה לנוטונים המאוחסנים או המועברים. במקביל, על התהיליך להבטיח שימוש מושתמש מורה יוכל, בתנאים המתאימים, לשחזר את הקובץ למצבו המקורי ללא פגיעה בנוטונים.

מרכיב מרכזי בבעיה האלגוריתמית הוא הצורך להבטיח שלמות מידע. כאמור, יש לוודא כי הקובץ שהתקבל על ידי משתמש זהה לקובץ המקורי נשמר או נשלח, ללא שינוי, השמטה או הוספה של נתונים. האלגוריתם נדרש לאפשר זיהוי חד-משמעי של כל שינוי שנעשה בקובץ, בין אם במכoon ובין אם כתוצאה מתקלה או תקיפה, ולהתריע על כך לפני שימוש במידע.

MORECOMBOT הבעיות גוברת לאור העובדה שהמידע עשוי לעבור דרך רשתות פתוחות ולהישמר בתשתיות חיצונית, אשר אין בשליטת בעלי המידע. לכן, האלגוריתם נדרש לצמצם את התלות באמון בגורמים חיצוניים, ולפעול בצורה שבטיחה את אבטחת המידע גם בסביבה עוינת. במצב זה, אין להניח שהשרת או אמצעי האחסון הם מהימנים, ועל האלגוריתם להבטיח שהמידע ישאר מגן גם אם תשתיית זו תיפגע.

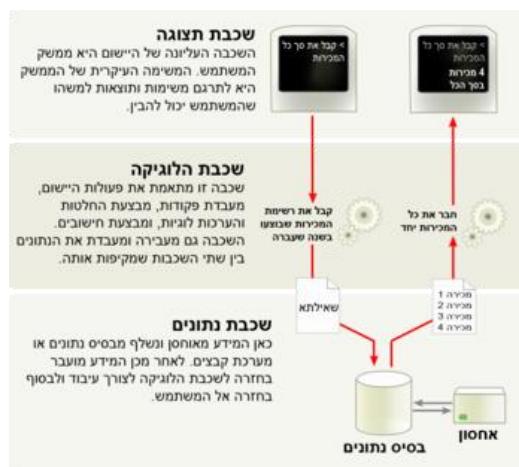
הבחירה באלגוריתמים **Camellia, ECC, Whirlpool** נובעת מהדרוש לבנות מערכת מאובטחת, עליה וgemäßיה המתאימה לסייעת עבודה מודרנית. **Camellia** מעניק הצפנה סימטרית חזקה ומהירה, הדומה ברמת האבטחה שלה ל-AES, ומאפשר להצפין קבצים שלמים בצורה יציבה ועליה גם בהיקפי מידע גדולים. **ECC** נבחר כפתרון להצפנה אסימטרית ולהעברת מפתחות מאובטחת, בזכות יכולת שלו לספק רמת אבטחה גבוהה באמצעות מפתחות קצרים במיוחד, מה שמייעל את ביצוע המערכת ומקטין עומסי חישוב לעומת חלופות מסורתיות. **Whirlpool** משלים את המערכת באמצעות מנגן Hash אמין היוצר טביעה אצבע ייחודית לכל קובץ, ובמבטיה יכולה לבדוק הוכחה זיהוי שינויים באופן מדויק. שילוב של שלושת האלגוריתמים יוצר פתרון הכלול הצפנה, העברת מפתחות ובדיקה שלמות שלושת המרכיבים החיוניים לאבטחת מידע וגייס במערכת לשיתוף קבצים.

לסיכום, הבעיות האלגוריתמית של מערכת זו משלבת מספר אתגרים מרכזיים: שמירה על סודיות המידע, הבטחת שלמותו. פתרון הבעיה מחייב הגדרה מדויקת של רצף פעולות אלגוריתמי, אשר יטפל בכל אחד מהאתגרים הללו באופן משולב ובקבוי, ויאפשר עבודה בטוחה עם קבצים רגילים לאורך כל מחזור חייהם.

## 5. הליכים עיקריים בפתרון בעיה בטכנולוגיות הנדסה מתקדמות

מערכת זו מבוססת על **מודל שלוש שכבות**, אשר מחלק את מבנה התוכנה לשולשה חלקים נפרדים, כך שלכל שכבה יש אחראיות מוגדרת וברורה. בנוסף לכך, המערכת פועלת במסבנה של שרת ללקוח (Client-Server), שבו קיימת הפרדה ברורה בין צד הלקוח (Frontend) לבין צד השירות (Backend). צד הלקוח אחראי על הצגת המשך למשתמש, קליטת פעולותיו ושליחת בקשות למערכת, בעוד צד השירות אחראי על עיבוד הבקשות, ניהול הלוגיקה העסקית, בקרת הרשאות ו互動 אינטראקטיבי עם המשתמשים. שילוב זה יוצר יישום מסובך **Full-Stack**, שבו חל כל שכבות המערכת, החל ממשך המستخدم ועד לניהול הנתונים והלוגיקה, ממומשת אחת שלמה ופועלות יחד במסבנה מבוקר ומוסדר.

להלן תרשימים המתאר את רצף הזרימה בין קצהו לקצה:



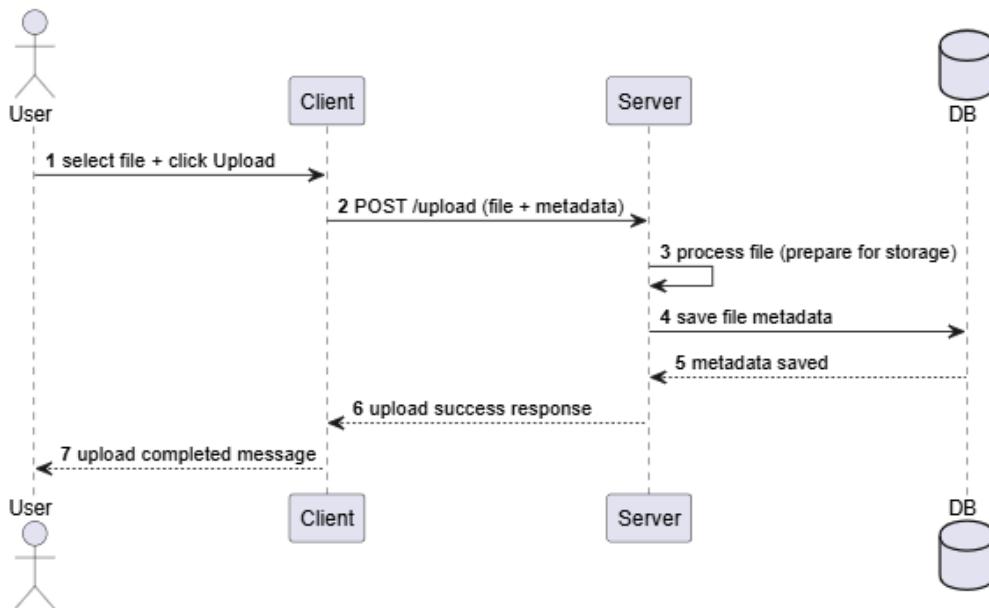
**שכבת הציגות** אחראית על הצגת המשך למשתמש ועל קליטת פעולותיו, כגון התחברות למערכת, העלאת קבצים, צפייה בקבצים ושיטופם. בפרויקט זה שכבה זו ממומשת כחלק מיישום ה-Web המבוסס על **Spring Boot**, אשר אחראי על ניהול הבקשות מהדף והציג התגובה המתאימה למשתמש. **Spring Boot** מטפל בתקשורת בין הדף לשרת ובניהול זרימת הנתונים בין שכבות התציגות לשכבות הפנימיות של המערכת, ללא צורך במימוש מנגנון תקשורת חיצוני נפרד. שכבה זו מתמקדת באינטראקטיביות עם המשתמש ואינה כוללת לוגיקה עסקית או גישה ישירה למסד הנתונים.

