UDC: 004.42.021:003.26 INFO M: str. 34-39

# EKSPERIMENTI SA MOGUĆIM MODIFIKACIJAMA AES ALGORITMA
# EXPERIMENTS WITH POSSIBLE MODIFICATIONS OF AES ALGORITHM

Boris Damjanović, FON
Dejan Simić, FON
Univerzitet u Beogradu

REZIME: Advanced Encryption Standard (AES) je kriptografski standard koji je nastao kao rezultat javno objavljenog takmičenja od strane Nacionalnog instituta za standarde i tehnologiju vlade SAD. Ovaj algoritam teoretski može da se podijeli na tri kriptografska algoritma: AES-128, AES-192 i AES-256. Standardom definisani AES nastao je na osnovama nešto šireg Rijndael algoritma od kojega je naslijedio veliku fleksibilnost. Upravo navedena fleksibilnost ovoga algoritma predstavljala je ideju vodilju na osnovu koje je proveden niz eksperimenata pomoću kojih je testirana mogućnost implementacije različitih modifikacija ovoga algoritma. U radu će biti prikazani dijelovi koji su ključni za implementaciju 64-bitne i 32-bitne enkripcije i dekripcije. Osnovna ideja ovoga rada je da ukaže na veliku fleksibilnost ovoga algoritma i da na primjeru pokaže kako se može implementirati redukovana verzija ovog algoritma koja se može koristiti u oblasti obrazovanja. U ovom radu su predstavljeni samo najinteresantniji koncepti nastali iz nešto šireg master rada koji je rađen na Fakultetu organizacionih nauka Univerziteta u Beogradu.
KLJUČNE REČI: kriptografija, algoritmi
ABSTRACT: Advanced Encryption Standard (AES) is a cryptographic standard that emanated trough a public competition announced by the US National Institute of Standard and Technology. Theoretically, this algorithm can be divided into three cryptographic algorithms: AES-128, AES-192 and AES-256. The AES specified by the standard is derived from the slightly more complex Rijndael algorithm from which it inherits fairly large flexibility. The flexibility of the algorithm led into a row of experiments to verify the ability of various modifications of the algorithm to be put in implementation. This paper sets out particular sequences that are substantial for 64-byte and 32-bity encryption and decryption. This work purposely points out a great flexibility of this algorithm and produces an example how its reduced version can be implemented in the field of education. The paper shows only the most interesting concepts elaborated in details in a Master Thesis done at Faculty of Organizational Sciences at the University of Belgrade.
KEY WORDS: cryptography, algorithms

## I. UVOD

Drugog januara 1997. godine, Nacionalni institut za standarde i tehnologiju (NIST) [23] je pokrenuo javno takmičenje za izradu algoritma koji će postati sljedeći standard vlade SAD i koji će zamijeniti nešto stariji DES algoritam. Sedam od prvobitno prijavljenih 22 kandidata nisu zadovoljili ulazne kriterijume, tako da je u dalji proces testiranja ušlo 15 algoritama. U 1999. godini, ova lista svedena je na 5 kandidata i počeo je sljedeći krug testiranja. Krajem 2000. godine, NIST je proglasio pobjednika – algoritam zvan Rijndael (po imenima pronalazača Vincent Rijmen i Joan Daemen, a izgovara se kao "Rhine Dahl", "Rain Doll" ili "Reign Dahl"). Ovaj algoritam je zadovoljio u svim testovima, a američka vlada je dio Rijndaela koji je prihvatila, nazvala Advanced Encryption Standard (AES). Rijndael dozvoljava da veličinu ključa kao i veličinu bloka biramo iz skupa {128, 160, 192, 224, 256} bita. S druge strane, dokument FIPS-197 [30] koji definiše AES algoritam, navodi da dužina bloka mora uvijek biti 128 bita, dok dužine ključeva mogu biti 128, 192 ili 256 bita.

