UNIVERSITETI I PRISHTINËS

FAKULTETI I SHKENCAVE MATEMATIKO-NATYRORE

Programi: Shkenca Kompjuterike

Punim Seminarik - Implementimi i Simplified AES dhe modelit OFB në Java **Lënda -** Siguria e të Dhënave

Studenti :

Egzonit Demhasaj

Gabriel Kolaj

Profesor :

Artan Berisha

Besnik Duriqi

Prill 2023, Prishtinë

Përmbajtja

1. Një prezantim i shkurtër i projektit

Në kuadër të projektit të dytë të lëndës Siguria e të Dhënave, kemi pasur si detyrë implementimin e algoritmit të enkriptimit **AES i thjeshtuar** (ang. **Simplified AES**) si dhe modelit **Output Feedback (OFB)** në gjuhën programuese Java, sipas skemës që na e ka dhënë profesori.
Programi të cilin kemi pasur për detyrë ta krijojmë ne, merr si input një text file dhe e enkripton atë duke përdorur Simplified AES të implementuar hap pas hapi nga vet ne studentët, e pastaj për arsyeje të testimit të saktësisë programi e bën edhe dekriptimin e file-it të enkriptuar, me ç'rast fitohet përmbajtja origjinale e file-it të cilin programi e ka marrë si input në fillim.
Me qëllim të kuptimit sa më të lehtë të funksionimit të programit tonë, ne do të përshkruajmë shkurtimisht se si realizohet enkriptimi/dekriptimi sipas Simplified AES.
Standardi i thjeshtuar i përparuar i enkriptimit (AES i thjeshtuar) është një shifër blloku me çelës simetrik që përdor një madhësi fikse blloku prej 16 bitësh, duke u bazuar në skemën që na e ka dhënë profesori, dhe një madhësi fikse çelësi prej 16 bitësh. Është një version i thjeshtuar i algoritmit të plotë AES, por gjithsesi ofron siguri të arsyeshme për shumë aplikacione.

Modeli i Feedback-ut në dalje (OFB) është një mënyrë funksionimi me shifra blloku që e kthen një shifër blloku në një shifër transmetimi. Në modelin OFB, dalja e mëparshme e shifrimit përdoret si hyrje për të enkriptuar bllokun tjetër të tekstit të dhënë (ang. **plaintext**) dhe dalja e shifrës XOR-izohet me tekstin e dhënë për të prodhuar tekstin e shifruar (ang. **ciphertext**). Ky proces përsëritet për çdo bllok në tekstin e dhënë.

Për projektin tonë, ne do të zbatojmë modelin e thjeshtuar AES dhe OFB në Java. Kjo do të përfshijë krijimin e klasave dhe metodave për të trajtuar gjenerimin, enkriptimin dhe deshifrimin e çelësave duke përdorur këto algoritme. Do t'na duhet gjithashtu të testojmë zbatimin tonë për t'u siguruar që ai është i saktë dhe i sigurt. Disa strategji të mundshme testimi mund të përfshijnë përdorimin e çifteve të njohura tekst të thjeshtë/shifror, testimin për ndjeshmërinë e çelësit dhe kontrollimin e dobësive kriptografike.

Në përgjithësi, projekti jonë do të kërkojë një kuptim solid të kriptografisë me çelës simetrik dhe programimit Java, si dhe vëmendje ndaj detajeve dhe testim të kujdesshëm për të siguruar që zbatimi jonë është i saktë dhe i sigurt. Për të kuptuar më shumë rreth programit që kemi krijuar ne, do të keni mundësi gjatë leximit në vijim, ku ju do të keni mundësinë të njoftoheni me arkitekturën e programit tonë, rolin e secilës metodë në klasat përkatëse, si dhe testimin se a funksionon progarmi jonë ashtu siç duhet apo jo. Andaj, ju ftojmë që të vazhdoni leximin e dokumentimit.