**שכנת הלוגיקה** בה מתבצע כל העבודה הנדסי ש מגדר את>User> המערך: קבלת פעולות מהמשתמש (העלאה, שיטוף, הורדה), בדיקת הרשאות, ניהול תהליכי הצפנה והפענוח. בפרויקט זה שכבה זו ממומשת בקוד Java בטור **Spring Boot**, באמצעות **מחלקות שירות (Services)** שמריצות את כל החוקים והשלבים של התהליך הקרייפטוגרפי והעסקי, ובמציאות הפרדה ברורה בין שכבת התציגות לבין שכנת הלוגיקה. בנוסף, שכבה זו אחראית על עבודה עם אובייקטים של נתונים שימושיים בתוך המערכת בפורמט **JSON**, ועל תרגום ביניהם לבין המבנים הפנימיים של המערכת, כך שהזרימה נשארת מסודרת, עקבית וניתנת לתחזוקה.

**שכבה נתונים** אחראית על שמירה, שטיפה וניהול של כל המידע שהמערכת מייצרת ומשתמשת בו, כולל נתונים משתמשים וארגוני, מידע של קבצים, הרשותות שיתוף, ערכי Hash, מפתחות מוצפנים ומידע נוספת נוספת. בפרויקט זה שכבה זו ממומשת באמצעות מסד נתונים מסוג NoSQL, המאפשר אחסון גמיש של מסמכים ומבנה נתונים בהתאם לצרכי המערכת, מוביל להיות מוגבל לסקמה קשיה כמו במודול נתונים. במקרה, Spring Boot, שכבה זו מיושמת באמצעות **משקי גישה לנתונים** (Repository), האחראית לבצע פעולות CRUD בצורה מבודדת משככת הלוגיקה, כך שהלוגיקה העיקרית אינה תליה בפרטי האחסון. Maven משמש לניהול התלוויות של הפרויקט ולהטמעת ספריות החיבור למסד הנתונים, מה שמאפשר בניה מסודרת, תחזוקה קלה והרחבנה עתידית של המערכת.

היתרון המרכזי במודול זה הוא הפרדה ברורה בין אחורי השכבות, מה שמאפשר פיתוח יעיל יותר, תחזוקה פשוטה, והרחבת המערכת בעתיד ללא תלות בחלוקת אחרים. בכל תהליך עבודה המידע זורם מהמשתמש אל שככת הלוגיקה, ומשם אל שכבת הנתונים וחזור אחורה לאחר העבודה.

התרשימים הבאים מציג **תרשים רצף** (Sequence Diagram) המתאר את תהליך העלאה קובץ במערכת, מנוקודת מבטן זרימת הפעולות בין המשתמש, צד הלוקה, צד השרת ומסד הנתונים. התרשים ממחיש את סדר הפעולות הלוגי מרגע בחירת הקובץ על ידי המשתמש ועד לסיום תהליך ההעלאה ואחסון המידע הנלווה במערכת, תוך הפרדה ברורה בין אחורי הרכיבים השונים.



בתחילת התהליך המשתמש בוחר קובץ ולחוץ על פעלת ההעלאה דרך משק הלוקה. צד הלוקה שולח בקשה העלאה לשרת הכללת את הקובץ ואת המידע הנלווה אליו. השירות מקבל את הבקשה וմעבד את הקובץ לצורך הכנה לאחסון ושמור במסד הנתונים את המידע של הקובץ, כגון מזיהה, בעלות ומידע תיאורי. עם השלמת הפעולה, השירות מחזיר ללקוח תגובה הצלחה, והלקוח מציג למשתמש הודעה המאשרת כי הקובץ הועלה בהצלחה למערכת.

## 6. הליכים עיקריים בתחום למידת מכונה - לא רלוונטי

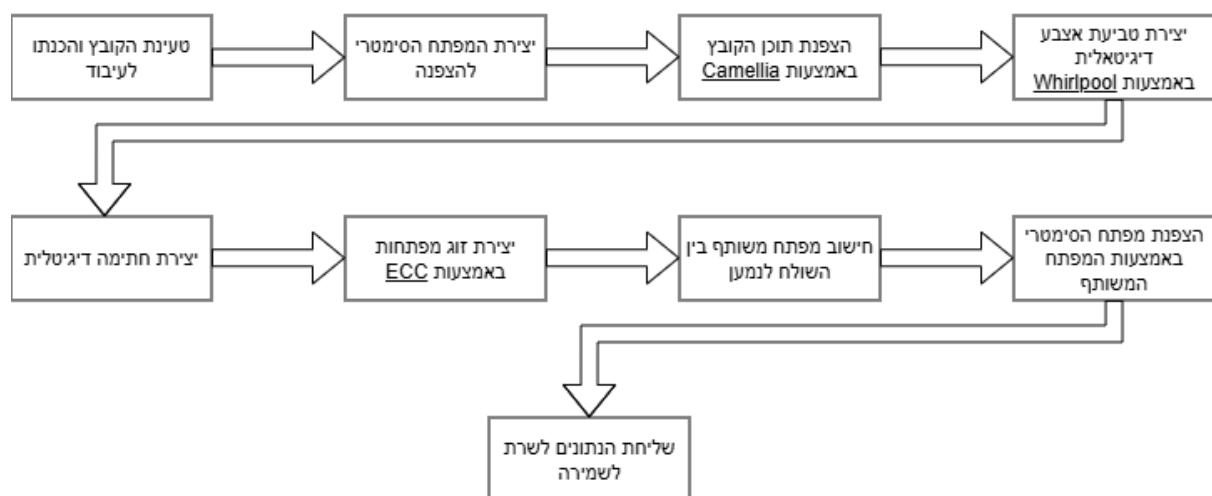
הצעה זו לא עוסקת בלמידה מכונה, لكن סעיף זה לא רלוונטי

## 7. הליכים עיקריים במע' הפעלה / רשות / תקשורת / אבטחת מידע

הערה חשובה: ההתייחסות למודל Client/Server מופיעה בהרחבה בסעיף 5 של הצעה.