Već i sama činjenica da AES algoritam predstavlja samo podskup Rijndael algoritma govori dovoljno o njegovoj velikoj fleksibilnosti i intuitivno upućuje na nekoliko pravaca njegovih mogućih modifikacija – npr. na izmjenu vezanu za veličinu bloka podataka (State), na proširenje/smanjenje prostora ključeva, povećanje/smanjenje broja rundi, pomjeranje matematičkih operacija iz GF(2^8) u npr. GF(2^16) i ojačavanje rutine za ekspanziju ključeva.

Tokom rada na ovom projektu proveden je niz eksperimenata vezanih za implementaciju AES algoritma:

- Implementacija standardom definisanog AES algoritma (Delphi i Java) bez optimizacija;
- Implementacija 32, 64, 320, 384, 448 i 512 bitnih modifikacija (Delphi i Java) bez optimizacija;
- Implementacija standardom definisanog algoritma i 32, 64, 320, 384, 448 i 512 bitnih modifikacija sa optimizacijama zasnovanim na modifikovanim Gladmanovim idejama (Java) [14] [17];
- Implementacija standardom definisanog algoritma i 32, 64, 320, 384, 448 i 512 bitnih proširenja sa optimizacijama zasnovanim na modifikovanim Bertonijevim idejama (Java) [2].

U ovom tekstu će biti predstavljen problem implementacije AES algoritma sa redukcijom dužine njegovih inicijalnih ključeva na 64 i 32 bita uz odgovarajuće smanjenje broja rundi algoritma. Osnovna ideja ovoga rada je da iskoristi veliku fleksibilnost algoritma AES i da konstruiše pojednostavljenu verziju algoritma koja se može koristiti u obrazovne svrhe. U tekstu će biti prikazani samo fragmenti koda koji su ključni

za implementaciju 32 i 64-bitnih modifikacija, dok se kompletan kod aplikacije koja implementira 128, 192, 256, te 32. i 64-bitnu enkripciju/dekripciju nalazi u [18]. Redukovana verzija algoritma koju ćemo ovdje prikazati, mogla bi biti iskorišćena u školskim ustanovama i u laboratorijama i institucijama koje se bave obrazovanjem.

## II. PREGLED SRODNIH RJEŠENJA

Ideja o korišćenju redukovanih verzija algoritama u obrazovne svrhe nije nova. Pojednostavljeni DES (S-DES) [26] koji je razvio profesor Edward Schaefer sa Univerziteta Santa Clara imao je slične osobine i strukturu kao pravi DES algoritam, ali su korišćene vrijednosti parametara bile mnogo manje. Ovakav pojednostavljeni DES je koristio 8-bitne blokove podataka, 10-bitne ključeve a za šifrovanje je koristio samo dvije runde. Pojednostavljenu verziju algoritma DES i Mollin u [22] takođe koristi kao nastavno sredstvo.

Mini verzija AES algoritma koju je razvio profesor Raphael Chung-Wei Phan sa [5] u radu koristi niblove (eng. nibble, 4 bita) za razliku od pune verzije algoritma koja koristi bajtove. Zbog ovoga su u mini verziji AES algoritma morale biti ponovo redefinisane operacije množenja i sabiranja tako da koriste GF(2^4). Ovakva verzija je koristila blok podataka (State) veličine 16 bita, za razliku od pune verzije koja je koristila blok veličine 16 bajta. Također i profesor Steven Gordon u nastavi koristi pojednostavljene verziije AES [16] i DES [15] algoritama. Pored ovoga zabilježeni su i različiti drugi pokušaji pojednostavljivanja kriptografskih algoritama za različite svrhe. Tako Manangi et al. [19] et al. predlažu pojednostavljenu verziju AES algoritma za upotrebu u embedded sistemima.

