Proiect IC

Transmisie de secret folosind steganografie audio, AES & RSA

Autor: **Martinescu Sorin-Alexandru**

Cuprins:

1. **Introducere in steganografie** ..... **pag. 3**
2. **Steganografia audio** ..... **pag. 4**
3. **Introducere in criptografie** ..... **pag. 5**
4. **AES** ..... **pag. 6**
5. **RSA** ..... **pag. 7**
6. **Metoda LSB** ..... **pag. 8**
7. **Metoda LSD** ..... **pag. 10**
8. **Metoda AM** ..... **pag. 12**
9. **Descrierea dialogului de transmisie securizata** ..... **pag. 16**
10. **Dezvoltarea pasilor de comunicare** ..... **pag. 17**
11. **Concluzii** ..... **pag. 18**
12. **Bibliografie** ..... **pag. 19**

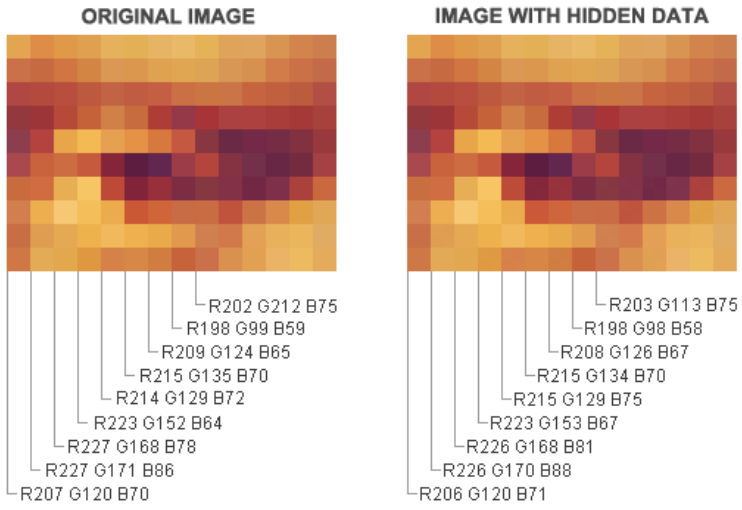
1. Introducere in stenografie

Stenografia este un termen derivat din limba greaca, din cuvintele stegos si graphia, si înseamnă scriere ascuns. Este arta si știința ascunderii comunicației. In esența, este o exploatare a simțurilor umane.

Folosind stenografia, un mesaj secret poate fi integrat într-o informație nesuspicioasa (informație pe care o vom numi purtătoare), fără ca cineva sa-si dea seama de existenta mesajului secret. Intre stenografie si criptografie este o legătură strânsă. Cele doua științe au același obiectiv de a secretiza comunicațiile, dar in timp ce criptografia asigura confidențialitatea mesajelor prin codificare, stenografia ascunde mesajele in informații oarecare (ce nu ar atrage atenția); in anumite situații stenografia fiind mai avantajoasa deoarece un mesaj codat poate trezi suspiciuni, pe când un mesaj invizibil poate trece neobservat.

Principiile definite de Kerckhoffs pentru criptografie sunt valabile si pentru stenografie: calitatea unui sistem criptografic ar trebuie sa depindă in proporții mici de informația folosita; aceasta proprietate este valabila si pentru sistemele stenografice de calitate: informația sistemului utilizat, nu ar trebui sa ofere informații despre existenta unui mesaj ascuns.

Pentru o mai buna siguranță a informației ascunse, aceasta înainte de a fi integrate in purtătoare, este criptata; însă in exemplele ce vor urma sa fie prezentate, s-a sărit peste etapa de criptare, deoarece nu reprezintă o etapa de interes in ilustrarea metodelor de stenografie.



2. Stenografia audio

Stenografia audio – reprezintă o clasa de metode prin care se ascund informații in semnale audio. Folosirea unui semnal audio ca purtătoare pentru mesajul secret este o sarcina mult mai complicata, decât utilizarea unei purtătoare imagine, deoarece sistemul vizual al omului este mai puțin senzitiv comparat cu sistemul auditive.
Stenografia audio este caracterizata de 3 mari parametrii: transparenta, robustețea si capacitatea.
Transparenta impune ca purtătoarea ce conține informația (denumita si fișier stego), sa nu difere perceptual fata de purtătoarea nealterata.
Robustețea măsoară capabilitățile datelor integrate in purtătoarea audio de a rezista in urma unor atacuri intenționate sau neintenționatei.
Capacitatea reprezintă cantitatea de informație ce poate fi integrata fără a altera vizibil purtătoarea, astfel încât un observator sa nu sesizeze prezenta unor date ascunse. In cazul purtătoarei audio, capacitatea se refera la cantitatea de informație ce poate fi ascunsa. Se măsoară in procente, sau chiar in biți / sec. de semnal audio.
Presupunere: purtătoarea audio are formatul .wav, iar lungimea trebuie sa fie de x ori mai mare ca lungimea mesajului (in cazul in care mesajul secret este un mesaj audio) sau numărul de biți ai purtătoarei audio trebuie sa fie mai mare ca numărul de biți ai mesajului (in cazul in care mesajul secret este un text). (valoarea x depinde de metoda de ascundere folosita)

Pentru implementarea stenografiei audio, am abordat 3 metode diferite: metoda LSB, o metoda pe care am sa o numesc metoda LSD (o metoda pe care am dezvoltat-o, plecând de la idea propusa de metoda LSB) si o metoda bazata pe modulația in amplitudine.