2. Arkitektura e Programit

Fillimisht duhet të ceki se në punimin e projektit tonë e kemi përdorur ariktekturën e programimit të cilën e kemi përdorur deri më tani edhe në ushtrimet tona, pra e kam përdorur **MVC (Model View Controller)** arkitekturën. Arkitektura e programit tonë është dhënë në figurën e mëposhtme :

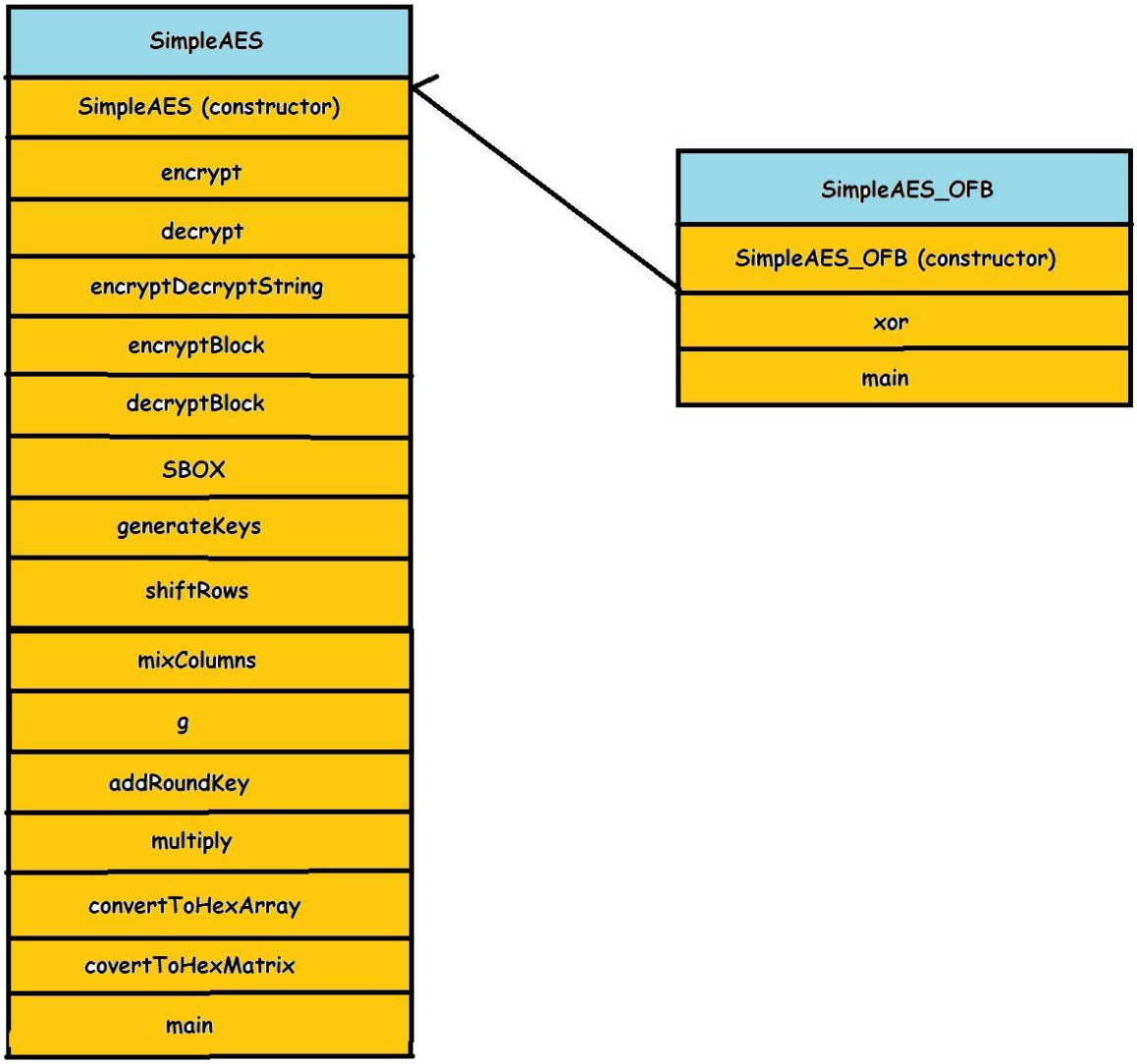

Figura 1. Arkitektura e aplikacionit

3. Dokumentimi Teknik (sqarimi i kodit burimor)

Problemin apo kërkesën e detyrës tone e kam shqyrtuar duke krijuar dy klasë, ashtu siç shihet edhe nga arkitektura e programit, ku njëra e ka emrin **SimpleAES**, e cila ka për detyrë implementimin e algoritmit AES i thjshtuar sipas skemës së profesorit, kurse tjetra e ka emrin **SimpleAES_OFB**, e cila implementon modelin **Output Feedback** të AES**.**

Klasa **SimpleAES** përmbanë metodën **main** për testim apo për ekzekutim të programit, pra thërret apo invokon metodat e krijuara tek kjo klasë, dhe gjithashtu në të përfshihen edhe disa modifikime të programit, për të cilat do të flasim më detajisht në vijim.

Kjo klasë përfshinë në vete **15** metoda, të cilat na mundësojnë implementimin e Simplified AES në gjuhën programuese Java. Normalisht që secila prej këtyre metodave ka përdorimin e saj të veçantë në aplikacion, kanë parametrat e tyre, etj., ku më detajisht do të flasim për secilën prej tyre në vijim.

Që ta kemi sa më të qartë funksionimin e progarmit tonë, si dhe në veçanti algoritmin AES i thjeshtuar dhe modelit OFB, në vijim do të spjegojmë rolin e gjithsecilës metodë nga të dyja klasat :

subBytes(byte[][] gjendja): Kjo metodë zëvendëson çdo bajt në grupin e gjendjes me një vlerë korresponduese nga një tabelë e paracaktuar zëvendësimi.

shiftRows(byte[][] gjendja): Kjo metodë i zhvendos rreshtat e grupit të gjendjes në mënyrë ciklike majtas. Rreshti i parë nuk zhvendoset, rreshti i dytë zhvendoset një pozicion në të majtë, rreshti i tretë zhvendoset dy pozicione në të majtë dhe rreshti i katërt zhvendoset tre pozicione në të majtë.