Objavljivanje pojednostavljenih verzija kriptografskih algoritama koje su konstruisane u obrazovne svrhe često su pratile i ideje o upotrebi ovakvih algoritama u kriptoanalizi. Tako je profesor Edward Schaefer u [26] prikazao nekoliko metoda za kriptoanalizu vlastite pojednostavljene verzije DES algoritma. Profesor Raphael Chung-Wei Phan je kasnije iskoristio svoju pojednostavljenu verziju AES algoritma (Mini AES) za izvođenje napada nemogućom diferencijalnom kriptoanalizom (eng. Impossible Differential Cryptanalysis) [4]. Matsui [20], Ooi [25], Sharma et al. [28] i Dunkelman et al. [13] koriste pojednostavljene verzije algoritma DES, a Demirci et al. [11], Doganakoy et al. [12], Biryukov et al. [3] i Adam Berent [1] koriste pojednostavljene verzije algoritma AES za potrebe kriptoanalize. Kada govorimo o kriptoanalizi, treba reći da je veliki broj redukovanih i pojednostavljenih verzija različitih algoritama našao primjenu u kriptoanalizi, jer dužina ključa koji se koristi u nekom kriptografskom algoritmu presudno određuje mogućnost izvršenja brute-force napada pri čemu su duži ključevi eksponencijalno otporniji od kraćih. U kriptoanalizi se kao razbijanje nekog algoritma računa bilo koji metod koji je brži od brute force metode. Kako Schneier [27] navodi, razbijanje nekog algoritma jednostavno znači pronalaženje njegovih slabosti koje mogu biti iskorišćene sa kompleksnošću koja je manja od kompleksnosti brute-force napada.

Prema Noveru [24], prvi napadi na DES algoritam su bili izvedeni na dovoljno pojednostavljenim i redukovanim verzijama ovog algoritma, ali na onim verzijama u koje je kriptografska zajednica mogla da ima povjerenja da ne sadrže neku namjerno sakrivenu manu. Ovdje treba podvući razliku između napada na redukovane i pojednostavljene verzije algoritma. Matsui u [20] predstavlja metod za linearnu kriptoanalizu verzije DES algoritma koja je redukovana na 8 i 12 rundi. Dunkelman et al. u [13] predstavlja meet-in-the-middle napad na verziju DES algoritma koja je redukovana na šest rundi. Demirci et al. u [11] predstavlja redukovanu verziju AES algoritma radi izvođenja meet-in-the-middle napada. Pojednostavljena verzija algoritma AES u kojoj je implementiran skraćeni S-Box i kome je izmjenjena funkcija ekspanzije ključeva u [29] je najprije redukovana na dvije runde, a zatim je takva pojednostavljena verzija analizirana metodama algebarske kriptoanalize. Biryukov et al. [3] opisuje mogućnost Related Key napada na 256 bitnu verziju AES algoritma koja je redukovana na 9 rundi, kao i Chosen-plaintext i Chosen-ciphertext napad na 256 bitnu verziju AES algoritma koja je redukovana na 10 rundi. Doganakoy et al. [12] daje pregled većeg broja napada na algoritme sa redukovanim brojem rundi.

## III. POLAZNE OSNOVE

U ovom tekstu će biti predstavljena implementacija redukovane verzije AES algoritma koja će koristiti inicijalne ključeve dužine 64 i 32 bita uz odgovarajuće smanjenje broja rundi algoritma i izmjenu rutine za ekspanziju ključeva. Kao osnova za rad na ovom projektu korišćen je dokument FIPS-197 [30], koji definiše AES algoritam. U daljem tekstu će se AES[6] [7] algoritam koji je na ovaj način implementiran nazivati standardom definisani ili klasični AES, dok će u tekstu za njegovu modifikovanu verziju biti korišćen naziv pojednostavljeni AES.

U ovom tekstu biće zadržane sve notacije i konvencije koje se koriste u klasičnom AES-u. Ove notacije i konvencije koje se odnose na označavanje i način rada sa ulaznim i izlaznim vektorima, pojedinačnim bitima i bajtima te blokom Stanje (State) preuzete su iz dokumenta FIPS-197 koji definiše standard i kao takve biće zadržane u ovom tekstu.

