

# תעודה דיגיטלית

אורון, אורי, מקסים ועמית סגלוב

# מי המציא את התעודה הדיגיטלית

בשנת 1976, שהיווה בסיס חשוב לפיתוח (PKI) ויטפילד דיפי ומרטין הלמן - הם הציעו את רעיון התשתית הציבורית למפתחות התעודות הדיגיטליות.

שהיה הבסיס, Netscape בשנת 1994 בחברת SSL (Secure Sockets Layer) רון ריוסט וריצ'רד סולומון - הם פיתחו את הפרוטוקול להצפנת תעבורה באינטרנט ולשימוש בתעודות דיגיטליות להזדהות אתרים.

ISO. לתעודות דיגיטליות בשנת 1988 במסגרת ארגון התקינה הבינלאומי X.509 פיל זימרמן - הוא פיתח את התקן

OpenSSL, Let's Encrypt פיתוחים נוספים - חברות ואנשים רבים תרמו לפיתוח תקנים, פרוטוקולים ותוכנות לשימוש בתעודות דיגיטליות, כמו ועוד.

כך שהמצאת התעודות הדיגיטליות הייתה תהליך מצטבר שכלל תרומות של אנשים, חברות וארגונים שונים במהלך שנים רבות, החל מרעיונות תיאורטיים ועד ליישומים מעשיים

# מהי תעודה?

במשפט אחד, תעודה משמשת כ-container למפתח הציבורי של בעל התעודה ומכילה חתימה דיגיטלית המספקת ערבות כי אכן מחזיק התעודה הוא מי שהוא טוען שהוא. בנוסף, קיימים מספר שדות (הרחבות) המגדירים כיצד יש להעביר את התעודה, איסורים על שימוש במפתח שנמצא בה ופרטים רבים נוספים

# קריפטוגרפיה והצפנות

קריפטוגרפיה עוסקת בהבטחת תקשורת מאובטחת בין ישויות מגורמים חיצוניים. לדוגמה, אם הראוטר שלך שולח מידע אישי (כמו הנתונים לא תרצה שמישהו יאזין לערוץ התקשורת ויחלץ את PayPal מספר כרטיס אשראי או תמונות מביכות מכיתה ד') לראוטר של האלה.

במדעי המחשב, כל "סוד" מתחיל ונגמר במפתחות. קריפטוגרפיה של מפתח ציבורי נשענת על מפתחות להצפנה ופענוח מידע. המפתחות קשורים מתמטית - מידע שמוצפן באחד ניתן לפענוח רק עם השני. שימוש במפתח יחיד לשתי הפעולות הוא הצפנה סימטרית, ובזוג מפתחות (ציבורי ופרטי) מדובר בהצפנה אסימטרית - המפתח הפרטי נשמר חשאי ואילו הציבורי מוטבע בתעודה דיגיטלית שמפורסמת במאגר נגיש.

הבעיה המרכזית בקריפטוגרפית מפתח ציבורי היא שאין קשר מובנה בין המפתח הציבורי לבעליו. תוקף יכול להתחזות למישהו אחר על ידי שימוש במפתח הציבורי שלו. איך נוכל לאמת את זהות הצד השני? האם המפתח שקיבלנו באמת ממי שחשבנו? האם הוא עדיין תקף? קריפטוגרפיית מפתח ציבורי מאבטחת את המפתח הציבורי עצמו אך לא את הבעיות של רכישה, ביטול, הפצה, אימות וחשוב מכל - קישור המפתח לישות מזוהה בעולם האמיתי. התקשורת יכולה להיות פרטית אך לא מאומתת.

נדרש גוף מוסמך שייתן מענה לפערים אלו על ידי אישור בעלות על מפתחות - וזה נעשה באמצעות תעודות דיגיטליות. תעודה דיגיטלית מקשרת בין מפתח ציבורי לזהות מאומתת של בעליו, ומאפשרת תקשורת פרטית ומאומתת. האמון בגוף המנפיק הופך למשתנה חדש במשוואת האבטחה.

# איך עובדות חתימות דיגיטליות

השולח מחשב את הפיסה האקראית של הנתונים/הודעה

הפיסה מוצפנת עם המפתח הפרטי של השולח

הפיסה המוצפנת היא החתימה הדיגיטלית, מוסיפה לנתונים

הנמען מפענח את החתימה עם המפתח הציבורי של השולח

הנמען מאמת שהפיסה שפוענחה תואמת לפיסה המחושבת של הנתונים





# יתרונות של חתימות דיגיטליות



אימות: מאמת את זהות השולח

תקינות נתונים: מבטיח שהנתונים לא נגעו

אי-התכחשות: מונע מהשולח להכחיש שחתם על הנתונים



# רכיבי תעודה דיגיטלית

נושא: מידע זהוטי (שם, ארגון, דוא"ל, וכו')

מפתח ציבורי: המפתח הציבורי של היישות להצפנה/אימות

מנתח: הרשות המנפקת וחותמת את התעודה

תקופת תוקף: תאריך התחלה ותוקף התעודה

חתימה דיגיטלית: החתימה הדיגיטלית של הרשות על נתוני התעודה

# יצירת תעודת דיגיטל

יצירת זוג מפתחות ציבורי/פרטי

יצירת בקשת חתימת תעודה (CSR) עם פרטי זיהוי ומפתח ציבורי

הגשת הבקשה לרשות המנפקת לאימות וחתימה

הרשות מנפקת את התעודה הדיגיטלית וחותמת אותה לאחר האימות



```
from cryptography import x509
from cryptography.x509.oid import NameOID
from cryptography.hazmat.primitives import hashes
from cryptography.hazmat.primitives.asymmetric import rsa
from cryptography.hazmat.backends import default_backend
from datetime import datetime, timedelta

# Generate a private key
private_key = rsa.generate_private_key(
    public_exponent=65537,
    key_size=2048,
    backend=default_backend()
)

# Create a self-signed certificate
subject = issuer = x509.Name([
    x509.NameAttribute(NameOID.COUNTRY_NAME, u"US"),
    x509.NameAttribute(NameOID.STATE_OR_PROVINCE_NAME,
u"California"),
    x509.NameAttribute(NameOID.LOCALITY_NAME, u"San
Francisco"),
    x509.NameAttribute(NameOID.ORGANIZATION_NAME, u"My
Company"),
    x509.NameAttribute(NameOID.COMMON_NAME, u"mycompany.com"),
])
```

```
cert = x509.CertificateBuilder().subject_name(
    subject
).issuer_name(
    issuer
).public_key(
    private_key.public_key()
).serial_number(
    x509.random_serial_number()
).not_valid_before(
    datetime.utcnow()
).not_valid_after(
    datetime.utcnow() + timedelta(days=365)
).add_extension(
    x509.SubjectAlternativeName([x509.DNSName("localhost")]),
    critical=False,
).sign(private_key, hashes.SHA256(), default_backend())

# Write the certificate to a file
with open("certificate.pem", "wb") as f:
    f.write(cert.public_bytes(encoding=x509.Encoding.PEM))

print("Digital certificate generated and saved as certificate.pem")
```



# שימושים של תעודות דיגיטליות

גלישה מאובטחת ברשת (HTTPS)

חתימה והצפנה של דוא"ל

חתימת קוד

רשתות פרטיות וירטואליות (VPN)

חתימה על מסמכים