3. Introducere in criptorgrafie

Un instrument trebuie văzut în primul rând în cadrul contextului din care face parte, criptografia reprezintă instrumentul de bază în domeniul mai larg al securității informației. Într-un secol în care informaţia este indispensabilă, asigurarea securităţii acesteia devine o preocupare de prim rang. Aceasta se datorează faptului că informaţia este lipsită de valoare atâta timp cât atributele ei de securitate nu sunt asigurate. În mare, securitate înseamnă protecţie în faţa unei potenţiale ameninţări iar în ceea ce priveşte informaţia ameninţările pot varia de la simpla alterare neintenţionată a acesteia până la accesarea de către persoane neautorizate sau distrugerea ei.

Criptografia este definită ca fiind studiul tehnicilor matematice referitoare la aspecte de securitatea informaţiei precum confidenţialitate, integritate, autentificarea entităţilor, autentificarea provenienţei datelor. Totuşi o astfel de definiţie nu este completă. Pe de o parte deoarece criptografia nu este în totalitate matematică (chiar dacă marea ei parte este), de exemplu criptarea cuantică face mai mult apel la cunoştinţe de fizică decât de matematică sau implementarea criptografiei ţine mai mult de ştiinţa calculatoarelor decât de matematică. Pe de altă parte pentru că nu ţine cont de fondul problemei. Ron Rivest a făcut o remarcă pe cât de simplă pe atât de profundă în ceea ce priveşte criptografia şi această remarcă poate fi considerată o excelentă definiţie a criptografiei: criptografia înseamnă comunicare în prezenţa adversarilor. Orice comentariu la adresa remarcii lui Rivest este superfluu.

4. AES

La nivelul anilor 2001 DES nu mai oferă securitatea necesară (de fapt încă din anii 90 sunt consemnate atacuri de succes asupra DES), pentru care, pe bază de concurs se alege un nou standard AES (Advanced Encryption Standard).
Standardul curent este candidatul la AES numit Rijndael ales din cei 5 finalişti: Rijndael, Serpent, Twofish, RC6 şi MARS.
AES este un cod bloc disponibil în trei variante de dimensiuni pentru cheie 128, 192, 256. Chiar şi cheia de 128 de biţi este considerată destul de sigură pentru cerinţele din ziua de azi. Necesită doar 10-14 runde în funcţie de dimensiunea cheii, este sigur şi este cel mai rapid dintre candidaţi. Deoarece AES este mai rapid decât alte coduri simetrice, chiar şi decât 3DES, şi oferă cel puţin acelaşi nivel de securitate nu există nici un motiv de a utiliza altceva decât AES în arhitecturi de securitate contemporane. AES nu foloseşte structura Feistel, are meritul de a fi un criptosistem inovator.
El procesează matrici de 4x4 bytes prin intermediul a 4 transformări:

- AddRoundKey – se adună cheia de rundă printr-un simplu XOR,
- SubBytes – se substituie fiecare byte prin intermediul unei tabele de look-up (substituţie neliniară),
- ShiftRows – se shiftează circular (rotire) fiecare linie astfel: prima linie e neatinsă, a 2-a linie 1 la stânga, a 3-a cu 2 la stânga şi a 4 cu 3 la stânga,
- MixColumns – se amestecă coloanele prin aplicarea unei transformări de această dată liniară şi reversibilă (de fapt este vorba de multiplicare matricială).

Totuşi trebuie să precizăm că singura suspiciune cu privire la securitatea AES-ului este faptul că foloseşte un design destul de non-conformist, spre deosebire de schemele simetrice clasice, care se construiesc pe reţea Feistel. Acest design nu a fost sub atenţia comunităţii criptologilor decât în ultimii ani, de la propunerea AES-ului.
În mod spectaculos, transformarea AES (Rijndael) este echivalentă cu o ecuaţie algebrică destul de simplă (comparativ cu alte coduri) faţă de care există suspiciunea că ar putea duce în viitor la o serie de atacuri.

5. RSA

RSA este prima realizare concretă de algoritm de criptare asimetrică şi semnătura digitală. Acest algoritm se bazează pe utilizarea pentru criptare a funcţiei f (x) = pow(x, ε) mode n, unde n este un întreg compozit produs a două numere prime iar ε este un exponent întreg care respectă c.m.m.d.c (ε, ϕ (n)) = 1. Această funcţie este o bijecţie şi admite ca inversă funcţia pow(x, δ) mod n care va fi utilizată la decriptare, δ este un întreg care satisface relaţia εδ ≡1modϕ(n) . Desigur că inversarea acestei funcţii este posibilă dacă se cunoaşte factorizarea lui n . Singura cale cunoscută de a sparge complet sistemul RSA este factorizarea modulului, reamintim însă că nu există nici o demonstraţie că aceasta este singura metodă de a sparge complet RSA-ul. Adică, nu există nici o demonstraţie cu privire la echivalenţa dintre RSA şi problema factorizării întregilor IFP. Mai mult, recent s-a instalat mult scepticism cu privire la echivalenţa între securitatea RSA şi problema factorizării întregilor odată cu apariţia articolului lui Boneh şi Venkatesan.