כדי להמחיש כיצד עקרונות המערכת מיושמים בפועל בתחום הנדסי מובנה. לצורך כך הוגדר תהליך הצפנה הקובץ, המתואר באמצעות תרשימים זרים, אשר מציג את רצף השלבים מרוגע בחירת הקובץ על ידי המשתמש ועד לשימושו بصورة מאובטחת בשרת. התרשימים מאפשרים הבנה ויזואלית של חלוקת האחריות בין הלוקוח לשרת ושל השימוש באלגוריתמים הקRIPTוגרפיים השונים. בהמשך פרק זה יפורט כל שלב בתרשימים בנפרד, תוך הסבר מעמיק על פועלתו, תפקido בתחום, והאופן שבו האלגוריתמים הקRIPTוגרפיים פועלים ברמה העקרונית.

### תהליך ההצפנה הקובץ



### שלב 1 - טיענת הקובץ והכנתו לעיבוד

בשלב הראשון בתחום ההצפנה, המשמש בוחר קובץ מקומי מהמחשב או מהתקן אחסון אחר ומעלה אותו למרכז דרכם משקל ה-Web. הקובץ נתען לזכרון לצד הלוקוח, לפני שליחתו לשרת, וזאת על מנת למנוע חשיפה של תוכן הקובץ במצב גולמי לגורמים חיצוניים. בשלב זה מתבצעת הכנה ראשונית של הקובץ לעיבוד קRIPTוגרפי, הכוללת קריאה של תוכן הקובץ כזרם נתונים בינאריארי (bytes) והגדלת מבנה אחד המאפשר הפעלת אלגוריתמים קRIPTוגרפיים עליו. שלב זה אינו כולל הצפנה בפועל, אלא

מהוות בסיס הנדרси הכרחי להמשך התהיליך, בכך שהוא מבטיח שהקובץ יטופל בצורה מובוקרת, עיקבית ובلتית תלויה בסוגו או בגודלו.

### שלב 2 - ייצרת המפתח הסימטרי להצפנה

בשלב זה המערכת יוצרת מפתח הצפנה סימטרי אקריאי, אשר ישמש להצפנה תוכן הקובץ. המפתח מוצר בצד הלוקוח באמצעות **מחולל מספרים אקריאים קרייפטוגרפי**, במטרה להבטיח רמת אבטחה גבוהה ולמנוע אפשרות לניחוש או שחזור המפתח. מפתח זה הוא מפתח זמן ויחודי לכל קובץ, והוא עשוי שימוש במפתחות קבועים או משותפים מראש. הבחירה בהצפנה סימטרית בשלב זה נובעת מהיעילות הגבוהה שלה בעבודה עם קבצים גדולים, ומהיכולת להצפין נפח מידע משמעותי בזמן קצר. ייצרת המפתח בשלב נפרד מאפשרת שליטה מלאה בתהיליך ההצפנה ומהוות בסיס לשילוב מאוחר יותר עם מנגנוני אבטחה נוספים לניהול והגנה על המפתח עצמו.

### שלב 3 - הצפנה תוכן הקובץ באמצעות Camellia

לאחר ייצרת המפתח הסימטרי, מתבצעת הצפנה תוכן הקובץ באמצעות אלגוריתם Camellia. אלגוריתם זה הוא אלגוריתם הצפנה סימטרי מסוג Block Cipher, הפועל על בלוקים בגודל קבוע של נתונים ומצפין כל בלוק באמצעות המפתח שנוצר בשלב הקודם. תהיליך ההצפנה כולל חלוקה של תוכן הקובץ לבלוקים, והפעלת סדרת סבבים (Rounds) על כל בלוק, כאשר בכל סבב מתבצעות פעולות מתמטיות ולוגיות כגון ערבות נתונים, החלפות והצפנה באמצעות תת-מפתחות הנגזרים מהמפתח הראשי.

המטרה של תהיליך זה היא להפוך את הנתונים המקוריים לנתונים בלתי קריאים לחלוטין עבור כל גורם שאינו ברשותו את המפתח המתאים. הבחירה ב-Camellia נובעת מהיותו אלגוריתם מאובטח ויעיל, המתאים להצפנה קבצים גדולים, ומאפשר שמירה על סודיות תוכן הקובץ גם אם הנתונים המוצפנים חשפים או נשלחים דרך רשת לא מאובטחת.

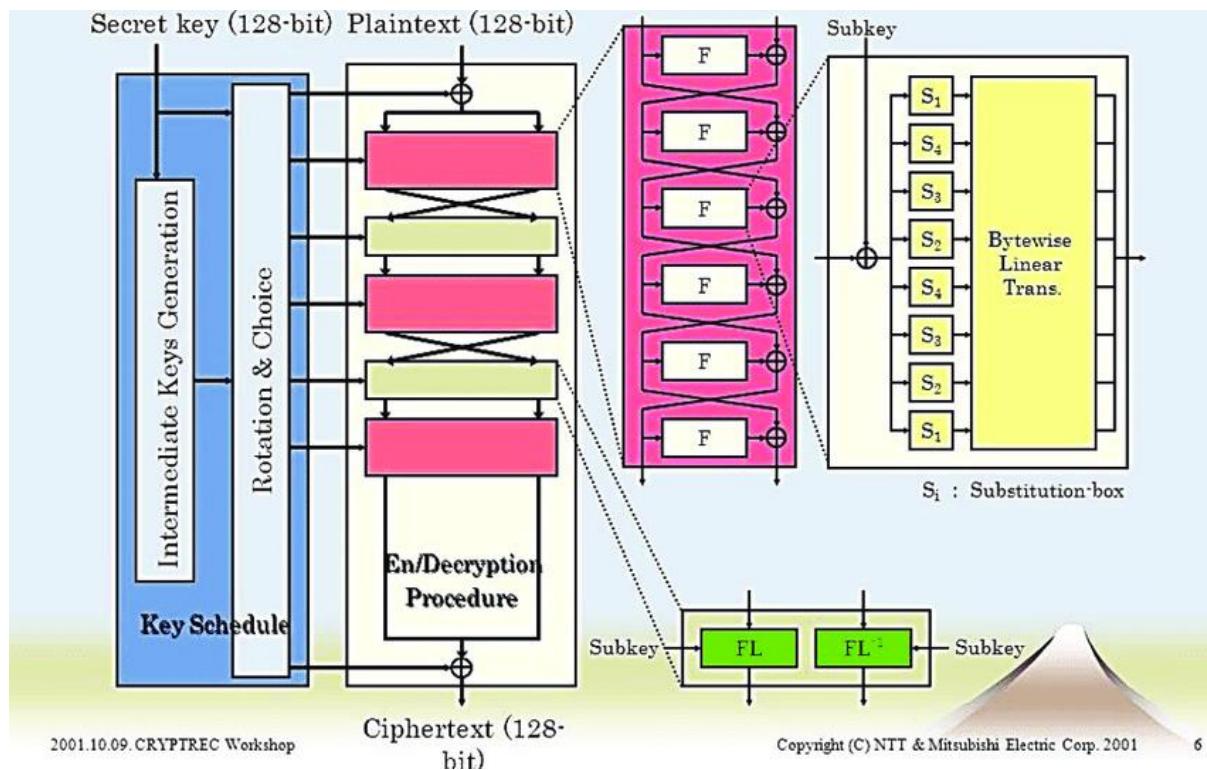
### **אלגוריתם Camellia**

אלגוריתם זה משמש להצפנה סימטרית מסוג Block Cipher, אשר פועל על בלוקים קבועים של נתונים בגודל 128 ביט. בשלב הראשון, תוכן הקובץ מחולק לרצף של בלוקים בינאריים בגודל זהה, אשר כל בלוק מוצפן בנפרד באמצעות אותו מפתח סימטרי. המפתח הראשי עבר תהיליך של הרחבת מפתח (Key Schedule) שבמהלכו נגזרים ממנו תת-מפתחות שונות המשמשים בכל אחד מסבבי ההצפנה.

