CRIPTOGRAFIE APLICATĂ

drnd. progr. Cătălin Untea
dr. ing. Laura Ciocoiu

Institutul Național de Cercetare Dezvoltare în Informatică

Rezumat: Proiectul **Criptografie aplicată** prezintă un model de criptare - decriptare, bazat pe cheie publică și cheie privată. Aplicația lucrează la nivel de server și la nivel de client. La nivelul server-ului se realizează generarea cheilor publică și privată, utilizînd algoritmul RSA; pe baza cheilor se realizează criptarea mesajului indicat direct sau electronic. Mesajul criptat împreună cu cheia privată sunt transmise "clientului". La nivelul clientului, mesajul este decriptat pe baza cheii private.

Cuvinte cheie: criptare, decriptare, cheie secretă, cheie publică, cheie privată, funcții hash de criptare, https.

1. Introducere

Schimbările care apar în infrastructura tehnologiei comunicației, alterează modul în care comunicăm. O dată cu beneficiile rezultate din creșterea vitezei, eficienței și a costurilor scăzute, "era digitală" a adus noi provocări pentru securitatea comunicațiilor și a informațiilor care străbat infrastructura globală de comunicație. Drept urmare, pe măsură ce tehnicile de criptare se diversifică, linia de marcaj între ceea ce s-a făcut și ceea ce nu s-a făcut a devenit neclară. Astăzi, criptografia poate fi sintetizată ca fiind suma metodelor și a aplicațiilor care depind de existența unui grad de dificultate în rezolvarea problemelor.

2. Algoritmi de criptare – prezentare generală

O metodă de criptare - decriptare este denumită cifru (cipher). Unele metode criptografice se bazează pe secretul algoritmului; astfel de algoritmi sunt numai de interes istoric, și nu sunt adecvați necesităților din lumea reală. Toți algoritmii moderni utilizează o cheie pentru a controla criptarea și decriptarea; un mesaj poate fi decriptat doar dacă cheia se potrivește cheii de criptare.

Există două clase de algoritmi: **simetrici** (bazați pe cheie secretă) și **asimetrici** (bazați pe cheie publică). Diferența este că algoritmii simetrici utilizează aceeași cheie pentru criptare și decriptare, în vreme ce algoritmii asimetrici utilizează chei diferite pentru criptare și decriptare, iar cheia pentru decriptare nu poate fi derivată din cheia de criptare.

Algoritmii simetrici pot fi grupati în algoritmi cu cifru secvențial (**stream ciphers**) și algoritmi cu cifru în bloc (**block ciphers**). Algoritmii cu cifru secvențial pot cripta un singur bit de text, în vreme ce algoritmii cu cifru în bloc pot cripta mai mulți bits (de exemplu 64 bits) ca o sigură entitate.

Algoritmii asimetrici permit cheii de criptare să fie publică (ea poate fi publicată în ziare), oricine putând să cripteze cu aceasta, în vreme ce destinatarul (care cunoaște cheia de decriptare) poate decripta mesajul. Cheia de criptare este denumită cheie publică, iar cheia de decriptare este denumită cheia privată.

În mod normal, algoritmii simetrici se execută mult mai rapid decât decât cei asimetrici.

În practică, ei sunt folosiți, de cele mai multe ori, împreună, astfel încât un algoritm de chei publice este utilizat pentru generarea aleatoare a unei chei de criptare, iar cheia aleatoare este utilizată să cripteze mesajul printr-un algoritm simetric. Această metodă este denumită **criptare hibridă**.

Algoritmii utilizați pentru generarea cheilor sunt: algoritmi de cheie publică, algoritmi de cheie secretă (cifru simetric), algoritmi de tip Cipher, funcții de criptare Hash, generatori de numere aleatoare.

Algoritmi de chei publice: folosesc chei diferite pentru criptare și decriptare. În plus, cheia de decriptare nu poate fi derivată (în practică) din cheia de criptare.

- **RSA (Rivest-Shamir-Adelman)** este **cel mai folosit algoritm de criptare, bazat pe chei publice**. El este considerat a fi sigur în situația în care cheile sunt suficient de lungi: cheile de 512 bits sunt nesigure, 768 bits moderate ca siguranță, iar 1024 bits reprezintă chei bune; cheile de 2048 bits se consideră că vor rămâne sigure în decadele care vor urma. Securitatea din algoritmul RSA se bazează pe dificultatea de a calcula factorialul unor numere întregi foarte mari. Acest lucru face din RSA cel mai important algoritm de chei

publice. Trebuie ştiut că algoritmul RSA este vulnerabil prin aşa numitele atacuri "chosen-plaintext attack" şi "timing attack". Implementări ale algoritmului RSA disponibile sunt: RSAREF, RSAEURO, SSLeay, PGP, Ssh, Crypto++;