Este însă demonstrat că a calcula o pereche de chei RSA (cheie publică şi cheie privată) este echivalent cu problema factorizării întregilor.

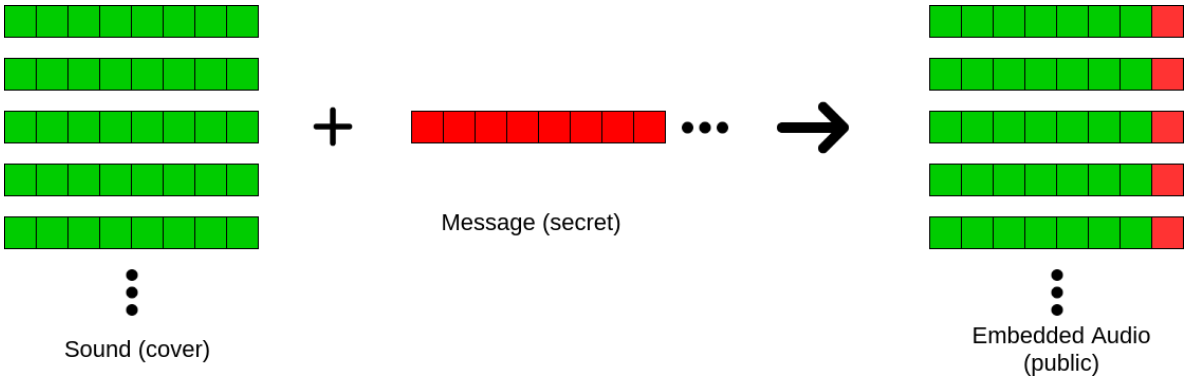
Alegerea corectă a parametrilor din stadiul de iniţializare a cheii este extrem de importantă. Pentru a evita atacuri prin factorizarea modului n se recomandă utilizarea unui modul de 1024-4096 biţi pentru o securitate pe termen lung (vezi tabelul din secţiunea introductivă). Alegerea celor două numere prime p şi q este critică pentru securitate. Este recomandabil ca numerele să fie alese astfel încât să aibă acelaşi număr de biţi iar p – q să fie suficient de mare pentru a preveni un atac exhaustiv prin căutarea unui factor mai mic decât sqrt(n) .

Pentru a spori rezistenţa în faţa atacurilor prin factorizare a fost propusă şi varianta RSA nebalansat în care factorii nu au număr egal de biţi. Aceasta este mult mai rezistentă la factorizare dar are o vulnerabilitate fatală în faţa unui atac de tip criptotext ales care v-a fi discutată mai jos. De asemenea exponentul public poate fi ales ca având forme speciale pentru a face criptarea mai eficientă. Sunt preferaţi exponenţii care au cât mai puţini biţi de 1, aceasta datorită algoritmului de exponenţiere care consumă mai mult timp când bitul exponentului este 1. Din acest motiv între exponenţii preferaţi pentru criptarea RSA sunt numerele 3, 17 şi 65537. În mod cert RSA este criptosistemul cu cheie publică cel mai intens studiat (nu în ultimul rând acest lucru se datorează simplităţii sale). Astfel, de-a lungul timpului o gamă relativ largă de atacuri şi vulnerabilităţi ale RSA au fost publicate.

6. Metoda LSB

Este una din primele metode de ascundere a informației, o metoda ușor de implementat; numele LSB este o abreviere a leat signifiant bit, ce sugerează si modul de lucru: integrează fiecare bit al mesajului in cel mai nesemnificativ bit al cadrelor purtătoarei audio. Când ultimul bit este modificat in cadrele purtătoarei audio, va apărea si un zgomot; daca zgomotul introdus nu este detectabil, stenografia a avut loc cu succes. (de cele mai multe ori, zgomotul este prezent si detectabil)

Cu cat numărul de biți ai mesajului este mai mic, cu atât transparenta este mai ridicata. Pentru metoda LSB robustețea si capacitatea nu pot coexista. Daca se alege capacitatea ca parametru principal, va rezulta o robustețe scăzută, astfel mesajul putând fi ușor extras din purtătoare.



Pentru a implementa metoda LSB am ales o purtătoare de tip .wav si am utilizat uneltele oferite de limbajul python; (pentru a demonstra doar partea de stenografie, am eliminat etapa de criptare a mesajului). Implementarea consta in următoarele etape:

- se extrag cadrele din fișierul .wav
- se va converti mesajul in format binar
- pentru primele n cadre de informație se vor efectua următoarele operații(un n reprezintă numărul de biți ai mesajului):
  - se generează o masca de biți (masca de biți are LSBul 0 iar restul de biți sunt 1)
  - se va aplica masca de biți pentru fiecare cadru prin operația SI logic
  - se va aplica operația SAU logic intre un bit al mesajului si cadrul obținut in urma aplicării măștii de biți

Avantaje:

- ușor de implementat

Dezavantaje:

- se ia in considerație robustețea scăzută, faptul ca este sensibila la manipularea fișierului, mesajul poate fi extras ușor.
- prezenta zgomotului, care de foarte multe ori este sesizabil.
- Numărul de biți al mesajului trebuie sa fie mai mic sau egal cu numărul de cadre ale fișierului .wav, de aceea mesajul nu trebuie sa fie foarte lung.