תהליך ההצפנה עצמו מתבצע במספר סבבים עוקבים, כאשר בכל סבב מופעלות פונקציות קרייפטוגרפיות הכוללות פעולות של החלפה (Substitution) באמצעות טבלאות קבועות, ערבות ביטים (Permutation), פעולות XOR בין נתונים הבלוק לתת המפתחות. פעולות אלו נעשו ליצור בלבול (Diffusion) ופיזור (Confusion) של הנתונים, כך שכל שינוי קטן בנתוני הקלט או במפתח גורם לשינוי

משמעותי בפלט. בסיום הסבבים מתקיים בлок מוצפן שאינו משמר קשר ישיר או ברור לנוטים המקוריים. תהליך זה חוזר על עצמו עבור כל בלוק בקובץ, עד להצפנה הקובץ כולו.

להלן תמונה אשר מייצגת את תהליך ההצפנה:



התרשימים מציג את מבנה ההצפנה של אלגוריתם Camellia ואת זרימת המידע בתהליך ההצפנה והפענוח. מצד שמאל מופיע שלב ייצור תת-המפתחות, (Key Schedule) שבו המפתח הסימטרי הראשי עובר עיבוד ונגזרים ממנו מפתחות משנה המשמשים לאורך הסבבים. במרכז התרשימים מוצג תהליך ההצפנה עצמו, שבו בלוק הנתונים עובר סדרת סבבים חוזרים הכוללים פונקציות פנימיות (F) ופעולות טרנספורמציה, כאשר בכל סבב מושלב תת מפתח שונה. מצד ימני של התרשימים ניתן לראות את מבנה פונקציית החלפה והערבוב, הכוללת שימוש ב- S-boxes ופעולות ליניאריות, שמטרתן ליצור פיזור ובלבול של הנתונים. השימוש בין סבבים חוזרים, תת מפתחות וטרנספורמציות קריפטוגרפיות מבטיח שהקשר בין הקלט לפלט יטושטש לחЛОוטין, כפי שנדרש מאלגוריתם הצפנה סימטרי חזק.

#### שלב 4 - ייצור חתימה דיגיטלית באמצעות Whirlpool

לאחר הצפנה תוכן הקובץ, המערכת יוצרת חתימה דיגיטלית שמטරתה להבטיח את שלמות המידע ולאפשר זיהוי של שינוי או פגעה בקובץ המוצפן. בשלב זה מופעלת פונקציית Hash ה-Hash ה-Hash ה-Hash על תוכן הקובץ המוצפן, ומתקיים ערך Hash קבוע המציג באופן ייחודי את הנתונים. פונקציית Whirlpool מבוססת על עקרונות של ערבות והפצה של ביטים, כך שכל שינוי קטן בתוכן הקובץ יגרום לשינוי משמעותי בערך Hash המתתקבל.

החתימה הדיגיטלית משתמשת בטביעת אצבע של הקובץ המוצפן, ומאפשרת בשלב מאוחר יותר לוודא שהקובץ לא שונה מאז יצירת החתימה. שלב זה אינו מחייב מידע נוסף, אלא מוסיף שכבת אבטחה המשלימה את תהליך הצפנה, בכך שהוא מאפשר שימוש הנתונים לפני תהליך הפענוח והשימוש בקובץ.

### **אלגוריתם Whirlpool**

אלגוריתם זה הוא אלגוריתם Hash קריפטוגרפי, שמטרתו להפיק ערך תקציר (Hash) באורך קבוע של 512 ביט מטור קלט באורך משתנה. בשלב הראשון, תוכן הקובץ המוצפן עובר תהליך הכנה הכלל השלמה (Padding) וחלוקת לבוקים בגודל קבוע של 512 ביט. כל בלוק מעבד באופן סדרתי, כך שההתוצאה של עיבוד בלוק אחד משפיעה על עיבוד הבלוק הבא.

LIBT האלגוריתם מבוססת על פונקציית דחיסה קריפטוגרפית הפעלת במספר סבבים עוקבים. בכל סבב מביצעות פעולות של החלפה לא ליניארית באמצעות טבלאות קביעות (S-boxes), ערבות ליניארי של בתים ופעולות XOR בין נתוני הבלוק לערך הביניים המctrber. פעולות אלו נועדו ליצור פיזור ובלבול של המידע, כך של שינוי קטן בקלט גורם לשינוי משמעותי בערך ה-Hash המתתקבל. בסיום עיבוד כל הבלוקים מתתקבל Hash סופי, המשמש בטיעת אצבע ייחודית של הקובץ, ואיןו אפשרות שחזור של המידע המקורי מתוךו.

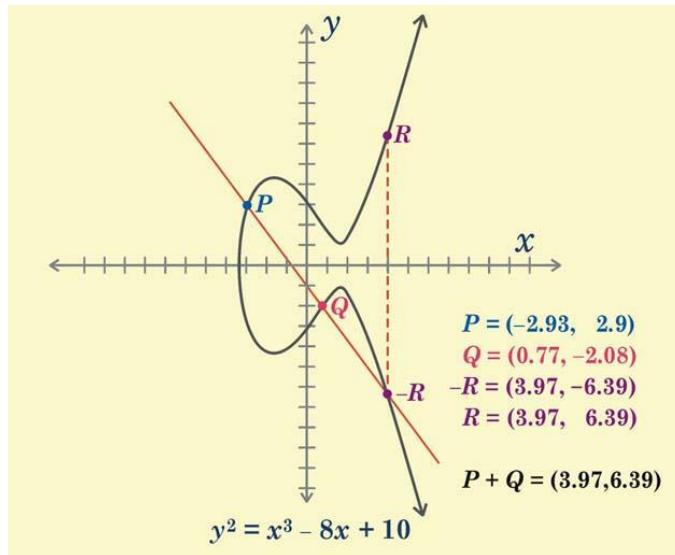
### **שלב 5 - הצנת מפתח הצפנה באמצעות ECC**

לאחר הצנת תוכן הקובץ באמצעות מפתח סימטרי, נדרש להגן על מפתח זה מפני חשיפה לגורםים לא מורשים. לצורך כך נעשה שימוש בהצנה איסימטרית מבוססת ECC, אשר פועלת באמצעות זוג מפתחות ציבורי ופרטני. בשלב זה, המפתח הסימטרי המשמש להצנת הקובץ מוצפן באמצעות המפתח הציבורי של המשתמש המושה, וכך רק בעל המפתח הפרטני התואם יוכל לפענה אותו. תהליך זה מבטיח שהמפתח הסימטרי אינו נשמר או מועבר בצורה גלויה, ואף גורם בעל גישה לשרת אינו מסוגל לחזורו.