- **Diffie-Hellman** este un algoritm bazat pe chei publice, a cărui securitate se bazează pe dificultatea problemei logaritmului discret ("discret logarithm problem"). El este considerat a fi sigur atunci când sunt utilizate chei suficient de lungi şi generatoare de chei adecvaţi. Mărimea exponentului secret este, de asemenea, importantă pentru securitatea oferită. De aceea, exponentul aleator trebuie să fie de două ori mai mare decât cheia dorită. Există şi modalităţi de spargere a criptării prin aşa numitul "timing attack". Implementări ale algoritmului Diffie-Hellman sunt: RSAREF, RSAEURO, SSLeay, alodes, Crypto++;
- **DSS (Digital Signature Standard)**. Algoritmul prezintă numeroase puncte slabe: dezvăluirea cheii secrete, în cazul în care se criptează două mesaje cu acelaşi număr aleator, o protecţie nu foarte bună a datelor criptate. Implementare a algoritmului: Crypto++;
- **Algoritmul cheii publice ElGamal**. Se bazează pe problema logaritmului discret ("discrete logarithm problem"). Implementare a algoritmului: Crypto++;
- **Algoritmul cheii publice LUC.** Implementare a algoritmului: Crypto++;

Algoritmii Cheii Secrete (Cifru Simetric): folosesc aceeaşi cheie pentru criptare şi decriptare;

- **IDEA(International Data Encryption Algorithm)** utilizează o cheie de 128 bits şi este considerat a fi sigur. Până în prezent, nu a fost semnalat nici un atac asupra acestui algoritm în ciuda numeroaselor încercări de analiză a lui. Implemantări ale algoritmului IDEA: SSLeay, PGP, Ssh, idea86, Crypto++;
- **RC4.** Algoritmul este foarte rapid şi, deşi are un mod de lucru cunoscut, decriptarea acestuia este un lucru la îndemâna oricui. **RS4** este un generator de numere pseudoaleatoare, a cărui ieşire este supusă operaţiei XOR cu un şir de caractere. De accea, este foarte important: cheia **RC4** să nu fie folosită la criptarea a două şiruri diferite. Implementări ale algoritmului: SSLeay, Crypto++, Ssh;
- **SAFER** este considerat că furnizează o criptare sigură, chiar şi pe procesoare de 8 bits;
- **Cifru bazat pe funcţii hash**. Orice funcţie de criptare, suficient de puternică, poate fi transformată într-un cifru. Ideea este de a folosi funcţia hash pentru a genera numere aleatoare, iar ieşirea este supusă unei operaţii XOR cu informaţia care se doreşte a fi criptată. Implementări ale algoritmului: MDC/SHA;
- **Enigma** este cifrul utilizat de Germania în timpul celui de-al doilea război mondial. Este uşor de spart folosind calculatoarele moderne;

Modele de criptare în bloc (Block Cipher Modes): Mulţi dintre algoritmii pentru chei secrete (IDEA, DES, BLOWFISH) sunt metode de criptare în bloc. Acest lucru presupune că ele iau un bloc fix de date (de exemplu 64 bits) şi îl transformă într-un bloc de 64 bits folosind o cheie.

Funcţii hash de criptare:

- **MD5 (Message Digest Algorithm 5)** este folosit pentru a trunchia un şir de orice dimensiune la o valoare de 128 bits. Implementari ale algoritmului: PGP, Ssleay, RSAREF, Crypto++, Ssh;
- **MD2, MD4:** implementări mai vechi ale algoritmului MD5. Implementări ale algoritmului: Ssleay, RSAREF;
- **SHA (Secure Hash Algorithm)** produce o valoare de 160 bits dintr-un mesaj de lungime arbitrară. Este considerat un algoritm destul de bun.

Generatoarele de numere aleatoare: algoritmii criptografici au nevoie de numere aleatoare, care să nu fie ghicite de către atacatori. Numerele aleatoare sunt utilizate pentru generarea cheilor, de aceea calitatea lor este critică pentru calitatea rezultatului. Unele maşini pot dispune de hardware special, genarator de zgomot. Perturbaţiile de la scurgerile de curent ale unei diode sau tranzistor, biţii cei mai puţin semnificativi ai intrărilor audio, timpii dintre intreruperi etc. sunt surse de numere aleatoare dacă sunt procesate cu o funcţie hash. Implementări ale generatoarelor de numere aleatoare sunt: PGP, Noiz, Ssh.

3. Aplicaţii ale algoritmilor de criptare

Proiectul **Criptografie aplicată** prezintă un model de criptare – decriptare, bazat pe cheie publică şi cheie privată, utilizînd algoritmul RSA.

3.1. Descrierea algoritmului RSA

Denumirea de RSA vine de la inţialele celor trei cercetători Ron Rivest, Adi Shamir şi Len Adleman care l-au inventat în 1977 la Massachusetts Institute of Tehnology (MIT). Securitatea oferită de algoritm provine din dificultatea calculării de numere prime de dimensiuni mari. Procesul presupune selectarea a două numere prime **p** si **q** (formate din sute de cifre) şi înmulţirea lor pentru a obţine un număr **n**. Aceste numere sunt trecute printr-un algoritm matematic de determinare a cheii publice **KU={e,n}** şi a cheii private **KR={d,n}**, chei care se găsesc într-o relaţie matematică. Este extrem de grea determinarea factorului **e** şi **d** pornind de la **n**, ceea ce oferă securitate algoritmului.