Observație:

Pentru a elimina zgomotul, in primul rând trebuie ca numărul de biți al mesajului sa fie considerabil mai mic raportat la numărul de cadre ale fișierului .wav (estimativ vorbind, ar trebui ca numărul de cadre sa fie de minim 4 ori mai mare decât numărul de biți ai mesajului).

De asemenea ar fi recomandat sa se lucreze pe un fișier .wav ce dispune de cel puțin 2 canale, pentru o aerisire a biților.

Abordare gândita de mine este următoarea:

- Se va calcula o valoare de increment = (numărul de cadre al purtătoarei – 5) / (numărul de biți ai mesajului).
- Daca incrementul este cel puțin 4, se va introduce in primul cadru in locul valorii curente valoarea incrementului, daca nu se va încheia procesul de integrare a mesajului secret.
- Pornind de la al șaselea cadru, vom memora valoarea poziție cadrului curent si pe canalul stâng al cadrului de la valoarea poziției curente vom aplica masca de biți si apoi voi integra un bit de mesaj; in canalul drept al cadrului de la valoarea poziție curente + 2 se va integra următorul bit de mesaj, apoi poziția curenta va fi = poziția curenta + valoarea incrementului, si procesul se va repeta pana când toți biții de mesaj au fost integrați;

Recuperarea este relativ ușoara, si aceasta metoda are ca avantaj principal eliminarea zgomotului sesizabil, si as putea spune ca un avantaj suplimentar îl reprezintă o creștere a robusteții (mica, dar relativ considerabila la atacurile neintenționate).

7. Metoda LSD

Este metoda propusa de mine (utilizata de asemena in securizarea comunicatiei), dezvoltata pornind de la ideea metodei LSB.
LSD este abreviere a least significant digit si a fost gândita pentru integrarea de mesaje audio in purtătoarea noastră, fără a pierde din calitatea semenului audio integrat.
Principala constrângere a acestei metode este ca raportul dintre (numărul de cadre ale purtătoarei – 5) si (numărul de cadre ale mesajului nostru audio) sa fie cel puțin 6.
Încă o constrângere ar fi ca purtătoarea sa dispună de cel puțin 2 canale.
De asemenea, am impus in cadrul implementării ca formatul mesajului audio sa fie .wav.

Implementarea este relativ similara cu implementarea observației aduse metodei LSB, numai ca in cazul curent se va modifica ultima zecimala dintr-un cadru al purtătoarei, nu LSBul.
Pasul 1:

- Se verifica daca purtătoarea audio nu este mono;
- Se va calcula raportul (numărul de cadrele ale purtătoarei – 5) / (numărul de cadre ale mesajului) si se va verifica daca este mai mare sau egal cu 6.

Daca pasul 1 este încheiat cu succes, se va putea trece la integrarea mesajului secret in purtătoare.
Pasul 2:

- Se va converti mesajul audio in format mono (daca nu este deja);
- Se vor citi complet, in format numeric (int16), cele doua semnale audio (purtătoarea si mesajul);

  Obs.: din purtătoare vom folosi, pentru a integra mesajul, doar doua canale, pe care le vom denumi: canal stâng si canal drept (aceste canale vor fi primele doua din purtătoarea extrasa, identificate prin […,0] si […,1], conform sintaxei python3)
- Din valorile extrase din purtătoare, vom salva in 2 vectori valorile pentru canalul stâng si pentru canalul drept; (vom nota cei doi vectori vs si vd, corespunzători pentru canalul stâng si respectiv pentru canalul drept)
- Pe prima poziții a vectorilor vs si vd se va afla valoarea = (numărul de eșantioane ale mesajului audio) / 1000;

• Pe ce-a de-a doua poziție a celor doi vectori vs is vd se va afla valoarea = (numărul de eșantioane al mesajului audio) % 1000;

• Începând cu poziția a 5 v-a începe integrarea propriu zisa astfel:

- o Se ia eșantionul curent din mesaj ce se urmează sa fie integrat in purtătoare, si-i vom memora valoarea in variabila e.
- o Se va memora poziția curenta in vectorii vs si vd, si o vom nota x.
- o Algoritm:
  1. vs[x+5] = t, unde t poate fi 2 daca eșantionul nostru este pozitiv, 0 daca eșantionul nostru este negativ si 1 daca eșantionul este 0.
  2. vd[x+4] = e % 10 si e = e / 10
  3. vs[x+3] = e % 10 si e = e / 10
  4. vd[x+2] = e % 10 si e = e / 10
  5. vs[x+1] = e % 10 si e = e / 10
  6. vd[x] = e % 10 si x = x + increment
  7. Daca mai sunt eșantioane in mesaj neintegrate, repetam acest algoritm (ne întoarcem la pasul 1).
- o La finalul algoritmului, cei doi vectori, vs si vd, împreuna cu celelalte canale nealterate ale purtătoarei vor fi aduse la forma necesar salvării pentru a se putea genera noul fișier .wav.