### **אלגוריתם ECC**

אלגוריתם זה הוא אלגוריתם הצפנה איסימטרי המבוסס על תכונות מתמטיות של **עקומים אליפטיים** מעל שדות סופיים. האלגוריתם עושים שימוש בזוג מפתחות, מפתח ציבורי ומפתח פרטי, כאשר המפתח הציבורי נוצר מהתמטית מהמפתח הפרטני באמצעות כפל נקודה על העקום. תהליך זה יוצר קשר חד-כיווני, שבו חישוב המפתח הציבורי מהמפתח הפרטני הוא פשוט, אך שחזור המפתח הפרטני מטור המפתח הציבורי נחשב בלתי ישם חישובי. במהלך פעולת ההצנה, נעשה שימוש במפתח הציבורי כדי להצפין מידע או מפתח אחר, וכך רק בעל המפתח הפרטני התואם יוכל לבצע את פעולה הפענוח. הביטחון של ECC נשען על הקושי שפתרון בעיית הלוגריתם הדיסקרטי בעקומים אליפטיים, אשר מקשה על תוקף להסיק את המפתח הפרטני גם כאשר המפתח הציבורי ידוע. יתרונו

המרכזי של ECC הוא ביכולת לספק רמת אבטחה גבוהה באמצעות מפתחות קצרים יחסית, דבר המאפשר יעילות חישובית ושיילוב נוח במערכות מודרניות לאבטחת מידע וניהול מפתחות. האיור מציג את אופן הפעולה של אלגוריתם ECC באמצעות חיבור נקודות על עקום אליפטי. נקודת



הבסיס  $P$  היא נקודה קבועה וידועה מראש, המשמשת כנקודת המוצא לכל החישובים הקרייפטוגרפים במערכת. נקודה  $Q$  מייצגת נקודה נוספת על העקום, אשר להרבה מתקבלת כתוצאה מכפל נקודות הבסיס  $P$  בפתח פרטי סודי, ולכן משמשת כפתח הציבורי. לצורך חיבור נקודות, מועבר ישיר דרך הנקודות  $P$  ו- $Q$ , והנקודה השלישית שבה הישר חותך את העקום מסומנת כ- $-R$ . נקודה זו משוקפת ביחס לציר ה- $x$  ומתקבלת הנקודה  $R$ , שהיא תוצאה החיבור  $Q + P$ . פעולה חיבור זו מהווה את הבסיס לכפל נקודות בעקום אליפטי, אשר משמש ליצירת מפתחות והצפנה ב-ECC. המבנה הגיאומטרי המוצע באיור מדגים כיצד פעולות אלו פשوطות לביצוע בכיוון אחד, אך קשות לשחזר לאחר ידיעת המפתח הפרטי.

#### שלב 6 - חישוב מפתח משותף בין השולח לנמען באמצעות ECC

בשלב זה מתבצע חישוב של **מפתח משותף** (Shared Secret) בין השולח לנמען, המבוסס על מגנון חילופי מפתחות באמצעות ECC. כל אחד מהצדדים עושה שימוש בפתח פרטי שלו ובפתח הציבורי של הצד השני, ובמצע כפל נקודה על העקום האליפטי. למרות לכך צד משתמש בערכיהם השונים, תוצאה החישוב המתקבלת זהה עבור שניהם ומיצרת מפתח משותף שאינו נשלח בפועל דרך הרשת. מפתח זה משמש כבסיס קרייפטוגרפי מאובטח להמשך התהליך, ובמיטח כי רק השולח והנמען יכולים לגזור אותו ערך, גם אם המידע המועבר ביניהם נחשף לגורם חיצוני. שלב זה מאפשר הקמת סוד משותף ללא העברת מפתח גלוי ומהווה מרכיב מרכזי באבטחת תהליך שיתוף הקבצים.

#### שלב 7 - הצפנה המפתח הסימטרי באמצעות המפתח המשותף

נעsha שימוש במפתח המשותף לצורך הצפנה נוספת של המפתח הסימטרי המשמש להצפנה הקובץ. המפתח המשותף משמש כבסיס להצפנה סימטרית קלה ומהירה, כך שהמפתח הסימטרי של הקובץ אינו נשמר או מועבר בצורה גלויה בשום שלב. תהליך זה יוצר שכבת הגנה נוספת, שבה רק גורמים שהצליחו לגוזר את אותו מפתח משותף יוכל לפענה את מפתח ההצפנה של הקובץ. שילוב זה מבטיח שגם במקרה במקורה של גישה לנוטונים המאוחסנים בשרת, לא ניתן יהיה להשתמש במפתח הסימטרי ללא ביצוע מוצלח של שלב חילופי המפתחות הкриיפטוגרפיים.

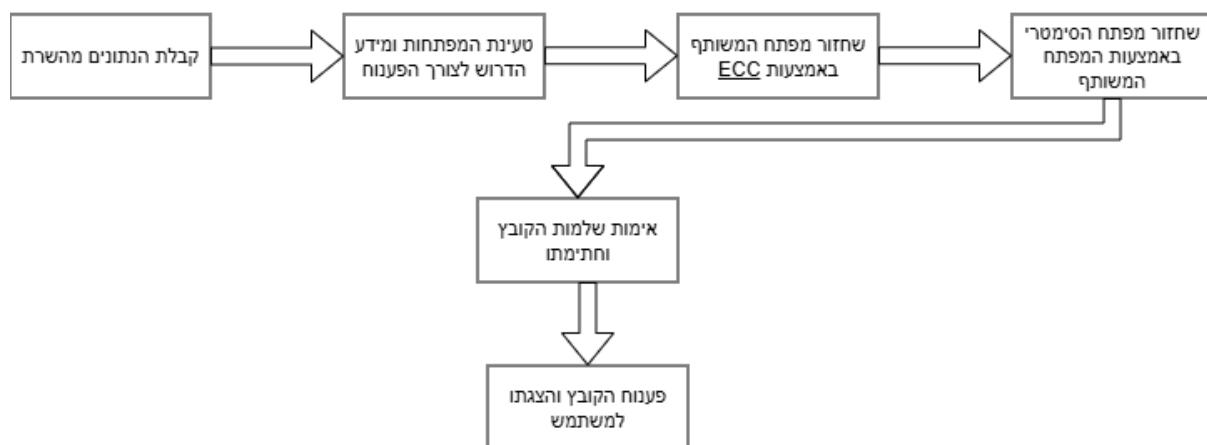
#### **שלב 8 - שליחת הנתונים המוצפנים לשרת ושמירתם**

בשלב הסופי בתהליך, הנתונים המוצפנים נשלחים לשרת לצורך אחסון. נתונים אלו כוללים את תוכן הקובץ לאחר הצפנה סימטרית, את המפתח הסימטרי כשהוא מוצפן באמצעות המפתח המשותף, וכן את ערך ה-Hash שנוצר לצורך בדיקת שלמות. השרת משתמש בשלב זה כגורם אחסון בלבד, ואין לו יכולת לפענה את הקובץ או את מפתח ההצפנה, מאחר שאינו ברשותו את המפתחות הפרטיים הנדרשים לכך. הפרדה זו מבטיחה שגם במקרה במקורה של גישה לא מורשית לשרת או לנוטוני, תוכן הקבצים נשאר חסוי. שמירת הנתונים בצורה זו מאפשרת גישה מאובטחת לקבצים על ידי משתמשים מורשים בלבד, תוך שמירה על עקרון של מינימום אמון בשרת.