Odată generate aceste chei, mesajele pot fi criptate în blocuri şi trecute prin următoarea ecuaţie:

$$C = M^e \bmod n$$

unde **C** este testul criptat, **M** este textul necriptat iar **e** este cheia publică a celui care recepţionează mesajul.

Analog, mesajul de mai sus poate fi decriptat folosind următoarea ecuaţie:

$$M = C^d \bmod n$$

unde **d** este cheia privată a celui care recepţionează mesajul.

3.2. Crypto

În continuare, este prezentată aplicaţia **Crypto** (figura 1) care realizează:

- la **nivel de server:** generarea cheii publice şi cheii private utilizînd algoritmul RSA, criptarea şi/sau decriptarea mesajelor utilizînd cheia publică şi cheia privată astfel generate, transmiterea cheii private şi a mesajului criptat la nivel de client, prin intermediul e-mail sau a unui server client de mail.
- iar, la nivel de client decriptarea mesajului

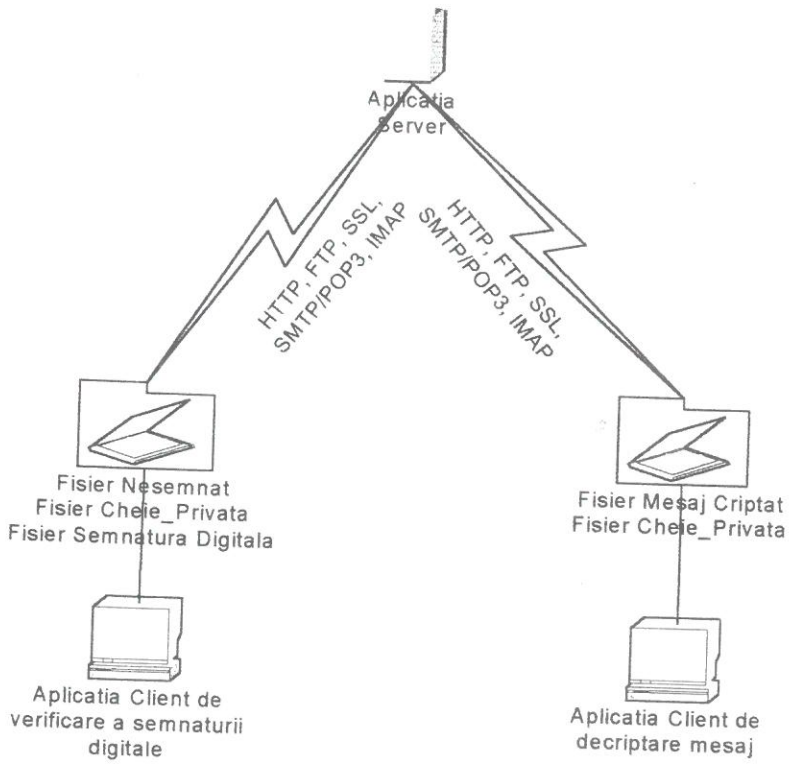


Figura 1. Aplicaţia criptare decriptare

3.2.1. Prezentarea aplicatiei CryptoRun (SRV) – la nivel de server

La nivelul server-ului se realizează generarea cheilor şi, pe baza cheilor, criptarea mesajului către client, mesaj indicat electronic sau direct prin înscrierea acestuia în fereastra corespunzătoare.

