TEMA: SAES-CTR

LËNDA: SIGURIA E TË DHËNAVE

UNIVERSITETI I PRISHTINËS

PUNOI:
BLEON BALAJ
ELJON KASTRATI

2023

PËRMBAJTJA

AES, FUNKSIONI DHE PËRDORIMI TIJ

Siguria e të dhënave është një aspekt thelbësor i përdorimit të internetit dhe AES (Advanced Encryption Standard) është një algoritëm simetrik i enkriptimit i përdorur gjerësisht për të mbrojtur të dhënat e ndjeshme. AES funksionon duke marrë një bllok teksti të thjeshtë dhe duke e enkriptuar atë në një bllok teksti shifror duke përdorur një çelës simetrik. Ky çelës përdoret për skajet e tekstit të thjeshtë në një format të panjohur dhe teksti i shifruar që rezulton mund të deshifrohet vetëm me të njëjtin çelës.

Një nga avantazhet kryesore të AES është aftësia e tij për të përdorur një sërë madhësish kryesore, duke përfshirë 128, 192 dhe 256 bit. Sa më e madhe të jetë madhësia e çelësit, aq më i sigurt është kriptimi, pasi ka më shumë kombinime të mundshme çelësash që mund të përdoren për të kriptuar dhe deshifruar të dhënat. AES përdor një proces të quajtur zgjerimi i çelësit për të gjeneruar nënçelësat e përdorur në procesin e kriptimit. AES përdoret gjerësisht në një gamë të gjerë aplikacionesh ku siguria e të dhënave është thelbësore, duke përfshirë sigurimin e të dhënave në tranzit përmes internetit, enkriptimin e skedarëve të ruajtur në disqet e ngurtë dhe mbrojtjen e informacionit të ndjeshëm si numrat e kartave të kreditit dhe fjalëkalimet.

Procesi i enkriptimit fillon duke e ndarë bllokun e tekstit të thjeshtë në copa më të vogla 128-bit. Secila pjesë pastaj kalon nëpër një sërë operacionesh matematikore, duke përfshirë zëvendësimin, zhvendosjen dhe XOR-in në bit. Këto operacione janë krijuar për të shkurtuar të dhënat në një mënyrë që të jetë e pamundur t'i lexoni ato pa përdorur çelësin e duhur. Më pas, teksti i shifruar që rezulton transmetohet ose ruhet në mënyrë të sigurt.

Përveç përfitimeve të sigurisë, AES është gjithashtu shumë i shpejtë dhe efikas, duke e bërë atë një zgjedhje ideale për përdorim në sistemet dhe aplikacionet e internetit në kohë reale që kërkojnë përpunim të shpejtë të të dhënave. Sistemet e enkriptimit, duke përfshirë AES, mund të prishen nga sulme të ndryshme kriptografike, duke përfshirë sulmet bruteforce, sulmet nga njeriu në mes dhe sulmet me injeksion. Për këtë arsye, është e rëndësishme që kushdo që përdor AES dhe teknologji të tjera sigurie të ketë një strategji të përgjithshme sigurie që përfshin ruajtjen e çelësave të sigurisë dhe përdorimin e protokolleve të sigurta të sigurisë së të dhënave.

COUNTER MODE (CTR)

Counter Mode (CTR) është një mënyrë funksionimi për shifrat e bllokut, si AES (Advanced Encryption Standard), i përdorur gjerësisht në sigurinë e të dhënave për të siguruar konfidencialitetin dhe integritetin e informacionit të ndjeshëm.

Modaliteti CTR është një shifër transmetimi që lejon një shifër blloku të krijojë blloqe të ndryshme të të dhënave të pavarura. Ai funksionon duke prodhuar një rrjedhë unike kyçe për çdo bllok të tekstit të thjeshtë, i cili më pas kombinohet me tekstin e thjeshtë duke përdorur një operacion XOR për të prodhuar tekstin e shifruar. Rrjedha e çelësit prodhohet duke koduar një vlerë numërues duke përdorur shifrën e bllokut, dhe teksti i shifruar i prodhuar përdoret si rrjedha kryesore për bllokun e tekstit të thjeshtë ngjitur. Vlera e numëruesit rritet për çdo bllok të mëtejshëm, duke siguruar që rrjedha kryesore të jetë unike për çdo bllok.