mixColumns(byte[][] gjendja): Kjo metodë kryen një operacion të shumëzimit të matricës në grupin e gjendjes, duke përdorur një tabelë shumëzimi të paracaktuar. Ky operacion kryhet në secilën kolonë të grupit të gjendjes.

addRoundKey(byte[][] gjendja, byte[][] roundKey): Kjo metodë kryen një operacion XOR në bit në çdo bajt në grupin e gjendjes me një bajt korrespondues në grupin e çelësave të rrumbullakët.

encryptBlock(int Block, int Key) : Metoda e encryptBlock përdor një ose më shumë nga metodat e thjeshtuara AES të përmendura më parë për të kryer procesin e enkriptimit në bllokun dhe çelësin e hyrjes.

decryptBlock(int matrixArray, int Key) : Ngjashëm me metodën encryptBlock, metoda e decryptBlock përdoret në një program enkriptimi për të deshifruar një bllok të vetëm të dhënash që është koduar më parë duke përdorur AES të thjeshtuar. Metoda e decryptBlock merrë dy parametra, njëri përfaqëson bllokun e koduar të të dhënave dhe tjetri përfaqëson çelësin e deshifrimit që do të përdoret. Procesi i deshifrimit përfshinë përdorimin e anasjelltë të operacioneve të thjeshtuara AES (të tilla si invSubBytes, invShiftRows, invMixColumns dhe addRoundKey) për të kthyer hapat e kriptimit dhe për të rikuperuar bllokun origjinal të tekstit të thjeshtë.

generateKeys (int key) : Kjo metodë përdoret në programin tonë për të gjeneruar një seri çelësash të rrumbullakët nga një çelës kriptimi hyrës. Çelësat e rrumbullakët përdoren në procesin e kriptimit dhe

deshifrimit për të shtuar një shtresë shtesë sigurie në algoritmin e kriptimit. Kjo metodë merr një vlerë të plotë ose një grup bajtësh si çelës hyrës dhe gjeneron një seri çelësash të rrumbullakët bazuar në atë çelës. Ky proces përfshin kryerjen e një sërë operacionesh të zgjerimit të çelësit në çelësin e hyrjes për të prodhuar një grup çelësash të rrumbullakët që do të përdoren në procesin e enkriptimit dhe deshifrimit.