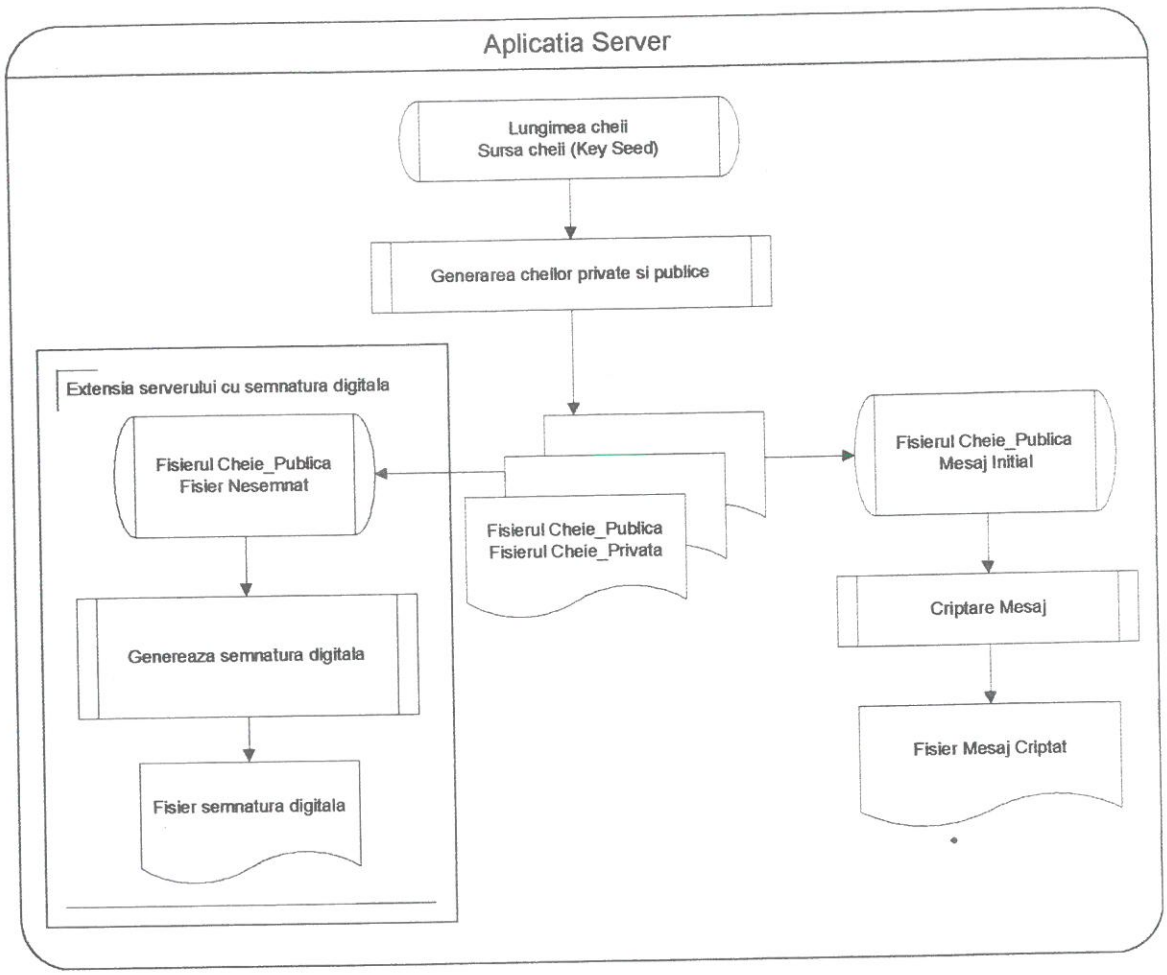


Figura 2. Aplicaţia CryptoRun (SRV)

Aplicaţia **CryptoRun (SRV)** (figura 2) constă în:

a) generarea cheii publice şi cheii private utilizînd algoritmul RSA

b) criptarea şi/sau decriptarea mesajelor utilizînd cheia publică şi cheia privată astfel generate

c) transmiterea cheii private şi a mesajului criptat la client, prin intermediul e-mail sau a unui server client de mail.

Un exemplu de implementare a algoritmilor de criptare (la nivel de server) este arătat în figura 3.

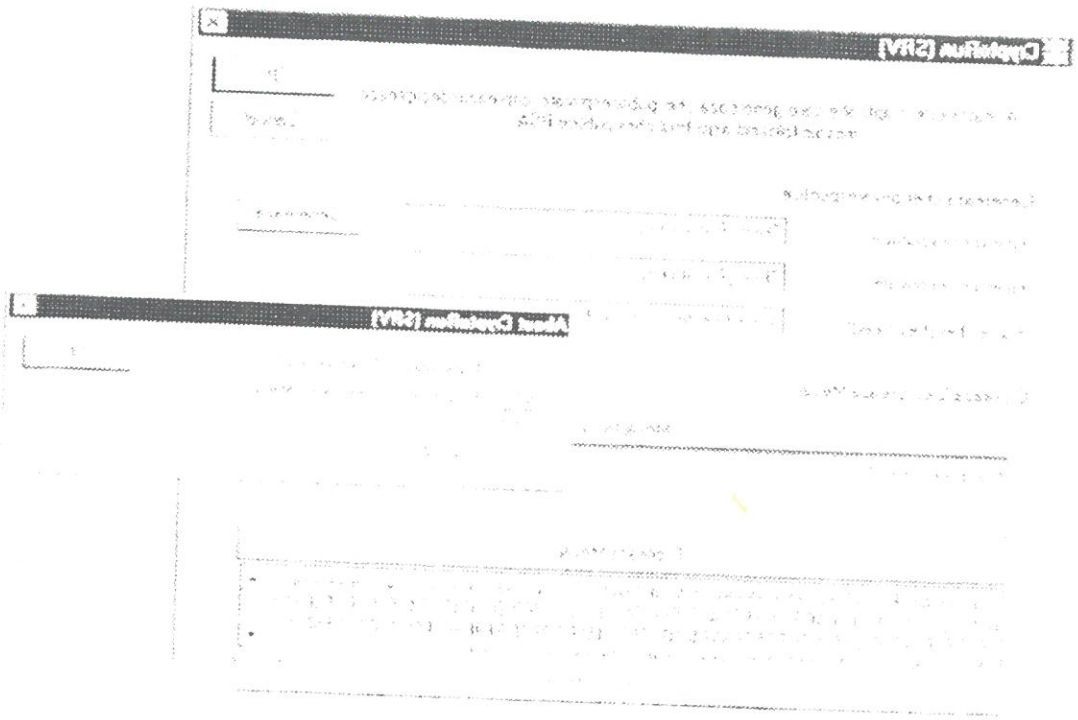


Figura 3. Aplicaţia de criptare CryptoRun(SRV) (la nivel de server)