Një nga përfitimet kryesore të modalitetit CTR është aftësia e tij për të koduar sasi të mëdha të dhënash në një model me akses të rastësishëm. Kjo do të thotë që çdo bllok i të dhënave mund të kodohet dhe deshifrohet pavarësisht nga pjesa tjetër e të dhënave. Kjo është veçanërisht e dobishme në situatat kur skedarët e mëdhenj duhet të kodohen ose deshifrohen, pasi mundëson përpunim efikas të të dhënave.

Një përfitim tjetër i modalitetit CTR është rezistenca e tij ndaj disa llojeve të sulmeve, siç janë sulmet e tekstit të koduar. Kjo është për shkak se rrjedha e çelësit nuk vjen nga teksti i qartë, duke e bërë të vështirë për një sulmues të modifikojë tekstin e shifruar pa u zbuluar. Për të përdorur modalitetin CTR në sigurinë e të dhënave, një organizatë zakonisht do të integrohet në protokollet e saj ekzistuese të enkriptimit dhe do të sigurojë që të gjitha të dhënat e ndjeshme të kodohen duke përdorur këtë mënyrë.

SIGURIA QË OFRON AES ME CTR

Kriptimi AES me modalitetin CTR është një shifër simetrik blloku që kodon dhe deshifron të dhënat në blloqe prej 128 bitësh. Ai përdor një çelës prej 128, 192 ose 256 bitësh dhe një numërues si një vektor inicializimi. Numëruesi rritet për çdo bllok dhe kodohet me çelës. Teksti i shifruar që rezulton XORohet me tekstin e thjeshtë për të prodhuar daljen e koduar. Procesi i deshifrimit është i njëjtë me enkriptimin, por në rend të kundërt.

PËRPARËSITË

Disa avantazhe të kriptimit AES me modalitetin CTR janë:

- Është e lehtë për t'u zbatuar si në harduer ashtu edhe në softuer, dhe mund të funksionojë paralelisht në bërthama të shumta.
- Nuk kërkon mbushje, pasi teksti i thjeshtë mund të jetë çdo gjatësi.
- Është rezistent ndaj përhapjes së gabimeve, pasi një gabim i vetëm bit në tekstin e shifruar prek vetëm një bit në tekstin e thjeshtë.
- Lejon akses të rastësishëm në të dhënat e koduara, pasi çdo bllok mund të deshifrohet në mënyrë të pavarur.

MANGESITË

Disa disavantazhe të kriptimit AES me modalitetin CTR janë:

- Kërkon një numërues unik për çdo kriptim, përndryshe rrezikohet siguria.
- Nuk ofron vërtetim ose mbrojtje të integritetit, kështu që duhet të kombinohet me një kod vërtetimi të mesazhit (MAC) ose një modalitet kriptimi të vërtetuar (si GCM ose EAX).
- Është i pambrojtur ndaj sulmeve të përsëritura, pasi një sulmues mund të ripërdorë një bllok teksti të shifruar më parë.
- Është i ndjeshëm ndaj sulmeve të kohës, pasi një sulmues mund të masë kohën që duhet për të enkriptuar ose deshifruar një bllok.