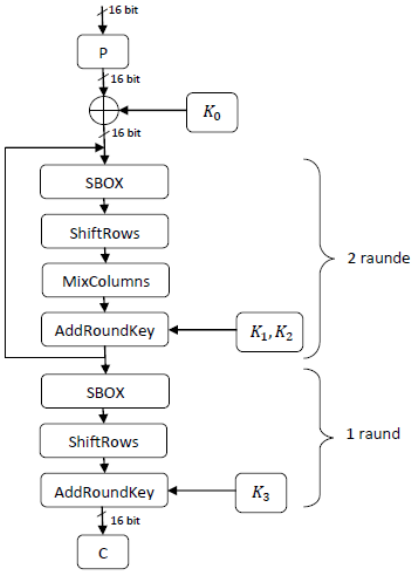

Figura 2. Skema e AES

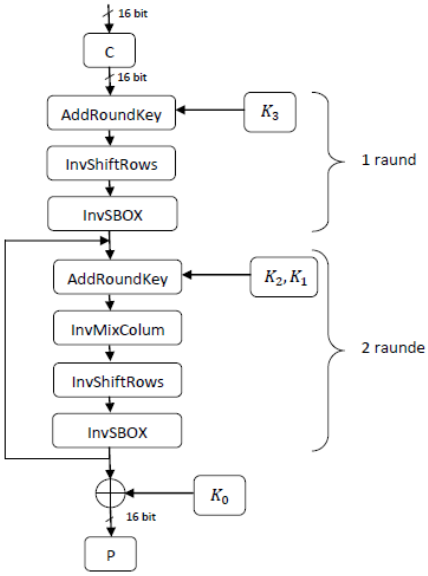

Figura 3. Dekriptimi AES

4. Testimi i Aplikacionit

Algoritmi i Avancuar i Enkriptimit (AES) është një sistem kriptografik simetrik i përdorur për të siguruar komunikime të sigurta në internet. AES është i njohur si një nga standardet kryesorë të enkriptimit, dhe është përdorur në shumë aplikacione, duke përfshirë shërbime të tilla si bankat dhe platformat e tregtisë elektronike. AES është një sistem kriptografik simetrik, që do të thotë se të njëjtit çelës të përdorur për të enkriptuar një mesazh është përdorur edhe për të dekriptuar atë. AES përdor një çelës kriptografik të njëjtë për të enkriptuar dhe dekriptuar mesazhe, dhe kjo bën që procesi të jetë shumë i shpejtë dhe efektiv. AES përdor një proces të quajtur "substitution-permutation network", në të cilin secili bllok i dhënës është ndarë në blloqe më të vogla dhe pastaj është kriptuar duke përdorur një seri procesesh të ndryshme. Ky

proces i kriptimit është bërë në mënyrë që të jetë shumë i vështirë për të ndaluar dikë që të kuptojë origjinalin nga mesazhi i enkriptuar. AES përdor një çelës 128-bit ose 256-bit për të enkriptuar mesazhet, dhe në funksionin e tij të enkriptimit përdor një serinë e rrethimeve të procesimit të të dhënave për të kriptuar mesazhin. Në fund, AES jep një mesazh të enkriptuar që është i njohur vetëm për ata që kanë qasjen në çelësin kriptografik të përdorur për të kriptuar mesazhin. AES është një sistem i sigurtë i enkriptimit dhe është i njohur si një nga standardet kryesorë të enkriptimit. Për ta përdorur AES në programimin në gjuhën shqipe, mund të përdorni bibliotekat standarde të Java, siç është klasa Cipher, për të enkriptuar dhe dekriptuar mesazhe në AES. Në vijim do të diskutohet AES i thjeshtuar.
Enkriptimi në AES të thejshtuar është si vijon:

Struktura e kodit është si vijon:

Marrim File me tekst që e ka dhe e kthejmë në bit, pastaj ato bite kalojnë nepër ato fazat që përmendëm më lartë. Këtu do të shpjegohen vetëm pjesët kryesore të kodit që e bën atë atraktive.

```
The Text file has these contents:
Hello World!
My name is David Beckham,
This is a project about encryption.
I'm gonna use some number's!
1+2+3 = 123 (Java LOL)
Java Juice?
```

Figura 4. Mesazhi i paenkriptuar

Teksti të cilin ne e kemi për të enkriptuar apo para enkriptimit, ky tekst nga file përmes një skaneri. Kodi pastaj kalon nepër disa metoda për enkriptim dhe dekriptim derisa të arrij tek rezultati që duam ne.

```
public String encryptDecryptString(String string, int key, boolean decrypt){
    if(!decrypt && string.length() % 2 != 0)
        string += " ";
    byte[] stringBytes = string.getBytes(StandardCharsets.ISO_8859_1);
    for(int i = 0; i < stringBytes.length; i += 2){
        int c = (Byte.toUnsignedInt(stringBytes[i]) << 8) | Byte.toUnsignedInt(stringBytes[i + 1]);
        int encrypt = decrypt? decryptBlock(c, key) : encryptBlock(c, key);
        stringBytes[i] = (byte)((encrypt & Integer.parseUnsignedInt(s:"FF00", radix:16)) >> 8);
        stringBytes[i+1] = (byte)(encrypt & Integer.parseUnsignedInt(s:"FF", radix:16));
    }
    return new String(stringBytes, StandardCharsets.ISO_8859_1);
```

Figura 5. Metoda encryptDecryptString qe kontrollon bllokun nese është i enkriptuar apo dekriptuar

Teksti kalon neper kete metod për të kontrolluar nëse është i enkriptuar apo jo. Nëse është atëherë metoda “encryptDecryptString” e kthen tekstin në të dekriptuar, përkunderazi e kthen atë në të enkriptuar.

```java
public int encryptBlock(int block, int key){
    int[] keys = generateKeys(key);
    block ^= keys[0];
    for(int i = 1; i < 3; i++){
        block = SBOX(block, inverse:false);
        block = shiftRows(block, inverse:false);
        block = mixColumns(block, inverse:false);
        block = addRoundKey(block, keys[i]);
    }
    block = SBOX(block, inverse:false);
    block = shiftRows(block, inverse:false);
    block = addRoundKey(block, keys[3]);
    return block;
}
```

Figura 6. Metoda qe enkripton bllokun nëpër raundet e shpjeguara me siper

Duke e përdorur metodën më lartë për të kontrolluar nëse është e enkriptuar apo jo, dhe na jep rezultatet. Supozojmë që nuk është e enkriptuar atëherë metoda e dërgon atë ne këtë metodën tjetër që e enkripton tekstin në mënyr blloku, pra 16 bit.

```
The Text file after encryption:
↕27@£iMZ↕?

YY??«?♦0â?ô?D»↕u/t?}=?«?z?
?ëâ?÷)?-i¥BWCâ?}
????¶
?&????µ[ù▲µT
Ys??§H‼═yß??♦0?
?▲N\çu?4~\ç²5D»i¥PQ▼-
(-?3▲↕ ♥
?Ñ?
```

Figura 7. Mesazhi pas enkriptimit

Mesazhi i enkriptuar do të na shfaqët nga karakterë të ndryshëm, si do qoftë “rëndomizimi” i këtyre karaktereve varet nga një paketë që ka një keyword të quajtur “charset” ku aty mund të zgjedhim shkronja dhe karaktere të ndryshëm(UTF-8, UNICODE, ASCII).