3.2.2. Generarea cheilor

Perechile de chei – privată şi publică sunt generate de algoritmul RSA utilizînd **sursa cheii** (key seed) care este o frază indicată de utilizator ce va sta la baza generării perechii de chei. Cheile privată şi publică sunt generate ca un şir de de ceractere hexa fără înţeles; ele sunt stocate în fişiere indicate de utilizator. Astfel, la fraza **Aceasta este sursa cheii** indicată pentru sursa cheii, sunt generate cheia publică şi cheie privată, stocate în fişierele specifice.

În urma executării programului CryptoRun(Srv) *<secţiunea generarea chei>*, avînd ca sursa a cheii textul *Aceasta este sursa cheii*, cheia publică şi cea privată au fost generate astfel :

- **Cheia privată** din fişierul "Cheie_Privata.key"

308202740201003000D06092A864886F70D0101010500048202 5E3082025A02010002818100B19288E8A65A7
AEF442E64069E16C5BD29D545F49F3545185AD6A4EE440FA2053461C491C781E70D20428701FB5C39C
2F67F507994A1FBAF531292C84699647EC79DED2AFA1D207A8DDB90958D58BEEDC11DD4E03EF5942
AB5244A04DB56653A77047D6E8C9E43CF13A3B381EEB5D370F0846A36319808EFD2DFA4AB05FF651
B0201110281800A72080DAF6EBBEFF4F3AB87EB2E841A2FA3223B90E504107DD0640E0400EB69B7C98
4089343E06A2F12F8E1FFBA2183F061D78EAE63E1A0E6C4DB75315451348925CDEC15DA813DC46A23
957CEFD0645820820525A967CF4ACB75F5FB8BC1104089002455F6E9C45C92CDD835CDC80D0474C627
36327E44017DCFF85EB5EA11024100C312B03C9B54C17B701AD5C7D8EC3C64FD327EA7002DA2BBA4
77DB51948A3A81C86D4A40212A54E1B8DDA616212ACF91A412D3124533C7C4E6B2F7B271292F51024
100E9089040EB45C9E512B45DE0687FABDEE9C2B3E1BE880CAB1929995D928458A4657E30C4B60E68
E3350662103AE0BB0200B06E895309DEA6D2D1DD7A4AC1AAAB0240226CB5B057A58B8E40F5AD416
2840AA868EACB0E697177120DF70895ED09559E6EA9DFED33077864119086B89C70F7740DE5344E84A
EC8E6831086106E528FE102410096C94E481FC3BEDF8492D354F8529C630FBA3828A876265097CF9F78C
83784A69C065BCA93EB34CF406D8ABF35280088D344FC3ABD429F20A687CB8B5D8C5F5F0240454D4
DFE56321AC79016E32F09A2D270B6845DAEFDA26831640DD259367FCA6A7CA9A4A4F7A844AEE1256
E8E2458CABCAE34C07DB08FBB2203F25F08D76FACC7

- **Cheia publică** din fişierul "Cheie_Publica.key"

30819D300D06092A864886F70D01010105000381B003081870281810OB19288E8A65A7AEF442E64069E1
6C5BD29D545F49F3545185AD6A4EE440FA2053461C491C781E70D20428701FB5C39C2F67F507994A1FB
AF531292C84699647EC79DED2AFA1D207A8DDB90958D58BEEDC11DD4E03EF5942AB5244A04DB566
53A77047D6E8C9E43CF13A3B381EEB5D370F0846A36319808EFD2DFA4AB05FF651B020111

3.2.3. Criptarea/Decriptarea Mesajelor

Mod de lucru: Mesajul de criptat este indicat, direct sau din fişier, în fereastra corespunzătoare. La apăsarea butonului "Cripteaza Mesaj" se generează pe disc un fişier text "Crypt_File.txt" care cuprinde mesajul criptat. Acelaşi mesaj criptat este vizibil şi in fereastra principala. La apasarea butonului "Cripteaza Mesaj", câmpul "Fisierul cheii publice" trebuie completat deoarece aceasta cheie este folosită in criptarea mesajului.

Un exemplu de criptare este indicat mai jos :

- Mesaj inainte de criptare

  Maria are mere.

