**Ministerul Educației, Culturii și Cercetării**

**Universitatea Tehnică a Moldovei**

**Facultatea Calculatoare, Informatică și Microelectronică**

**Departamentul Ingineria Software și Automatică**

**Raport**

Lucrarea de laborator nr.3

Disciplina: Securitatea Informațională

Tema: Algoritmul DSA

**Efectuat**: st.gr. TI-202 Bunescu Gabriel

**Verificat**: asist. univ. Todus Alexandru

Chișinău 2023

**Scopul lucrării:**

Elaborarea unui algorit de semnale original, digital.

**Mersul lucrării:**

1. Studierea noţiunii de semnătură digitală
2. Domeniul de aplicare
3. Metode de calcul
4. Dezvoltarea unei aplicaţi ce utilizează semnătura digitală
5. **Studierea noţiunii de semnătură digitală**

Semnătura digitală este o componentă cheie în domeniul securității cibernetice și criptografiei, fiind utilizată pentru a asigura autenticitatea, integritatea și necontestabilitatea datelor electronice sau a documentelor digitale. Iată câteva noțiuni esențiale legate de semnătura digitală:

* Ce este o semnătură digitală: Semnătura digitală este o marcă electronică sau criptografică aplicată pe un document digital sau un mesaj electronic pentru a atesta faptul că documentul a fost semnat sau a fost creat de o entitate specifică (de exemplu, o persoană sau o organizație) și că acesta nu a fost modificat de la momentul semnării.
* Cum funcționează semnătura digitală: Semnătura digitală se bazează pe utilizarea unui algoritm criptografic asimetric, precum RSA, DSA sau ECDSA. Procesul implică două chei: o cheie privată, cunoscută doar de semnatar, și o cheie publică, disponibilă public. Mesajul sau documentul este semnat digital folosind cheia privată pentru a genera o semnătură digitală, iar apoi oricine poate verifica semnătura folosind cheia publică. Dacă semnătura este validă, acest lucru confirmă autenticitatea semnatarului și integritatea datelor.
* Autenticitate: Semnătura digitală asigură autenticitatea semnatarului. Când o persoană semnează digital un document, semnătura este legată de identitatea sa și de cheia privată asociată. Astfel, dacă semnătura este validată cu succes, poți avea încredere că documentul a fost creat de către semnatarul specific și nu a fost falsificat.
* Integritate: Semnătura digitală protejează integritatea documentului sau a mesajului. Orice modificare a conținutului documentului după semnare va face ca semnătura digitală să devină invalidă. Prin urmare, semnătura digitală furnizează o modalitate de detectare a modificărilor neautorizate.
* Necontestabilitate: Semnătura digitală furnizează un mecanism de necontestabilitate. Semnatarul nu poate nega ulterior semnarea documentului, deoarece semnătura digitală este legată de cheia privată a semnatarului și este matematic legată de documentul semnat.
* Utilizări frecvente: Semnăturile digitale sunt utilizate într-o gamă largă de aplicații, inclusiv în tranzacțiile financiare online, autentificarea utilizatorilor, semnarea electronică a documentelor legale, transmiterea de e-mail-uri securizate și multe altele.
* Norme și standarde: Există norme și standarde internaționale pentru semnături digitale, cum ar fi PKCS#7, X.509, și altele, care definesc formatele și protocoalele pentru semnăturile digitale.

Semnăturile digitale joacă un rol esențial în protejarea securității și încrederea în mediul digital, contribuind la prevenirea fraudelor, autentificarea utilizatorilor și asigurarea autenticității documentelor și a mesajelor electronice.