Pasi është enkriptuar, dhe ne dëshirojme ta dekriptojmë atëherë përdorim metodën decryptBlock që e ka për detyrë dekriptimin e bllokut të të dhënave.

```java
public int decryptBlock(int matrixArray, int key){
    int[] keys = generateKeys(key);
    matrixArray = addRoundKey(matrixArray, keys[3]);
    matrixArray = shiftRows(matrixArray, inverse:true);
    matrixArray = SBOX(matrixArray, inverse:true);
    for(int i = 2; i > 0; i--){
        matrixArray = addRoundKey(matrixArray, keys[i]);
        matrixArray = mixColumns(matrixArray, inverse:true);
        matrixArray = shiftRows(matrixArray, inverse:true);
        matrixArray = SBOX(matrixArray, inverse:true);
    }
    matrixArray ^= keys[0];
    return matrixArray;
}
```

Figura 8. Metoda qe e dekripton bllokun

Pas enkriptimit dhe dekriptimit rezultati do të na shfaqet mesazhi përsëri siq ka qenë i paenkriptuar:

```
The Text file after decryption:
Hello World!
My name is David Beckham,
This is a project about encryption.
I'm gonna use some number's!
1+2+3 = 123 (Java LOL)
Java Juice?
```

Figura 9. Mesazhi pas dekriptimit

5. OFB (Output FeedBack)

AES OFB (Output Feedback) është një nga modalitetet e AES (Advanced Encryption Standard) që përdoret për të siguruar komunikimin e të dhënave në mënyrë të sigurt. Ky modalitet është i njohur për sigurinë dhe performancën e tij, dhe është përdorur në aplikacione të ndryshme, duke përfshirë internetin dhe sistemet e komunikimit.

AES OFB bazohet në përdorimin e një funksioni feedback (shenjë) që llogaritet në mënyrë të pavarur nga mesazhi origjinal. Kjo mundëson që sistemi i enkriptimit të jetë më fleksibël, dhe siguron që mesazhi origjinal të mos përsëritet nëse ky proces do të kryhet sërish. Me këtë mënyrë, sistemi nuk lejon ndikimin e një personi të tretë në të dhënat që po dërgohen.

Procesi i AES OFB fillon me përzgjedhjen e një vektori inicializues (IV) të rastësishëm prej 128 bitësh. Ky vektor do të jetë një kodi fillimor për çdo transmetim të dhënash. Pastaj, AES OFB kryen operacionin e AES me çelësin sekret prej 128 bitësh dhe vektorin inicializues si input. Rezultati i kësaj operacioni quhet stream cipher dhe është një seri e dhënash të randomizuara.

Pas kësaj, stream cipher është kombinuar me mesazhin origjinal duke përdorur operacionin e XOR. Si rezultat, stream cipher është përdorur si një funksion feedback për të prodhuar stream-in tjetër. Kështu, AES OFB siguron një seri të stream-ave të ndryshëm që përdoren për të kriptuar dhe dekriptuar mesazhin origjinal.

Një avantazh i AES OFB është se ai mund të përdoret për të enkriptuar të dhëna të mëdha në kohë reale, pa nevojën për të ndarë të dhënat në blloqe të vogla. Gjithashtu, ai siguron një siguri të lartë në transmetim të të dhënave duke përdorur funksionin feedback dhe stream-e të randomizuara.

Sidoqoftë, AES OFB nuk është pa mungesa. Për shembull, ai nuk është i përshtatshëm për aplikacione që kanë nevojë për një siguri maksimale në disa situata të veçanta. Për më tepër, nuk ka mekanizma që ndalon një sulmues nga përdorimi i të njëjtit stream për të gjetur këtë stream dhe shkat. Skema e OFB:

$$
\begin{aligned}
&s_1 = E_k(IV),\ y_1 = s_1 \oplus x_1,\\
&s_i = E_k(s_{i-1}),\ y_i = s_i \oplus x_i,\ i \geq 2\\
&s_1 = E_k(IV),\ x_1 = s_1 \oplus y_1,\\
&s_i = E_k(s_{i-1}),\ x_i = s_i \oplus y_i,\ i \geq 2
\end{aligned}
$$