Autori AES algoritma i Rijndael-a redefinišu operacije sabiranja i množenja na nivou bajta u okviru konačnih (Galoa) polja [30] [21]. U ovom projektu su zadržane neizmjenjene operacije sabiranja i množenja u okviru G(2^8) onako kako ih definiše sam standard. Sve transformacije standardnog AES-a koje su definisane u [30] – SubBytes, InvSubBytes, ShiftRows, InvShiftRows, MixColumns i InvMixColumns će u ovom tekstu biti zadržane u funkcionalno neizmjenjenom obliku i pod neizmjenjenim nazivima. U samoj implementa-

ciji je većina ovih transformacija predstavljena funkcijama ili procedurama koje nose navedena imena.

Rutina ekspanzije ključeva je za slučaj klasičnog AES algoritma ostala funkcionalno neizmjenjena u odnosu na [30], te je tokom implementacije takvog standardom definisanog algoritma podijeljena u tri odvojene funkcije (KeyExpansion128, KeyExpansion192 i KeyExpansion256) koje se odnose na generisanje ključeva razlilčitih dužina. Ovo je i prva transformacija koja je u slučaju pojednostavljenog AES-a morala biti izmjenjena da bi omogućila generisanje ključeva čija je dužina različita od onih koje su definisane standardom.

Transformacije koje su zadužene za šifrovanje i dešifrovanje (Cipher i InvCipher) su takođe zadržane neizmjenjene u odnosu na standard za slučaj klasičnog AES-a. Međutim, ove transformacije su, zahvaljujući velikoj fleksibilnosti ovog algoritma, dograđene tako da u pojednostavljenom algoritmu mogu da koriste manji broj rundi i manje dužine ključeva.

U ovom tekstu će biti predstavljena redukovana verzija algoritma AES kojoj je djelimično izmjenjena (pojednostaljena) rutina za ekspanziju ključeva.

## IV. STANDARDOM DEFINISANI AES ALGORITAM

U standardnom algoritmu AES, dužine ulaznog, izlaznog bloka i stanja su 128 bita, a dužine inicijalnog ključa su 128, 192 i 256 bita. Da bi AES algoritam bio funkcionalan, inicijalni ključ se mora u funkciji ekspanzije ključeva razviti do određenog broja bajta, koji zavisi od tipa algoritma (AES-128, AES-192 ili AES-256). Broj rundi u algoritmu zavisi od dužine ključa i obrnuto. Ako dužinu inicijalnog ključa u bitima predstavimo sa Nbk, a broj rundi sa Nr, imamo:

- Nr = 10 kada je Nbk = 128,
- Nr = 12 kada je Nbk = 192,
- Nr = 14 kada je Nbk = 256.

Kada se radi o enkripciji, niz Stanje tokom svake runde prolazi kroz ranije pomenute transformacije SubBytes, ShiftRows, MixColumns i AddRoundKey kako je to prikazano u [30], [8], [9] i [10].

Suština ideje za redukciju i pojednostavljenje ovoga algoritma leži u razumjevanju procesa proširenja njegovih ključeva, koji će u nastavku biti nešto detaljnije prikazan.

### A. Niz Rcon

Prema [30], niz Rcon sastoji se od konstantnih vrijednosti koje su date formulom (1):

$$[X^{(i-1)}, \{00\}, \{00\}, \{00\}] \quad (1)$$

Niz Rcon čine eksponenti od x, gdje je x {02hex}, dok prema [30] treba voditi računa da i počinje od 1. Izvorni kod funkcije Rcon() koja koristi AES-ovo množenje za (galoa_mul_tab) može se vidjeti na lokacijama [8] i [10].

### B. Transformacija RotWord()

Ova transformacija se koristi pri ekspanziji ključeva i izvornom kodu aplikacije zadržana je u neizmjenjenom obliku u odnosu na [30]. Kako je to prikazano u [8] i [10], 8-bitna rotacija 32-bitne riječi se izvodi na jednostavan način.

$$\text{RotWord}[b0,b1,b2,b3] = [b1,b2,b3,b0] \quad (2)$$

### C. Ekspanzija ključeva

Rutina za ekspanziju ključeva na osnovu inicijalnog ključa dužine 128, 192 ili 256 bita generiše listu ključeva dužine Nk.

$$Nk = Nb(Nr+1) \quad (3)$$

gdje je Nb broj bajta Stanja (za AES uvjek 16), a Nr broj rundi algoritma (10, 12 ili 14).