#### **תהליך פענוח הקובץ**

לאחר השלמת תהליך ההצפנה והעברת הקובץ המאובטח, נדרש תהליך פענוח מבוקר מצד הנמען על מנת לשחזר את הקובץ המקורי בצורה בטוחה. תהליך זה מבוסס על אימותות שלמות ומקור המידע, שחוור מפתח ההצפנה הסימטרי והפעלת פועלות הפענוח עצמה. כל שלב בפענוח נשען על נתונים שנוצרו בשלב ההצפנה, ומטרתו לוודא שהקובץ שהתקבל לא שונה בדרך, וכן נשלח מגורם מורשה, ונitin לפענוח רק על ידי הנמען המתאים. כתום יפורטו שלבי הפענוח שמבצעת המערכת מצד מקבל, לפי סדר ביצועם.

#### **תהליך פענוח הקובץ**



### קבלת הנתונים מהשרת

בשלב הראשון של תהליך הפענוח, המערכת בצד הנמען מקבלת מהשרת את חבילת הנתונים המלאה שנשלחה בתהליך ההצפנה. חבילה זו כוללת את הקובץ המוצפן, מפתח ההצפנה הסימטרי כשהוא מוגן באמצעות ECC, טביעת האצבע הדיגיטלית של הקובץ, החתימה הדיגיטלית והמידע הנלווה. בשלב זה המערכת עדין אינה מפענחת את הקובץ, אלא רק טוענת את כל הנתונים הדרושים להמשך תהליך האימות והפענוח. שלב זה מהוות את נקודת הכניסה של המידע המוצפן אל תהליך השחזור המאובטח.

### טעינת מפתחות והכנות לתהליך האימות

בשלב זה המערכת טוענת את זוג המפתחות הקרייפטוגרפיים של הנמען, הכוללים מפתח פרטי ומפתח ציבורי, וכן את המפתח הציבורי של השולח הנדרש לאימות החתימה הדיגיטלית. שלב זה נדרש להכין את כל החומרים הקרייפטוגרפיים הדרושים לביצוע פעולות האימות ושחזור מפתח ההצפנה בהמשך. בשלב זה עדין לא מתבצעת בדיקה או פענוח של הקובץ עצמו, אלא רק הכנה של סביבת העבודה הקרייפטוגרפית לצורך המשך התהליך בצורה מאובטחת.

### שחזור המפתח המשותף באמצעות ECC

בשלב זה המערכת משתמשת במפתח פרטי של הנמען ובמפתח הציבורי של השולח כדי לבצע את תהליך ה-ECC ולחשב את המפתח המשותף. חישוב זה מתבצע באופן מקומי בצד הנמען ואינו מצרי' העברת מפתח כלשהו דרך הרשת. המפתח המשותף שמתתקבל זהה למפתח שנוצר בצד השולח בשלב ההצפנה, והוא משמש כבסיס לשחזור מפתח ההצפנה הסימטרי. שלב זה מהוות את החליה המרכזית בתהליך הפענוח, שכן הוא מאפשר גישה למפתח ההצפנה מוביל לחסוף אותו לגורמים אחרים.

### שחזור מפתח ההצפנה הסימטרי

בשלב זה המערכת משתמשת במפתח המשותף שהושב באמצעות ECC כדי לשחזור את מפתח ההצפנה הסימטרי של Camellia. המפתח הסימטרי שהוא שמור בשרת בצורה מוגנת מופיע כעת באופן מקומי בצד הנמען, מוביל שהמפתח נחשף במהלך ההעברה. עם סיום שלב זה, המערכת מחזיקה לראשונה במפתח ההצפנה המקורי הדרוש לפענוח הקובץ עצמו, אך עדין אינה מפענחת את הקובץ עד לאimoto שלמותו ומקורו.

### אimoto שלמות הקובץ והחתימה הדיגיטלית

בשלב זה המערכת מחשבת מחדש ערך Hash על הקובץ המוצפן שהתקבל, באמצעות אותו אלגוריתם Hash שבו נעשה שימוש בשלב ההצפנה. לאחר מכן מתבצעת בדיקה של החתימה הדיגיטלית באמצעות המפתח הציבורי של השולחן, כדי לוודא שטביעת האצבע אכן נוצרה על ידו ולא שונתה בדרך. אם ערך ה-Hash המוחושב תואם לערך החתום והחתימה מאומנת בהצלחה, ניתן להסיק שהקובץ שלם, לא עבר שינוי, ונשלח מגורם מורשה. רק במקרה שבו שלב זה מסתיים בהצלחה, המערכת ממשיכה לשלב הפענוח של הקובץ עצמו.

#### פענוח הקובץ והציגו למשתמש

בשלב הסופי של תהליך הפענוח, ולאחר שהמערכת אימנה בהצלחה את שלמות הקובץ ואת מקורו, מtbody פענוח הקובץ עצמו באמצעות אלגוריתם Camellia והמפתח הסימטרי ששוחרר בשלב הקודם. פעולה פענוח מחזירה את הקובץ למצבו המקורי והקריא. לאחר מכן המערכת שומרת את הקובץ המופיע במקום שנבחר על ידי המשתמש או מציגה אותו בהתאם לצורך. שלב זה מסכם את תהליך האבטחה כולו ו מבטיח שהקובץ נפתח רק לאחר שכל בדיקות האבטחה עברו בהצלחה, ללא חשיפה מוקדמת של מידע רגיש ולא תלות באמינות הרשת.

## 8. תיאור פרוטוקולי תקשורת

המערכת עשויה שימוש בפרוטוקולי תקשורת סטנדרטיים להעברת מידע בין צד הלוקוח לבין צד הרשת בצורה מאובטחת, אמינה ו מבוקרת. פרוטוקולים אלו מאפשרים תקשורת רציפה בין רכיבי המערכת, תוך שמירה על סודיות ושלמות המידע המועבר.

### • פרוטוקול HTTPS / HTTP

התקשורת בין הלוקוח לשרת מתבצעת באמצעות פרוטוקול HTTP בגרסתו המאוובטחת HTTPS. שימוש ב-HTTPS מבטיח כי כל הנתונים הנשלחים בין הצדדים מועברים בערז Mozfun, כך שגם במקרה של "ירוט" תעבורת הרשת, לא ניתן יהיה לקרוא או לשנות את המידע. בנוסף, הפהוטוקול מאפשר אימונות של זהות הרשת ומספק הגנה מפני שינוי נתונים במהלך הטעבה.