1. **Domeniul de aplicare**

Algoritmul DSA (Digital Signature Algorithm) este utilizat pentru semnături digitale și verificarea acestora într-o varietate de domenii, în special în criptografia și securitatea informatică. Aici sunt câteva dintre domeniile de aplicare ale DSA:

* Securitatea comunicațiilor: DSA este utilizat în protocoale de securitate precum SSL/TLS pentru autentificarea serverelor și clienților în conexiunile securizate la internet. Acesta asigură că informațiile trimise și primite între două părți sunt autentice și nu au fost alterate în tranzit.
* Semnarea electronică: DSA este folosit pentru semnarea electronică a documentelor și contractelor în mediul digital. Prin semnarea electronică, semnatarul poate să-și ateste identitatea și să confirme că documentul nu a fost modificat de la momentul semnării.
* Autentificarea utilizatorilor: DSA poate fi utilizat pentru autentificarea utilizatorilor în sistemele securizate. Utilizatorii pot semna digital cereri de autentificare, iar sistemul poate verifica semnăturile pentru a confirma identitatea lor.
* Criptare și integritatea datelor: DSA poate fi folosit pentru a proteja datele prin asigurarea integrității acestora. O semnătură digitală poate fi atașată unor date pentru a verifica dacă acestea au fost modificate după semnare.
* Securitatea rețelelor de socializare: DSA poate fi utilizat în platformele de socializare pentru a autentifica și valida mesajele și postările utilizatorilor, evitând astfel falsificările și manipulările.
* Semnarea de software și actualizări de firmware: DSA poate fi folosit pentru a semna digital software-ul și actualizările firmware-ului, asigurând astfel că acestea provin de la dezvoltatorii legitimi și nu au fost modificate sau compromise de terțe părți.
* Securitatea e-mail-ului: DSA poate fi utilizat pentru a semna electronic e-mail-urile, asigurând că mesajele nu au fost modificate și pentru autentificarea expeditorilor.
* Sisteme de vot electronic: DSA poate fi utilizat pentru a asigura autenticitatea și integritatea voturilor în sistemele de vot electronic.
* Securitatea datelor financiare: În industria financiară, DSA este utilizat pentru a autentifica și a se asigura că tranzacțiile și transferurile de fonduri sunt legitime și sigure.
* Securitatea IoT (Internet of Things): DSA poate fi utilizat pentru autentificarea și semnarea datelor în dispozitivele IoT pentru a proteja informațiile și pentru a preveni manipularea acestora.

DSA oferă o modalitate sigură de a verifica autenticitatea și integritatea datelor în mediul digital și este folosit într-o gamă largă de aplicații care necesită securitate și încredere în datele și comunicările electronice.

1. **Metode de calcul**

Algoritmul DSA (Digital Signature Algorithm) implică mai multe etape în procesul de semnare digitală și verificare a semnăturilor digitale. Iată o descriere a principalelor metode și calcule implicate în DSA:

1. Generarea de chei:
2. Calculul primei și a doua numere prime mari, p și q.
3. Generarea unei chei secrete (privată), x, unde 0 < x < q.
4. Calculul valorii cheii publice (public\_key), y, bazată pe x, p, și q: y = (g^x) mod p, unde g este un generator al unui subgrup (1 < g < p).
5. Semnarea digitală:
6. Selectarea unui număr aleatoriu k, unde 0 < k < q.
7. Calculul punctului (x1, y1) = (g^k mod p, y^k mod p).
8. Calculul valorii s = (k^-1 \* (H(m) + x \* r)) mod q, unde H(m) este rezultatul unei funcții de hash aplicată mesajului m și r este coordonata x1 a punctului (x1, y1).
9. Semnarea constă din perechea (r, s).
10. Verificarea semnăturii:
11. Primirea semnăturii (r, s) și a cheii publice (y).
12. Calculul inversului lui s în modul q pentru a obține s^-1.
13. Calculul valorii w = s^-1 mod q.
14. Calculul valorilor u1 = (H(m) \* w) mod q și u2 = (r \* w) mod q.
15. Calculul punctului (x1, y1) = ((g^u1 \* y^u2) mod p) mod q.
16. Verificarea semnăturii constă în compararea r cu x1. Semnătura este validă dacă r este egal cu x1.
17. Calculul valorii hash a mesajului:
18. Pentru semnarea și verificarea semnăturii, mesajul m este trecut printr-o funcție de hash (de exemplu, SHA-256 sau SHA-512) pentru a obține un rezultat hash fix de lungime.
19. Generarea de numere aleatoare:
20. Algoritmul DSA implică generarea de numere aleatoare pentru a asigura că semnăturile sunt unice și imprevizibile.
21. Operații aritmetice modulare:
22. În întregul proces DSA, se efectuează multe operații aritmetice modulare, cum ar fi ridicarea la putere modulo p sau q și calculul inversului modular.