Pseudo kod dat u listingu 1 prikazuje kod rutine za proširenje ključeva za slučaj 128 bitne enkripcije - KeyExpansion128() i pomoćne rutine KeyExpBase(), čije usluge koriste i sve ostale rutine za generisanje ključeva.

Listing 1: Pseudo kod transformacija keyExpBase i keyExpansion128

```
keyExpBase(byte i)
begin
byte a = 0
Call RotWord
while a <= 3
  tmpkey[a] = sbox[ tmpkey[a] ]
  tmpkey[0] = tmpkey[0] XOR Rcon(i)
  a = a + 1
end while
end

keyExpansion128
begin
byte c = 16
byte i = 1
byte a
while(c < 176)
  a = 0
  while(a <= 3)
    tmpkey[a] = key[a + c - 4]
    a = a + 1
  end while
  if (c mod 16 = 0)
    Call keyExpBase(i)
    i = i + 1
  end if
  a = 0
  while(a <= 3)
    key[c] = key[c - 16] XOR tmpkey[a]
    c = c + 1
    a = a + 1
  end while
end while
end
```

Za potpun uvid u implementaciju i način rada ove rutine pogledajte [8] i [10].

## V. IZMJENA STANDARDOM DEFINISANOG AES ALGORITMA

Standardni AES algoritam u svojoj najslabijoj verziji sa 128 bitnim ključem šifruje blok podataka tokom 10 rundi, iako bi prema [7] i 6 rundi bilo dovoljno da se izbjegnu shortcut napadi – ostale četiri runde su tu samo kao dodatno obezbjeđenje. Samo dvije runde Rijndael-a obezbjeđuju potpunu difuziju, što znači da svaki bit niza Stanje u trenutnoj rundi zavisi od svih bita Stanja prije dvije runde, ili drugačije, promjena jednog bita niza Stanje (State) u trenutnoj rundi će vjerovatno uticati na polovinu bita niza Stanje nakon dvije runde [7]. Prethodne dvije činjenice upućuju na to da se radi o vrlo čvrstom algoritmu. Ovakav algoritam se može do određene mjere oslabiti za određene specifične primjene. Ideja konstrukcije oslabljenog algoritma jeste stvaranje pretpostavki za njegovu primjenu u oblasti obrazovanja.

### A. Implementacija modifikacija: EAES-64 i EAES-32

Prema jednačini 2, inicijalni ključ mora biti proširen do odgovarajućeg broja bajta da bi algoritam bio funkcionalan.

**TABELA I.** *Odnosi dužina ključeva i broja rundi standardnog i modifikovanog AES algoritma*

| Br.rundi AES | Br.rundi pojednostavljeni AES | Inicijalni ključ (bita) | Inicijalni ključ (bajta) | Prošireni ključ (bajta) |
|---|---|---|---|---|
| 14 | | 256 | 32 | 240 |
| 12 | | 192 | 24 | 208 |
| 10 | | 128 | 16 | 176 |
| | 8 | 64 | 8 | 144 |
| | 7 | 32 | 4 | 128 |

Iako je mogao biti odabran i drugačiji pristup, pri implementaciji pojednostavljenja dužina inicijalnog ključa je smanjena na 64 odnosno 32 bita, sa odgovarajućim smanjenjem broja rundi na 8 i 7, respektivno. Ovakav pristup omogućava da pojednostavljeni algoritam ostane "u duhu" standardom definisanog algoritma i da se modifikacija može izvesti uz minimalne izmjene njegovih transformacija. Odnosi broja rundi i dužina ključeva standardom definisanog i modifikovanog AES algoritma dati su u tabeli I.