Pentru a se profita la maxim de spațiul de integrare pus la dispoziție de purtătoare (in cazul in care purtătoarea are mai mult de 2 canale), se recomanda exploatarea celorlalte canale.

In cazul in care vrem ca mesajul integrat sa nu-si piardă din calitate, vom sari peste etapa de conversie a mesajului in semnal mono, dar va trebuie ca valoare de pe poziția a 3 a vectorilor ce conțin valorile canalelor, sa fi numărul de canale ale mesajului; de asemenea, se va impune o constrângere noua, ca raportul (numărul de cadre ale purtătoarei - 5) / (numărul de cadre ale mesajului) sa fie mai mare sa egal cu 6*c, unde c reprezintă numărul de canale ale mesajului.

Avantaje:

- o Mesajul ascuns in purtătoare va fi recuperat la un nivel de calitate similar cu mesajul adus la format mono.
- o Se poate integra un mesaj audio, fără a se pune problema pierderii informațiilor de ce au frecvente înalte.
- o Purtătoarea in urma integrării mesajului este lipsita de zgomot, comparat cu metoda LSB.

Dezavantaj:

- Mesajul recuperat este însoți de un ușor zgomot digital.(nu este deranjant si nici nu deteriorează informația din mesaj)
- Dimensiunea mesajului este relativ mica.

8. Metoda AM

Numele de AM vine de la modulația in amplitudine, ce reprezintă etapa principala a acestei metode. Metoda AM exploatează auzul uman: urechea umana, in medie, poate auzi sunete in intervalul 20Hz – 20 KHz, la nivel teoretic; practic vorbind, urechea poate auzi, in medie, sunete in intervalul 31Hz – 17.6KHz. (aceasta este prima informație ce permite stenografia audio ce utilizează modulația in amplitudine).

De asemeni, cam toate dispozitivele ce pot reda conținut audio au frecventa de eșantionare de 44100Hz, ceea ce înseamnă ca avem o banda de frecvente audio intre 0Hz si 22050Hz. (aceasta este a doua informație ce permite stenografia audio ce utilizează AM)
Se mai cunoaște ca urechea umana este foarte sensibila in intervalul 2KHz – 5 KHz.
De asemenea se, intervalul frecventelor audio pentru vocea umana este 85Hz – 255Hz.
(acestea sunt ultimele informații necesare implementării metodei de stenografie audio cu modulația in amplitudine)

Din informațiile prezentate anterior, se observa ca frecventele peste 17.5KHz nu sunt sesizabile de o ureche umana normala, iar ca frecventa maxima a purtătoarei audio poate fi de 22KHz. Din acestea observam ca avem o banda de frecvente de 4.5KHz (in intervalul 17.5KHz – 22KHz) ce nu poate fi perceputa de o ureche umana normala si știm ca frecventa maxima pentru care urechea umana este foarte sensibila este de 5KHz ne produce următoarea concluzie – putem filtra mesajul nostru audio cu un filtru trece jos (pana la frecventa de 4.5KHz) fără a pierde foarte mult din informația acestuia(mai exact, fără a pierde informație utila) si rezultatul filtrării sa-l deplasam in intervalul 17.5KHz – 22KHz, aici intervenind modulația in amplitudine – in urma acestui proces rezultând un semnal audio nesesizabil de o

ureche umana normala, ce conține mesajul nostru audio (mesaj filtrat). Aceasta operație, reprezintă chiar prima etapa a metodei AM.

Reprezentare spectrului unui mesaj oarecare:

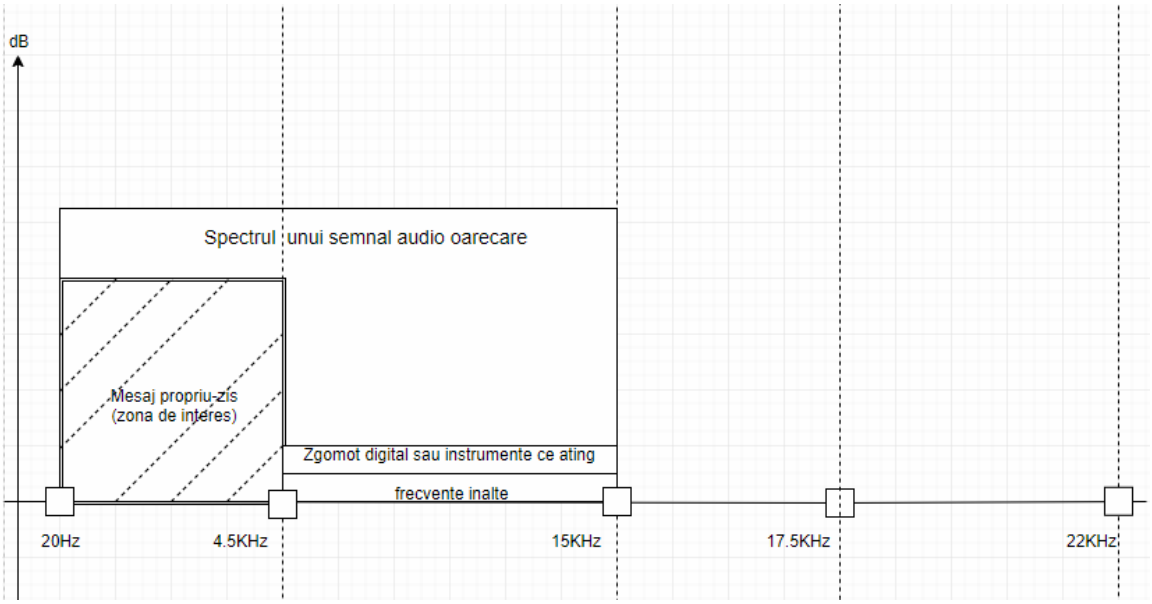
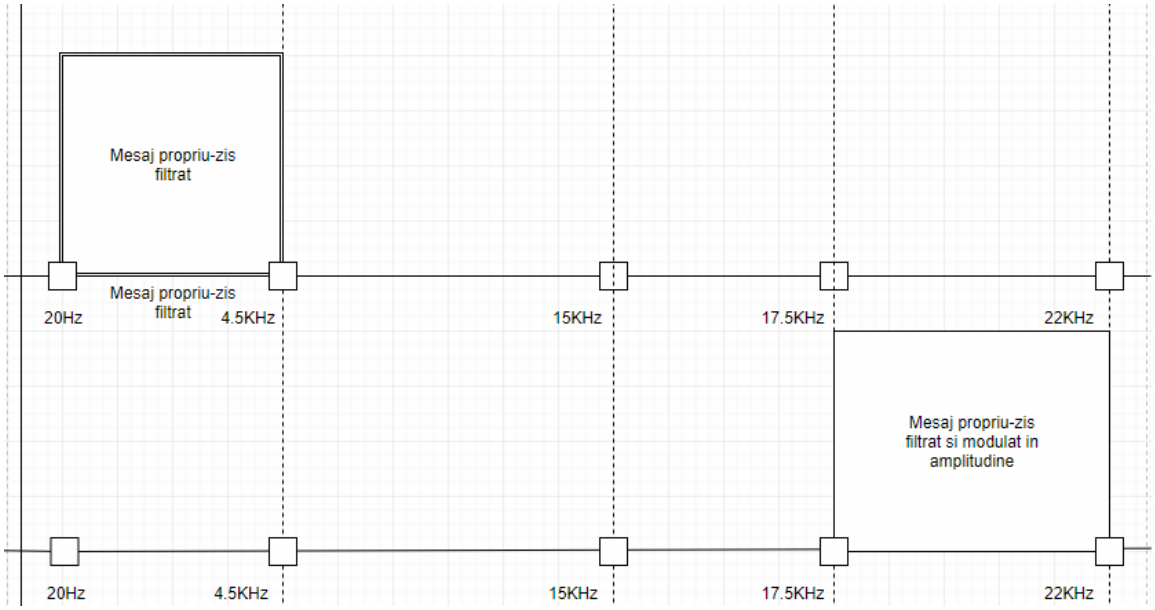


Fig. 1. Spectrul unu semnal audio oarecare

Modulația in amplitudine, este o etapa relativ de ușor de implementat, la nivel conceptual, implementarea are forma:

(Mesaj filtrat) * cos(2*π*22050*t) → Mesaj modula in amplitudine

După filtrare si modulația in amplitudine a mesajului, spectrul ar trebuie sa fie de forma:

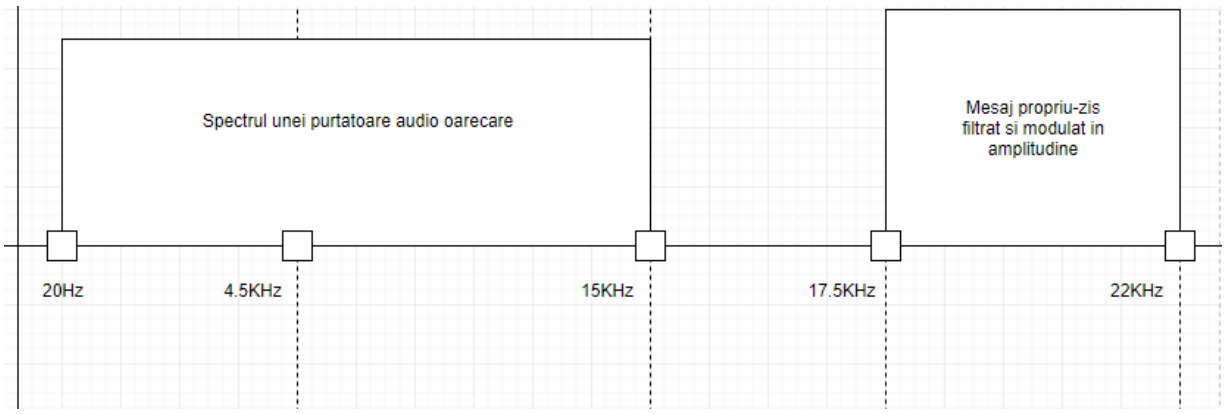


Ultima etapa a metodei AM o reprezintă adunarea mesajului filtrat si modulat in amplitudine cu purtătoarea noastre;

Constrângere: lungimea purtătoarei trebuie sa fie cel puțin la fel de mare ca lungimea mesajului.