Este important de menționat că calculul DSA necesită manipularea cu numere mari și se bazează pe proprietățile matematice ale aritmeticii modulare. Implementările corecte ale DSA ar trebui să aibă grijă de gestionarea cu atenție a numerelor mari și a operațiilor matematice pentru a asigura securitatea și corectitudinea semnăturilor digitale.

1. **Dezvoltarea unei aplicaţi ce utilizează semnătura digitală**

Algoritmului DSA (Digital Signature Algorithm) pentru a crea și verifica semnături digitale pentru un mesaj. Iată o scurtă descriere a ceea ce face acest cod:

* Se generează o pereche de chei DSA, o cheie privată și o cheie publică, cu o dimensiune de 1024 de biți. Cheia privată este utilizată pentru a semna digital mesajele, în timp ce cheia publică este utilizată pentru a verifica semnăturile.
* Cheia privată și cheia publică sunt afișate sub formă de șiruri de caractere PEM (Privacy-Enhanced Mail) pentru a le putea folosi mai târziu.
* Utilizatorul introduce un mesaj de la tastatură, care este apoi convertit într-un șir de octeți (bytes).
* Mesajul original este apoi trecut printr-o funcție de hash SHA-256 pentru a calcula un rezultat hash fix de lungime.
* Mesajul este semnat digital folosind cheia privată și algoritmul de hash SHA-256, rezultând o semnătură digitală.
* Semnătura este verificată folosind cheia publică și același algoritm de hash. Dacă semnătura este validă, se afișează un mesaj că semnătura este validă; în caz contrar, se afișează un mesaj că semnătura nu este validă.
* La final, sunt afișate hash-urile mesajului original și ale semnăturii digitale.

Acest cod demonstrează cum puteți utiliza biblioteca cryptography pentru a implementa semnături digitale și pentru a verifica integritatea și autenticitatea mesajelor. Semnăturile digitale sunt utile în securitatea cibernetică și criptografie pentru a asigura comunicarea și stocarea datelor într-un mod securizat și autentic.

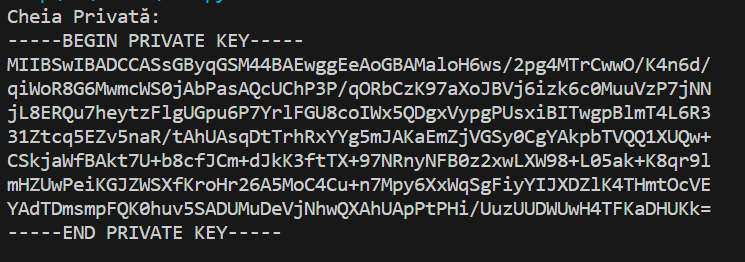


Figura 1 – Afișarea cheii private.

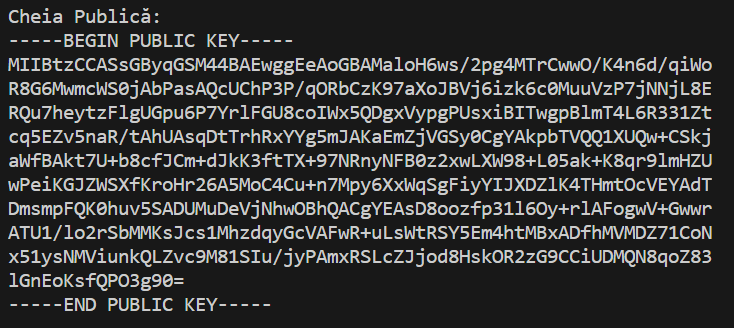


Figura 2 – Afișarea cheii publice.