Tokom implementacije ovako modifikovanog algoritma, morale su biti izmjenjene rutine za ekspanziju ključeva, dok su funkcije za šifrovanje i dešifrovanje pretrpile tek neznatne izmjene. Zbog toga su u kod dodate još dvije nove rutine za proširenje ključeva pod nazivom KeyExpansion64 i KeyExpansion32. Pseudo kod obje rutine prikazan je u listingu 2.

Listing 2: Pseudo kod rutina za proširenje 32-bitnih i 64-bitnih ključeva

```
keyExpansion64
begin
byte c = 8
byte i = 1
byte a
while(c < 144)
  a = 0
  while(a <= 3)
   tmpkey[a] = key[a + c - 4]
   a = a + 1
  end while
  if (c mod 8 = 0)
   Call keyExpBase(i)
   i = i + 1
  end if
  a = 0
  while(a <= 3)
   key[c] = key[c - 16] XOR tmpkey[a]
   c = c + 1
   a = a + 1
  end while
end while
end

keyExpansion32
begin
byte c = 4
byte i = 1
byte a
while(c < 128)
  a = 0
  while(a <= 3)
   tmpkey[a] = key[a + c - 4]
   a = a + 1
  end while
  Call keyExpBase(i)
  i = i + 1
  a = 0
  while(a <= 3)
   key[c] = key[c - 16] XOR tmpkey[a]
   c = c + 1
   a = a + 1
  end while
end while
end
```

Prethodno prikazane rutine za ekspanziju ključeva predstavljaju osnovu ideje za modifikaciju ovoga algoritma. Navedene rutine proširuju inicijalne ključeve dužine 64 i 32 bita na 144 i 128 bajta, te na taj način omogućavaju smanjenje broja rundi u funkciji za šifrovanje i dešifrovanje. Zahvaljujući ovome, rutine za šifrovanje i dešifrovanje pretrpile su ovom prilikom samo neznatne izmjene.

## VI. IMPLEMENTACIONA ANALIZA

U uvodnom dijelu navedeno je da se algoritam AES teoretski može podjeliti na tri algoritma za šifrovanje/dešifrovanje: AES-128, AES-192 i AES-256. Međutim, ovaj algoritam sastoji se od još jednog veoma važnog dijela - od rutine za proširenje ključeva koja je zadužena za snabdjevanje ostalih rutina neophodnim bajtima ključa.

Ulazni podatak za ovu transformaciju je inicijalni ključ odgovarajuće dužine koji se u njoj proširuje na potreban broj bajta. Pošto navedena rutina ni na koji način ne zavisi ni od otvorenog teksta ni od šifrata, prošireni ključ se može unaprijed izračunati i/ili se njegovo izračunavanje može do određene mjere paralelizovati.

Prikazana verzija AES algoritma nije u znatnijoj mjeri izmjenjena u odnosu na standardom definisani AES algo-

Listing 3: Dio međurezultata proizveden pri ekspanziji ključeva za 128, 64 i 32-bitni inicijalni ključ

```
A) Ključ 128 bita 000102030405060708090a0b0c0d0e0f
  |          | After    | After    |          | After    |           | key[i]=temp
  | temp     | RotWord  | SubWord  | Rcon     | xor Rcon | key[i-16] | xor key[i-16]
  |----------|----------|----------|----------|----------|-----------|--------------
  | 0C0D0E0F | 0D0E0F0C | D7AB76FE | 01000000 | D6AB76FE | 00010203  | D6AA74FD
  | D6AA74FD |          |          |          |          | 04050607  | D2AF72FA
  | D2AF72FA |          |          |          |          | 08090A0B  | DAA678F1
  | DAA678F1 |          |          |          |          | 0C0D0E0F  | D6AB76FE
  | D6AB76FE | AB76FED6 | 6238BBF6 | 02000000 | 6038BBF6 | D6AA74FD  | B692CF0B
  | B692CF0B |          |          |          |          | D2AF72FA  | 643DBDF1...
B) Ključ 64 bita 00010203040506O7
  |          | After    | After    |          | After    |           | key[i]=temp
  | temp     | RotWord  | SubWord  | Rcon     | xor Rcon | key[i- 8] | xor key[i- 8]
  |----------|----------|----------|----------|----------|-----------|--------------
  | 04050607 | 05060704 | 6B6FC5F2 | 01000000 | 6A6FC5F2 | F0FFE201  | 9A9027F3
  | 9A9027F3 |          |          |          |          | 9A9027F3  | 00000000
  | 00000000 | 00000000 | 63636363 | 02000000 | 61636363 | 00010203  | 61626160
  | 61626160 |          |          |          |          | 04050607  | 65676767
...
C) Ključ 32 bita 00010203
  |          | After    | After    |          | After    |           | key[i]=temp
  | temp     | RotWord  | SubWord  | Rcon     | xor Rcon | key[i- 4] | xor key[i- 4]
  |----------|----------|----------|----------|----------|-----------|--------------
  | 00010203 | 01020300 | 7C777B63 | 01000000 | 7D777B63 | E480E201  | 99F79962
  | 99F79962 | F7996299 | 68EEAAEE | 02000000 | 6AEEAAEE | F0FFE201  | 9A1148EF
  | 9A1148EF | 1148EF9A | 8252DFB8 | 04000000 | 8652DFB8 | 8652DFB8  | 00000000
...
```