- Mesaj după criptare - În urma executării programului CryptoRun(Srv) *<secţiunea criptare mesaj>*, 9BECFE4AEBC537F871B5936E6B0DCA10FD5AAC17C9437903DFD1B2D94773BF6B98BC6A
  5849AE4BCCEF2C4E9C396B214A226ED54F31352A27E32B0074D524B515CFE55FB1018ED61F1B37
  F0F550CFAEC5C69A84A9B925D3451ED9ECDBF8B681BD911C07D55D3ECC1AC0B15A7E11743214
  ADC8C2B852DA9E3C457F575689F4C50A

Pentru verificarea mesajului criptat, programul dispune de opţiunea de decriptare la nivel de server, prin apăsarea butonului "Decripteaza Mesaj".

Mesaj decriptat: **Maria are mere**.

Fişierul ce conţine mesajul criptat **Crypt_File.txt** şi fişierul cheii private **Cheie_Privata.key** sunt transmise destinatarului, prin server de mail, pentru aplicaţia client CryptoRun (CL).

3.3. Prezentarea aplicatiei CryptoRun (CL) (Figura 4)– la nivel de client

La nivelul clientului, sunt primite fişierele cu mesajul criptat **Crypt_File.txt** şi cel ce conţine cheia privată **Cheie_Privata.key**. Mesajul este decriptat pe baza cheii private.

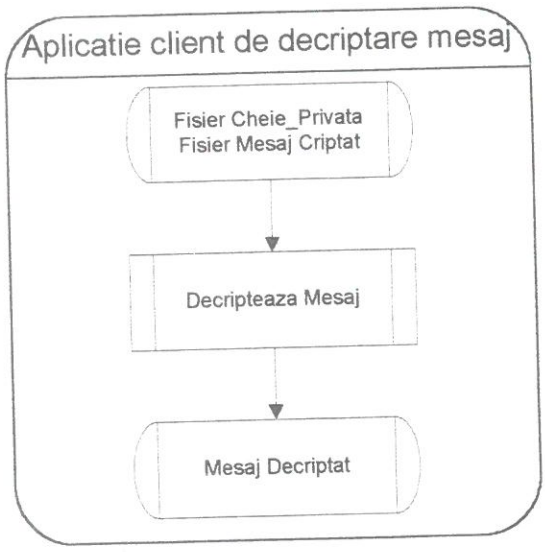


Figura 4. Aplicaţia CryptoRun (CL)

Mod de lucru (figura 5): Aplicaţia primeşte, drept intrare, numele fişierului criptat şi numele fişierului cheii private. La apăsarea butonului **Decriptează!,** mesajul criptat este citit din fişierul Crypt_File.txt şi mesajul decriptat este afişat în fereastra corespunzătoare.

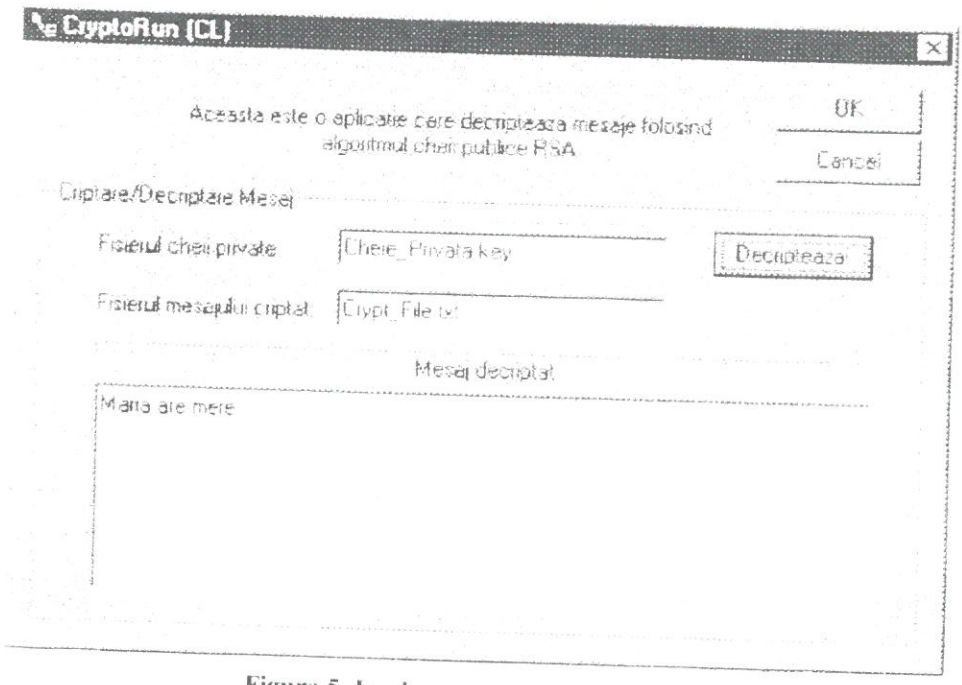


Figura 5. Implementare Aplicaţia CryptoRun (CL)

Într-o dezvoltare ulterioară, se doreşte realizarea unei aplicaţii pe WinSocket care va permite comunicarea între aplicaţiile client-server pe portul SSL (443) sau a unei aplicaţii client-server de mail (SMTP, POP3, IMAP4). De asemenea, se doreşte extinderea aplicaţiei de client-server pentru transferul de fişiere, atât pentru http(port 80) cât şi pentru ftp (port 20).