Observație: purtătoarea noastră audio poate fi orice, o discuție intre persoane, un instrumental, etc; dar se observa ca in general informația utila a purtătoarei nu conține frecvente ce depășesc pragul de 15KHz, frecventele ce sunt peste acest prag fiind in mare parte doar zgomot digital.

In urma adunării purtătoarei cu mesajul filtrat si modulat ar trebuie sa se obțină in reprezentare spectrala:



Recuperarea mesajului are loc pe același principiu cu modulația in amplitudine, purtătoarea ce conține ceasul nostru este înmulțita cu aceeași funcție cos utilizata la deplasarea in frecventa a mesajului.

Se recomanda o filtrare a rezultatului pentru a diminua zgomotul introdus in urma deplasărilor in frecventa si a Avantaje:

- Mesajul integrat poate avea lungime maxima = lungimea purtătoarei audio
- Mesajul recuperat (fără a fi filtrat) este însoțit de un zgomot ce nu afectează calitatea mesajului, zgomot care poate fi sesizabil sau nu, in funcție de natura mesajului si sa purtătoarei; (in general zgomotul nu este sesizabil, sau este foarte ușor sesizabil)

Dezavantaje:

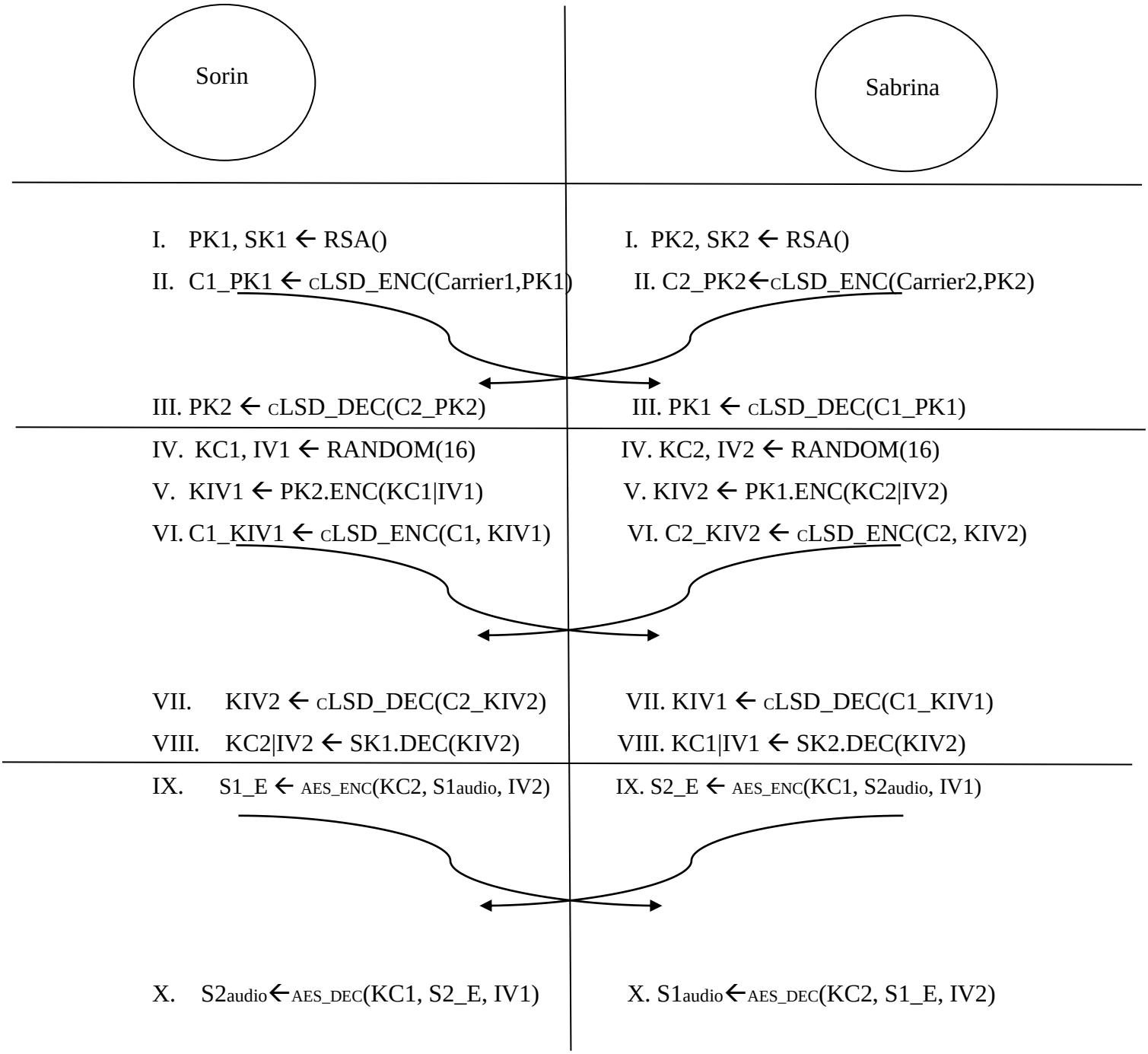
- Pentru a integra mesajul in purtătoare, acesta este filtrat, pierzând din calitate
- Metoda nu este foarte robusta, deoarece in urma unei analize spectrale a purtătoarei ce conține mesajul, se poate observa ușor o activitate suspicioasa in banda de frecvente pe care urechea umana nu o poate percepe. adunării mesajului cu purtătoarea.