formi. Jedina rutina koja je pretrpila određene izmjene je rutina za ekspanziju ključeva. U okviru ove transformacije se kod standardom definisanog AES 256, 192 i 128 bitnog algoritma svakih 32, 24 i 16 bajta poziva rutina keyExpBase. Zadatak rutine keyExpBase je da na trenutna četiri bajta ključa primjeni AES S-Box te da se prvi bajt pomoću xor operacije izmješa sa nizom RCon. U slučaju modifikovane verzije, rutina za proširenje 64-bitnog inicijalnog ključa poziva funkciju keyExpBase na svakih 8 bajta, dok je rutina za proširenje 32-bitnog inicijalnog ključa poziva za svaki bajt, kako je to prikazano u siječku međurezultata na listingu 3:

Na listingu 3 su prikazane tri sekcije međurezultata koje su proizvedene pri proširenju 128 bitnog, 64 bitnog i 32 bitnog inicijalnog ključa. Da bi rješenje ostalo što je više moguće u skladu sa standardom definisanim algoritmom, u slučaju 32 bitne funkcije proširenja funkciju keyExpBase se poziva za svaki bajt inicijalnog ključa.

Modifikovana verzija algoritma AES je implementirana na jasan i razumljiv način korišćenjem dva programska jezika (Java i Pascal). U toku ove implementacije nisu korišćene nikakve optimizacije koda da bi on ostao što čitljiviji i razumljiviji. Na ovaj način su stvoreni preduslovi da ovakva verzija algoritma bude iskorišćena kao uvod i za upoznavanje sa standardom definisanim AES algoritmom.

## VII. ZAKLJUČAK

U ovom tekstu je prikazana modifikovana verzija AES algoritma koja može biti iskorišćena u nastavnom procesu za potrebe demonstracije algoritma. Modifikovana verzija algoritma implementirana je tako da su njeni osnovni elementi pojednostavljeni i smanjeni, pri čemu je sačuvana originalna struktura ovoga algoritma. Sama implementacija je provedena funkcioniše biće u stanju da bez većih problema pređu na savladavanje pravog algoritma AES.

Treba primjetiti i da u dostupnoj literaturi nije pronađeno da je bilo pokušaja implementacije redukovanih ni pojednostavljenih verzija bilo kojeg kriptografskog algoritma na prostorima Srbije i Republike Srpske, čime obrazovni tekstovi sa ovakvom tematikom na srpskom jeziku dobijaju na dodatnoj težini.

## ZAHVALNOST

Ovaj rad je deo projekta Primena multimodalne biometrije u menadžmentu identiteta, finansiranog od strane Ministarstva Prosvete i Nauke Republike Srbije, pod zavodnim brojem TR-32013.