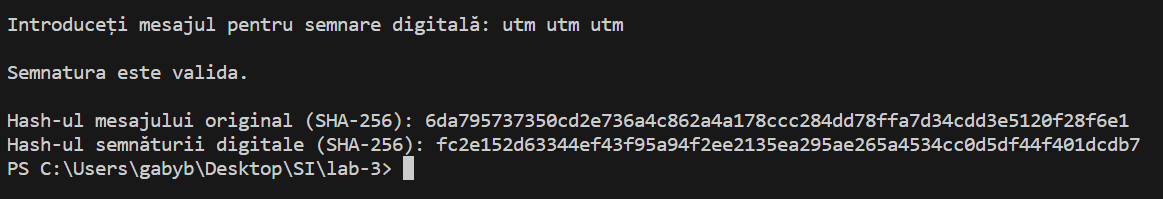


Figura 3 – Afișarea hash-ul mesajului original și digital.

**Concluzie:**

Algoritmul DSA (Digital Signature Algorithm) este unul dintre algoritmii criptografici asimetrici utilizați pentru semnături digitale. Procesul DSA implică generarea de chei, semnarea mesajelor și verificarea semnăturilor. Acesta se bazează pe operații matematice modulare și funcții de hash pentru a asigura securitatea semnăturilor.

În dezvoltarea aplicației noastre, am creat o pereche de chei DSA, am semnat digital un mesaj și am verificat semnătura folosind cheia publică. Afișarea hash-urilor mesajului original și ale semnăturii digitale ne permite să vizualizăm integralitatea datelor și validitatea semnăturii.

Semnăturile digitale sunt o componentă vitală a securității digitale și au o gamă largă de aplicații practice. Într-o lume digitală în continuă creștere, încrederea în autenticitatea și integritatea datelor este esențială, iar semnăturile digitale oferă un mecanism puternic pentru atingerea acestui obiectiv.

**Bibliografie:**

Codul sursă [17.09.2023] Disponibil:

https://www.devglan.com/online-tools/rsa-encryption-decryption

**Anexa:**

**import** hashlib

**from** cryptography.hazmat.backends **import** default\_backend

**from** cryptography.hazmat.primitives **import** hashes

**from** cryptography.hazmat.primitives.asymmetric **import** dsa

**from** cryptography.hazmat.primitives **import** serialization *# Importăm serialization*

**from** cryptography.exceptions **import** InvalidSignature

*# Generăm o pereche de chei DSA (cheie privată și cheie publică)*

private\_key = dsa.generate\_private\_key(key\_size=1024, backend=default\_backend())

public\_key = private\_key.public\_key()

*# Afișăm cheia privată*

private\_key\_pem = private\_key.private\_bytes(

encoding=serialization.Encoding.PEM,

format=serialization.PrivateFormat.PKCS8,

encryption\_algorithm=serialization.NoEncryption()

)

**print**("Cheia Privată:")

**print**(private\_key\_pem.decode())

*# Afișăm cheia publică*

public\_key\_pem = public\_key.public\_bytes(

encoding=serialization.Encoding.PEM,

format=serialization.PublicFormat.SubjectPublicKeyInfo

)

**print**("**\n**Cheia Publică:")

**print**(public\_key\_pem.decode())

*# Citim mesajul de la tastatură*

message = input("Introduceți mesajul pentru semnare digitală: ").encode()

*# Calculează hash-ul mesajului original (SHA-256)*

message\_hash = hashes.Hash(hashes.SHA256(), backend=default\_backend())

message\_hash.update(message)

message\_digest = message\_hash.finalize()

*# Semnăm mesajul folosind cheia privată*

signature = private\_key.sign(

message\_digest,

hashes.SHA256()

)

*# Verificăm semnătura folosind cheia publică*

**try**:

public\_key.verify(

signature,

message\_digest,

hashes.SHA256()

)

**print**("**\n**Semnatura este valida.")

**except** InvalidSignature:

**print**("**\n**Semnatura nu este valida.")

*# Afișăm hash-urile mesajului și ale semnăturii*

**print**(f"**\n**Hash-ul mesajului original (SHA-256): {message\_digest.hex()}")

**print**(f"Hash-ul semnăturii digitale (SHA-256): {hashlib.sha256(signature).hexdigest()}")