IMPLEMENTIMI I ALGORITMIT NË GJUHËN JAVA

```
 1 import javax.crypto.Cipher;
 2 import javax.crypto.spec.IvParameterSpec;
 3 import javax.crypto.spec.SecretKeySpec;
 4
 5 public class SAES {
 6     private static final String ALGORITHM = "AES";
 7     private static final String TRANSFORMATION_ECB =
"AES/ECB/NoPadding";
 8     private static final String TRANSFORMATION_CTR =
"AES/CTR/NoPadding";
 9
10     public static void main(String[] args) {
11         try {
12             // Per te importuar nga text file, vendoset path-i
13             String celesi = "0123456789abcdef";
14             String mesazhi = "0123456789abcdef";
15
16             System.out.println("Mesazhi: " + mesazhi);
```

```
17
18              byte[] encryptedBytes = encrypt(mesazhi, celesi);
19              System.out.println("Mesazhi i enkriptuar: " +
bytesToHex(encryptedBytes));
20
21              String decryptedMessage = decrypt(encryptedBytes, celesi);
22              System.out.println("Mesazhi i dekriptuar: " +
decryptedMessage);
23
24              byte[] encryptedBytesCTR = encryptCTR(mesazhi, celesi);
25              System.out.println("Mesazhi i enkriptuar (CTR): " +
bytesToHex(encryptedBytesCTR));
26
27              String decryptedMessageCTR = decryptCTR(encryptedBytesCTR,
celesi);
28              System.out.println("Mesazhi i dekriptuar (CTR): " +
decryptedMessageCTR);
29          } catch (Exception e) {
30              e.printStackTrace();
31          }
32      }
33
34      public static byte[] encrypt(String mesazhi, String celesi) throws
Exception {
35          SecretKeySpec secretKey = new SecretKeySpec(celesi.getBytes(),
ALGORITHM);
36          Cipher cipher = Cipher.getInstance(TRANSFORMATION_ECB);
37          cipher.init(Cipher.ENCRYPT_MODE, secretKey);
38
39          byte[] mesazhiBytes = mesazhi.getBytes();
40          byte[] encryptedBytes = cipher.doFinal(mesazhiBytes);
41
42          return encryptedBytes;
43      }
44
45      public static String decrypt(byte[] encryptedBytes, String celesi)
throws Exception {
46          SecretKeySpec secretKey = new SecretKeySpec(celesi.getBytes(),
ALGORITHM);
47          Cipher cipher = Cipher.getInstance(TRANSFORMATION_ECB);
48          cipher.init(Cipher.DECRYPT_MODE, secretKey);
49
50          byte[] decryptedBytes = cipher.doFinal(encryptedBytes);
51          String decryptedMessage = new String(decryptedBytes);
52
53          return decryptedMessage;
54      }
55
56      public static byte[] encryptCTR(String mesazhi, String celesi)
throws Exception {
57          SecretKeySpec secretKey = new SecretKeySpec(celesi.getBytes(),
ALGORITHM);
58          Cipher cipher = Cipher.getInstance(TRANSFORMATION_CTR);
59
60          byte[] ivBytes = new byte[16]; // Initialization Vector (IV) for
CTR mode
61          IvParameterSpec ivSpec = new IvParameterSpec(ivBytes);
```

```
62          cipher.init(Cipher.ENCRYPT_MODE, secretKey, ivSpec);
63
64          byte[] mesazhiBytes = mesazhi.getBytes();
65          byte[] encryptedBytes = cipher.doFinal(mesazhiBytes);
66
67          return encryptedBytes;
68      }
69
70      public static String decryptCTR(byte[] encryptedBytes, String
celesi) throws Exception {
71          SecretKeySpec secretKey = new SecretKeySpec(celesi.getBytes(),
ALGORITHM);
72          Cipher cipher = Cipher.getInstance(TRANSFORMATION_CTR);
73
74          byte[] ivBytes = new byte[16]; // IV vektori per CTR
75          IvParameterSpec ivSpec = new IvParameterSpec(ivBytes);
76          cipher.init(Cipher.DECRYPT_MODE, secretKey, ivSpec);
77
78          byte[] decryptedBytes = cipher.doFinal(encryptedBytes);
79          String decryptedMessage = new String(decryptedBytes);
80
81          return decryptedMessage;
82      }
83
84      public static String bytesToHex(byte[] bytes) {
85          // Konverteri
86          StringBuilder result = new StringBuilder();
87          for (byte b : bytes) {
88              result.append(String.format("%02x", b));
89          }
90          return result.toString();
91      }
92  }
```

KOMPLEKSITETI I ALGORITMIT

Kompleksiteti i kriptimit AES me modalitetin CTR varet nga zbatimi i algoritmit. AES-CTR përdor të vetmin operacion të enkriptimit AES (si për enkriptim ashtu edhe për deshifrim), duke i bërë implementimet AES-CTR më të vogla se zbatimet e shumë mënyrave të tjera AES.