4. Semnătura electronică

Aplicaţia de criptare este necesar a fi continuată şi extinsă pentru transmiterea de documente criptate însoţite de semnătură electronică. Semnăturile digitale sunt folosite pentru a verifica faptul că mesajul provine din partea expeditorului declarat.

Semnătura digitală a unui document este creată prin sumarizarea sau condensarea documentului, urmată de concatenarea rezultatului cu informaţii despre emitent. Următorul pas constă în criptarea rezultatului cu cheia privată a emitentului folosind algoritmul de criptare RSA. Informaţia criptată rezultată este **semnătura**. Ea este trimisă de cele mai multe ori impreună cu informaţia despre cheia publică folosită pentru semnare. Pentru verificare, destinatarul trebuie să determine dacă cheia aparţine persoanei care afirmă că este proprietarul cheii, după care urmează decriptarea semnăturii utilizînd cheia publică. Dacă semnătura este decriptată corect iar rezultatul decriptării se potriveste mesajului (care este supus aceluiaşi proces de sumarizare sau condensare, etc.), semnătura este acceptată ca fiind validă.

Semnăturile digitale pot fi folosite şi pentru a **certifica** faptul că o cheie publică aparţine unei anumite persoane. Acest lucru se realizează prin semnarea cheii publice şi a informaţiei privitoare la proprietar cu o cheie sigură, de către o terţă parte care reprezintă autoritatea de certificare. Semnatura digitală, cheia publică şi informaţia despre proprietarul cheii publice sunt denumite **certificate (certificates)**.

În continuare vom face o scurtă prezentare a aplicaţiei **CryptoRun (Semnat) (Figura 6)** care are 3 secţiuni distincte şi anume : generarea cheilor publice şi private, criptare / decriptare mesaje şi generarea semnăturii electronice. Primele două secţiuni au fost dezvoltate la punctul 2.

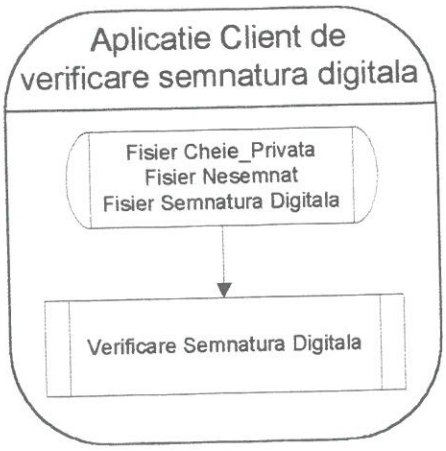


Figura 6. Aplicaţia CryptoRun (Semnat)

Mod de lucru (figura 7) Pentru generarea semnăturii electronice, la **nivel server**, se specifică în câmpul **Fisierul sursa** numele fişierului pentru care se doreşte generarea semnăturii şi, de asemenea, **Fisierul destinatie** care reprezintă semnătura electronică astfel generată. În continuare, se alege butonul **Generează Semnatură Electronică** şi se apasă butonul **Execută**!.

La **nivel client** este recepţionat atât documentul sursă cât şi fişierul ce conţine semnătura electronică. Tot la nivel client, se face verificarea semnăturii. Pentru testele iniţiale, verificarea semnăturii este realizată la nivel de server.

(Pentru generarea semnăturii electronice, trebuie indicat **fişerul cheii publice**.)