### • TCP/IP

ברמה הבסיסית יותר, התקשרות בין הלוקוח לשרת נשענת על פרוטוקול IP/TCP האחראי על העברת המידע ברשות בצורה אמינה ו מסודרת. TCP/IP מבטיח הגעה מלאה של הנתונים, שמירה על סדר המנות, ויזיהו תקלות בהעברה. פרוטוקול זה מהווה את התשתייה שעלייה פועלם שאר פרוטוקולי התקשרות במערכת.

### • REST API

בנוסף לתקשרות בין הלוקוח לשרת, המערכת עשויה שימוש ב-API REST חיצוני לצורך קבלת

מידע אבטחתי תומך החלטה. בעת יצירת חשבון משתמש או שינוי סיסמה, השירות שלוח בקשה REST לשירות [Pwned Passwords](#) של [I Have I Been Pwned](#), באמצעות נקודת הῆקזה endpoint הייעודית לבדיקה אונומית של סיסמאות. הבקשה מתבצעת באמצעות קריאה ל-

בפורמט {GET /range/{hashPrefix}}

כאשר {hashPrefix} מייצג חלק מערך ה-Hash של הסיסמה, בהתאם למנגנון בדיקה מבוסס anonymity-k בתגובה מוחזרת רשיימת ערכיו Hash חלקים המאפשרת לשרת לקבוע האם הסיסמה מופיעה במאגרי דיליפות ידועים, מבלי להעיר את הסיסמה עצמה או את ערך ה-Hash המלא. תוצאת הבדיקה מאפשרת למערכת למשתמש, לחסום שימוש בסיסמאות שנפרצו, ולהעלות את רמת האבטחה של חשבונות המשתמשים. שימוש ב-endpoint זה מאפשר שילוב מידע אבטחתי חיצוני בתהיליך קבלת החלטות של המערכת, תוך שמירה על סודיות המידע הרגיש והימנעות מחשיפת פרטי הזדהות.

#### פורמט JSON

- JSON הוא פורטט נתונים טקסטואלי מובנה שמקובל מאוד ב-API REST והוא משמש לייצוג הבקשות והtagיות בין מערכות בצורה אחורית. בהקשר של API REST, הלוקוח שלוח בקשה שכללת נתונים בפורטט JSON, והשרת מחזיר תשובה בפורטט JSON שמכילה שדות ברורים כמו סטטים, פרטיטים ותוצאות. היתרון בכך הוא שהפורטט פשוט לפירוק ולעיבוד, כך קל מאד לוודא מה התקבל ומה צריך לעשות עם זה בצד המקבל.

## 9. פיתוחים עתידיים

- בשלב עתידי ניתן להרחב את מערכת במגוון יכולות מתקדמות שיחזקו את חווית השימוש ואת רמת האבטחה. אפשרויות מרכזיות היא הוספה צפיה לשירה בקבצים דרך הדפסה, כך שימושים יכולים לצפות במסמכים ובתמונה ללא הורדה. פיתוח נוסף עשוי לכלול תמייה בשיתוף קבצים, המאפשר להעניק הרשות גישה למספר משתמשים בארגון בצורה פשוטה ומובנית. בנוסף, ניתן לשלב מנגנון מהיקאה אוטומטית לקובץ לאחר זמן קצוב או מספר הורדות מוגדר, כדי להקטין את הסיכון לדיליפות מידע.
- אפשרות אחרת היא הטמעת מערכת ניהול ארגונית, שתאפשר למנהלים לצפות ברישומות קבצים, לקבוע מדיניות הצפנה, ולהגדיר רמות הרשות שונות לכל משתמש. בעתיד ניתן יהיה להוסיף גם דוחות פעילות, שיציגו היסטוריית גישה לקובץ, ניסיונות הת לחברות, ופעולות שבוצעו במערכת. פיתוחים נוספים עשויים לכלול תמייה בהעלאת קבצים גדולים במיוחד באמצעות פרוטוקולי חלוקה למקטעים, או שילוב זיהוי דו שלבי להגברת אבטחת הגישה למערכת. כל רעיון כזה נדרש לשפר את המערכת מבלי לשנות את ליבת העבודה הקיימת.

## 10. תיאור טכנולוגיה הנדסה

- **שפת תכנות:**
  - **Java 21** שפת תכנות עילית OOP המאפשרת לכתוב קוד גם בצד השרת וגם בצד הלקוח.
  - **מסגרות עבודה (Frameworks)** (Spring Boot 3.5.9, Business Logic Layer) המשמשת לבניית הלוגיקה המרכזית של המערכת וניהול תהליכי העבודה בה. המסגרת מאפשרת קבלת בקשות מהמשתמשים, עיבודן בהתאם לכללי המערכת והחזרת תוצאות מתאימות, תוך מתן תשתיות מובנית לניהול בקשות וזרימת נתונים בצורה מסודרת. במודל שלוש שכבות מסגרת זו פועלת שכבת הלוגיקה היא אחראית על מימוש כללי המערכת, ניהול הרשאות ואינטגרציה עם מסד הנתונים, תוך שמירה על הפרדה ברורה בין הצגת המידע לבין אופן הטיפול והאחסון שלו. הפרדה זו תורמת לאבטחת המערכת, לתחזוקה נוחה ולהרחבתה עתידית.
  - **Vaadin 24.9.5** היא ספריית עזר לבניית ממשקי משתמש לישומי Web, אשר פועלת בצד של Spring Boot בצד השרת. ייעודה של Vaadin הוא לאפשר יצירת ממשק משתמש גרפי מלא באמצעות קוד Java בלבד, ללא צורך בכתיבה ישירה של JavaScript, CSS, HTML. במסגרת עבודה עם Vaadin, היא משלבת שכבת התצוגה של המערכת, ומאפשרת להציג טפסים, כפורים, טבלאות וMSCים שונים, ולקשר אותם ישירות ללוגיקה העסקית שבשרת. שילוב זה מאפשר זרימת מידע ישירה ואמובטת בין המשק למערכת, תוך שמירה על הפרדה בין שכבת התצוגה לשכנת הלוגיקה.
  - **סביבה פיתוח (IDE):**
  - **VSCode 1.108** היא סביבת פיתוח קלה וგמישה המשמשת לכתיבה, ניהול וDİאג של קוד בפרויקט תוכנה. היא תומכת בעבודה עם פרויקטי Java וכלי בנייה כגון Maven, ומאפשרת פיתוח יעיל ונוח באמצעות הרחבות וכליים מובנים.
  - **ספריות עזר: אין ספריות הכרען לשימוש בפרויקט.**

## 11. מסד נתונים

בפרויקט זה נעשה שימוש במסד נתונים **MongoDB** מסוג **NoSQL**, המתאים במיוחד למערכות שבahn מבני הנתונים אינם קבועים מראש ועלולים להשתנות בהתאם לצרכים תפעוליים ואבטחתיים. בנויגוד למסדי נתונים רלוונטיים, מסד נתונים זה מאפשר אחסון מידע במבנה מסמי גמש, שבו כל

רשומה יכולה להכיל שדות שונים ומבנה מקוונים, ללא תלות בסכמתה קשיחה. תכונה זו מתאימה לאופי המערכת, שבה נשמרים סוגים שונים של נתונים כגון פרטי משתמשים, מפתחות ציבוריים, מידע נלווה על קבצים והרשאות גישה, אשר אינם בהכרח אחידים במבנה.