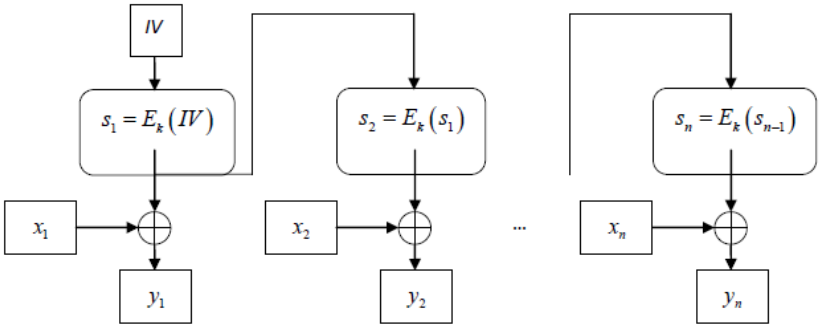

Figura 10. Enkriptimi përmes OFB (Skema)

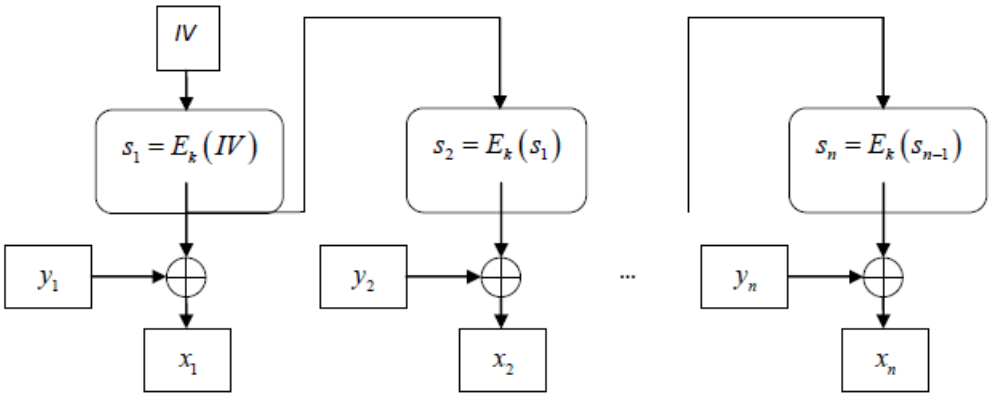

Figura 11. Dekriptimi përmes OFB (Skema)

6. Përmbledhje

Dhe kështu erdhëm në përfundim të dokumentimit të projektit tonë. Projektin tonë jemi munduar që ta marrim me seriozitet të plotë, ashtu siç edhe e kërkon fakulteti, jemi munduar që ta punojmë jashtëzakonisht shumë mirë, pa gabime, me shumë modifikime që e bëjnë programin më efektivë ndaj shfrytëzuesit, dhe jemi munduar që të përdorim gjëra që kemi mësuar nga të dy profesorët e nderuar gjatë këtij semestri në lëndën **Siguria e të Dhënave**.
Edhe Dokumentimin e Projektit jemi munduar ta bëjmë sa më mirë, pasi që një projekt sado i mirë që të jetë, nëse nuk ka dokumentim, atëherë ai është një "**hiç**".

Jemi munduar që ta spjegojmë shumë mirë projektin tonë, duke përfshirë edhe figura ashtu që të jemi sa më të qartë me ju profesorë, por edhe me ndonjë shfrytëzues tjetër. Të gjitha veçoritë e projektit tonë, të cilat i kemi treguar tek **Dokumentimi Teknik**, jemi munduar t'i vërtetojmë përmes **Testimit të Aplikacionit** dhe të jemi sa më të sinqertë.
Ne e kemi dhënë më të mirën tonë, andaj shpresojmë që të ju pëlqejë projekti jonë. Natyrisht se ka vend për shumë përmirësime, fundja ky është qëllimi ynë që të marrim kritika nga profesorët tonë dhe të punojmë në përmirësimin e tyre. Ne ju falënderojë shumë për përkrahjen dhe spjegimin e qartë gjatë këtij semestri, ishte shumë kënaqësi që të ishim student i juaji.

Me shumë rrespekt, Egzonit Demhasaj & Gabriel Kolaj
Prishtinë, Prill 2023