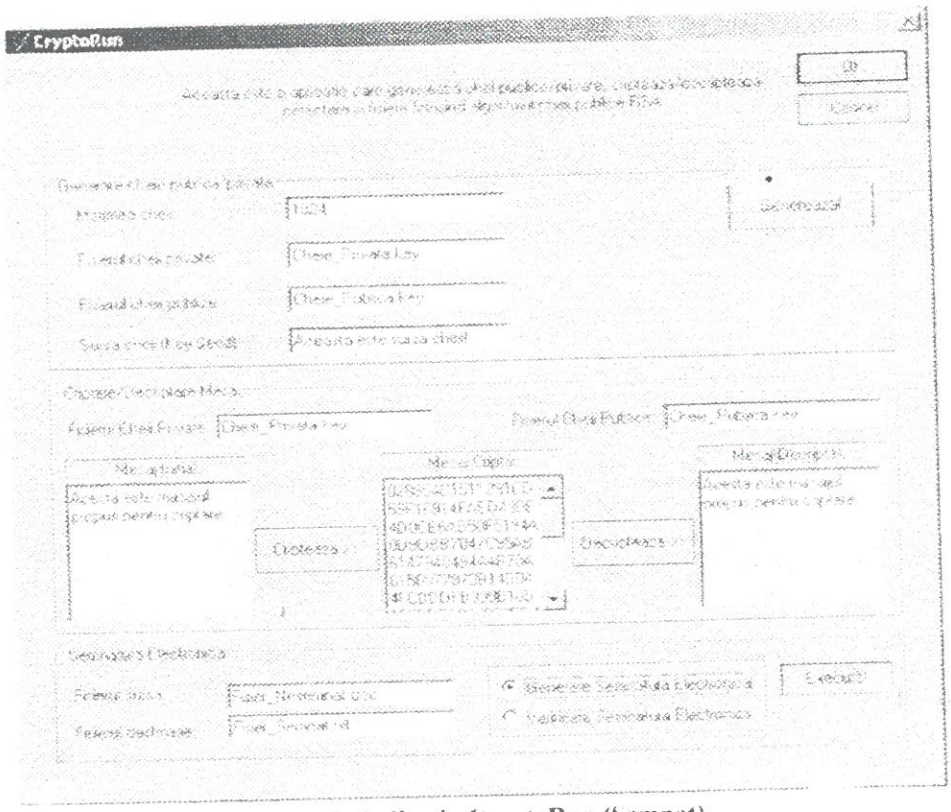
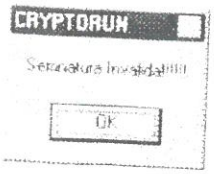


Figura 7. Aplicaţia CryptoRun (Semnat)

Fişierele astfel generate se trimit destinatarului care poate verifica semnătura prin aplicarea asupra semnăturii a cheii sale priyate. Daca fişierele trimise au fost alterate sau au suferit erori la transmiterea datelor, la verificare, destinatarul, va primi mesajul "Semnatura Invalida!!!!".



Conținutul **Fisier_Nesemnat.txt**

- **MD5 (Message Digest Algorithm 5)** este folosit pentru a trunchia un șir de orice dimensiune la o valoare de 128 biți. Implementări ale algoritmului: PGP, Ssleay, RSAREF, Crypto++, Ssh.
- **MD2, MD4:** implementări mai vechi a algoritmului MD5. Implementări ale algoritmului: Ssleay, RSAREF.
- **SHA (Secure Hash Algorithm)** produce o valoare de 160 biți dintr-un mesaj de lungime arbitrară. Este considerat un algoritm destul de bun.

Conținut **Fisier_Semnat.txt** (semnătura electronică)

9BB749AAF10F82F3ADE23D5B16B3501DE2807C4F95B4711DD65A71F3B35C5EED702DE8AE5D452
46EF1961536200639C40287706955339912 2C809D9579773A5F4F5A687081D649BEFB19D19080AAA62B7
3CC29271AB3D5A503DB437ABD56DA1485800D4A40E2236ABB30B3A2E7B80AC88B6DD1EC9AD863C
DF1CC4866D106815D

5. Concluzii

Societatea informatizată implică securitatatea informațiilor transmise prin intermediul Internetului.. Utilizarea protocoalelor de securizare moderne au transformat Internet-ul într-o rețea sigură de tip WAN. Internetul a introdus termeni noi cum ar fi http ca varianta securizată **https.**

O aplicație de securizare a informațiilor transmise prin Internet o constituie **criptarea**. Sistemele de criptare cu cheie simetrică sau cele cu cheie publică au fost perfecționate și standardizate în ideea asigurării confidențialității și garantarea integrității datelor transmise, asigurarea autentificării destinatarului și a nerespingerii mesajelor la acesta. Securizarea documentelor în Internet este o problemă majoră atât la nivelul grupurilor de standardizare cât și la nivelul marilor companii.

Proiectul **Criptografie aplicată** a fost dezvoltat în cadrul temelor de cercetare A23/2001 și Relansin R87/1999 finanțate de MEC.

Bibliografie

1. **STĂNESCU, I.**: Infrastructura națională de chel puplice , tema A23/2001.
2. **BĂNICĂ, L. CIOCOIU, C. UNTEA**: Sistem experimental pentru documente electronice, Pitești, iulie 2002.
3. *** * ***: ABA – Digital Signature Guidelines
4. *** * ***: Digital Signature Standard (DSS) – http://csrc.nist.gov/publications
5. *** * ***: http://www.ssh.com/ SSH Communications Security
6. *** * ***: http://archives.math.utk.edu/topics/computerAlgebra.html
7. *** * ***: http://theory.lcs.mit.edu/~rivest/crypto-security.html

8. * * *: http://www.mirrors.wiretapped.net/security/cryptography/algorithms/rsa/

9. * * *: http://www.rsasecurity.com/rsalabs/faq/3.html

10. **KRAWCZYK, H., M.BELLARE, R.CANETTI**: HMAC Keyed Hashing for Message Authentification, 1997–http://www.ietf.org/rfc

11. **RIVEST, R.**: The MD5 Message Digest Algorithm, 1992.