השימוש במסד נתונים SQLoS נאפשר שמיירה נוחה של מידע הקשור לקבצים מוצפנים, מלבד לאחסן את תוכן הקבצים עצם במצב קרייא. המידע המאוחסן כולל נתונים תיאוריים בלבד, כגון מזהים, בעלות על קבצים, תארכי פעללה ומידע אבטחה נלווה, דבר המאפשר הפרדה ברורה בין תוכן רגש לבין נתונים ניהול. בנוסף, מבנה מסמכי מאפשר קישור טבעי בין משתמשים, קבצים והרשאות, באמצעות מזהים ושדות מקוונים, ללא צורך בקשרים רלוונטיים מורכבים.

יתרונות נוספים של שימוש ב-SQLoS בפרויקט הוא היכולת להרחיב את המבנה הנתונים בעתיד בקלות, למשל הוספת שדות אבטחה חדשים, נתונים בקרה או מידע פעולה'י נוספים, מלבד לפגוע נתונים קיימים. גישה זו תורמת לתחזוקה נוחה של המערכת, להטאה לדרישות משתנות ולפיתוח עתידי, תוך שמירה על ביצועים ויציבות. הבחירה במסד נתונים SQLoS תומכת במטרות האבטחתיות של הפרויקט ומאפשרת ניהול מידע גמיש, מאובטח ומאזין.

להלן פירוט האוספים המרכזיים במסד הנתונים של המערכת והאופן שבו כל אוסף משמש לניהול המידע ולתמכה בתהליכי העבודה והבטחה של הפרויקט.

#### • **אוסף פרטי משתמש**

אוסף זה משמש לשימרת נתונים הדיאליים והאינטגרטיביים של המשתמשים במערכת, כולל פרטי התחברות, מפתח ציבורי והקשר לקבצים השייכים להם. המידע באוסף זה מאפשר ניהול הרשאות, דיאליים משתמשים ושירוף פעולות וקבצים לבנייהם בצוואר מבוקרת ומأובטחת.

#### • **אוסף קבצים מוצפנים**

אוסף זה משמש לשימרת מידע נלווה לקבצים המאוחסנים במערכת, כאשר תוכן הקבצים נשמר במצב מוצפן בלבד. המידע באוסף זה מאפשר ניהול קבצים, בדיקת שלמותם, שליטה בגישה אליהם ו恢舊ה מאובטח על ידי משתמשים מורשים בלבד.

להלן מידע מפורט אשר יוצג במסד הנתונים:

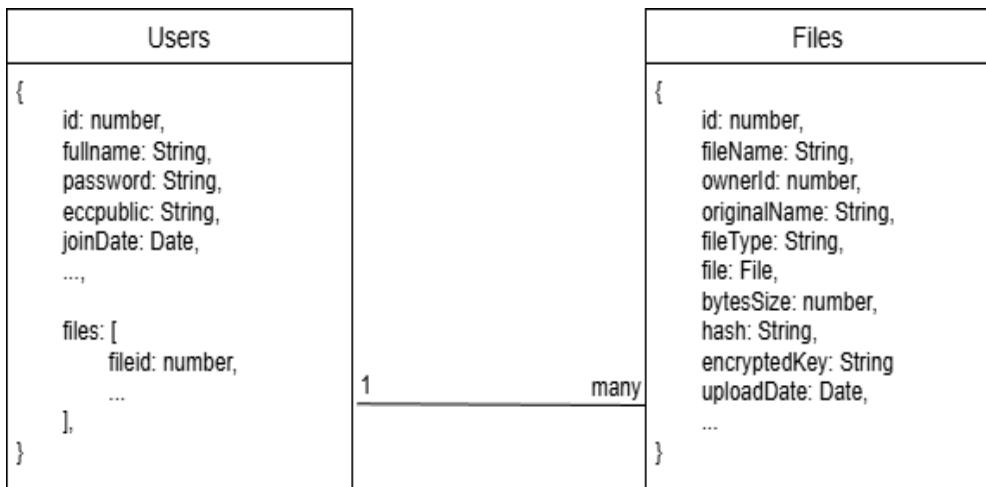
#### מידע של פרטי משתמש:

- שם המשמש
- שם מלא
- סיסמה
- מפתח ECC ציבורי
- קבצים שמורים במערכת

מידע של קבצים מוצפנים:

- מספר זהה
- שם המשתמש של הבעלים
- שם הקובץ המקורי
- סוג הקובץ
- קובץ מוצפן
- תאריך העלאה
- גודל בティים
- ערך ה-Hash של הקובץ המוצפן
- מפתח סימטרי מוצפן

להלן תרשימים ERD של האוספים המציג את המידע:



ניתן לראות בתרשימים שני אוספים, האויף השמאלי הוא אויף המשתמשים (Users) המציג את המידע לכל משתמש כגון: שם מלא, סיסמה, תאריך הצטרפות, ועוד שנמצא בצד ימין של התרשימים שהוא אויף קבצים מוצפנים (Files) שם מאוחסן כל המידע על הקובץ כגון: שם הקובץ, סוג הקובץ, ו��ונתו, והקובץ עצמו, הקשר בין האוספים הללו הוא 1:N, כאשר יש משתמש אחד שמקשרים אליו כמה קבצים.

**הערה חשובה:** ניתן שבמהלך בניית המערכת יתוסף מידע נוסף או ישנה מעט, لكن התרשימים והמידע המופיעים בסעיף זה מהווים תצוגה כללית ולא דבר וודאי.

## 12. פרטיים פורמליים

**לוח זמנים:**

לסיום עד תאריך	שלבי עבודה
<b>1.12.2025</b>	בחירת פרויקט, חקירה ולמידה לעומק של נושא הפרויקט
<b>11.12.2025</b>	כתיבה והגשת הצעת הפרויקט לאישור משרד החינוך
<b>13.1.2026</b>	ימוש הקוד של האלגוריתם המרכזי, ביצוע בדיקות ושיפורים
<b>3.2.2026</b>	בנייה צד שרת
<b>17.2.2026</b>	בנייה מסד הנתונים ושלובו
<b>3.3.2026</b>	בנייה צד לקוח
<b>24.3.2026</b>	כתיבת ספר הפרויקט
<b>7.4.2026</b>	הגשת הפרויקט כולו (ספר + קוד) להגנה וקבלת ציון מגן

**חתימת מנהה הפרויקט:** \_\_\_\_\_

**חתימת הסטודנט:** \_\_\_\_\_

**חתימת רכז המגמה :** \_\_\_\_\_