Descrierea dialogului de transmisie securizata

9. Descrierea dialogului de transmisie securizata

Problematica de la care am plecat a fost: comunicarea unui secreat audio intre 2 persoane. Implementarea, realizata cu succes foloseste steganografie audio (metoda LSD), pentru a transmite Public Keyul (PK) generat de RSA si PK.encrypt(KC) (unde KC este cheia utilizata in ecriptia AES a secretlului audio).

Dialogul are forma:



10. Dezvoltarea pasilor de comunicare

La pasul II, cLSD_ENC, reprezinta o metoda similara cu LSD (ENC/DEC), in cod, indetificata prin denumirea de write_to_carrier.

Aceasta preia sirul de biti si la o singura iteratie a buclei while, din sirul de biti, de la pozitia ramasa, extrage 6 biti, si le face split in 2 grupuri de 3 biti, aceste doua grupuri sunt convertite in decimal si sunt integrate in carrier similar ca in metoda LSD (cu observatia, ca primul grup se va integra in canalul stang si celalalt grup, va fi integrat in canalul stang). cLSD_DEC functioneaza in mod revers functiei cLSD_ENC.

La pasul IX si X, AES_ENC is AES_DEC functioneaza astfel:

1. Secretul audio e facut single_channel
2. Sunt extrasele esantiaonele din secret si din ele se formeaza un string cu caracteru „|” ca separator intre esantioane si ca padding pentru a indeplini conditia necesara pentru encriptia AES(lungimea_sirului % 16 == 0)
3. Sirul obtinut e encriptat (am incercat doua varinate, ecriptie completa a stringului si ecriptia individuala a fiecarui bloc de 16 caractere; am ramas la versiunea in care face ecnriptie pe tot sirul)
4. Rezultatul encriptiei in encodez cu base64 si fiecare caracter al noului sir de caractere e transformat in decimal folosind functia ord() si decimala rezultata e integrata in noul audio frame
5. Avand audio frameul complet, il scriu si astfel rezulta secretul audio codata cu AES

Obsv.: AES_DEC functioneaza in mod invers pasilor descrisi mai sus.

11. Concluzii

Pentru comunicarea descrisa mai sus, am incercat initial sa utilizez RSA, dar outputul generat de acesta era greu de prelucrat, iar lungimea secretului criptat devenea inutilizabila.

Astfel am preferat sa criptez secretul audio folosind AES si sa transmit cheia de criptare si IVul folosit de AES via RSA. Astefl, pot spune ca rezultatul comunicatiei este destul de bun si sigur.

Pentru urmatoarea iteratie a implementarii, am de gand sa pastrez primele doua etape (cand se schimba intre terminalele canlului de comunicatie, PKul si KC|IVul) dar am sa schimb ultima etapa, printr-o imbunatatire: Am sa criptez secretul audio tot cu AES, cum am descris mai sus, doar ca outputul criptat nu va fi trimis in starea accea (pentru ca ar atrage atentia), va fi integrat intr-o purtatoare audio folosind metoda LSD, astfel comunicarea nu va atrage atentia nimanui. Pot preciza ca am incercat sa implementez aceasta noua iteratie a comunicatiei, doar ca, la nivel numeric, dupa aplicarea „carrier ← LSD_ENC(carrier, secret_enc_aes)” pentru integrarea secretului in purtatoare si aplicarea „secret_enc_aes ←LSD_DEC(carrier_enc)”, secretul encodat cu aes este alterat. Aceste probleme sunt rezultate ca urmare a extractiei secretului respectiv, rescrierii cadrului audio si o serie de operatii efectuate pe cadrul purtatoarei.

Observatie: pentru transmisia unor secrete audio, ce nu sunt critice, se pot folosi metodele LSD si AM fara nici o problema.

12. Bibliografie

1. Principles of Voice Production-Titze, I.R. (1994)
2. Primate Hearing From a Mammalian Perspective-RICKYE S. HEFFNER
3. Clinical Measurement of Speech and Voice. London: Taylor and Francis Ltd.-Baken, R. J.
4. Optimizarea si securitatea sistemelor de e-business – Rațiu Crina A.
5. https://ieeexplore.ieee.org/document/6798347
6. https://epxx.co/artigos/ammodulation.html
7. Introducere în Criptografie Funcții Criptografice, Fundamente Matematice și Computaţionale: Bogdan Groza