## REFERENCE

[1] Berent A., ABI Software Developement, AES (Advanced Encryption Standard) Simplified, dostupno na: http://www.ime.usp.br/~rt/cranalysis/AESSimplified.pdf (april 2013)
[2] Bertoni G., Breveglieri L., Fragneto P., Macchetti M., Marchesin S.: Efficient Software Implementation of AES on 32-Bit Platforms. CHES 2002: 159-171 (2002)
[3] Biryukov A., Dunkelman O., Keller N., Khovratovich D., Shamir A., Key recovery attacks of practical complexity on AES-256 variants with up to 10 rounds, EUROCRYPT'10 Proceedings of the 29th Annual international conference on Theory and Applications of Cryptographic Techniques, Pages 299-319, (2010)
[4] Chung-Wei Phan R., Impossible Differential Cryptanalysis of Mini-AES, Cryptologia , Vol. XXVII, No. 4, October (2003)
[5] Chung-wei Phan R., Mini Advanced Encryption Standard (Mini-AES): A Testbed for Cryptanalysis, Cryptologia, 26(4):283–306, (2002)
[6] Cid C., Murphy S. and Robshaw M., Algebraic aspects of the Advanced encryption standard, Springer Science-Business Media, LLC. (2006)

[7] Daemen J., Rijmen V., The design of Rijndael, Springer-Verlag, Inc. (2002)
[8] Damjanovic B., AES Algorithm, (Java izvorni kod), dostupno na: http://code.google.com/p/crypto2, (april 2013)
[9] Damjanović B., Implementacija i proširenje AES algoritma, Master rad, Univerzitet u Beogradu - Fakultet organizacionih nauka (2010)
[10] Damjanović B., Simić D., AES algorithm for student education (Delphi izvorni kod) dostupno na: http://code.google.com/p/crypto1, (april 2013)
[11] Demirci H., Selçuk A. A., A Meet-in-the-Middle Attack on 8-Round AES, Fast Software Encryption, Springer-Verlag Berlin, Heidelberg, (2008)
[12] Doganaskoy A., Darbuka A., Özberk D., Öztop N., Sulak F., A Survey of the Attacks on AES, Proceedings of the 3rd INFORMATION SECURITY & CRYPTOLOGY CONFERENCE WITH INTERNATIONAL PARTICIPATION, Ankara, (2008)
[13] Dunkelman O., Sekar G., Preneel B., Improved Meet-in-the-Middle Attacks on Reduced-Round DES, INDOCRYPT'07 Proceedings of the cryptology 8th international conference on Progress in cryptology Pages 86-100 (2007)
[14] Gladman B., A Specification for Rijndael, the AES Algorithm, dostupno na: http://gladman.plushost.co.uk/oldsite/cryptography_technology/rijndael/, (april 2013)
[15] Gordon S., Block Ciphers and DES Examples, Security and Cryptography, Sirindhorn International Institute of Technology (SIIT) in Thammasat University, Bangkok, Thailand, dostupno na: http://ict.siit.tu.ac.th/~steven/css322/unprotected/CSS322Y12S2H01-DES-Examples.pdf
[16] Gordon S., Simplified AES Example, Security and Cryptography, Sirindhorn International Institute of Technology (SIIT) in Thammasat University, Bangkok, Thailand, dostupno na: http://hw.siit.net/files/001283.pdf (april 2013)
[17] IBM DeveloperWorks, Robust Java benchmarking, dostupno na: http://www.ibm.com/developerworks/java/library/j-benchmark1.html, (april 2013)
[18] Interna dokumentacija, Univerzitet u Beogradu – Fakultet organizacionih nauka (2010)
[19] Manangi S.J., Chaurasia P., Singh M.P., Simplified AES for Low Memory Embedded Processors, Global Journal of Computer Science and Technology, Vol. 10 Issue 14 (2010)
[20] Matsui M., Linear Cryptanalysis Method for DES cypher, EUROCRYPT '93 Workshop on the theory and application of cryptographic techniques on Advances in cryptology, Pages 386-397 (1994)
[21] Michael D. Fried, Moshe Jarden, „Field arithmetic", Third Edition, Springer-Verlag (2008)
[22] Mollin R. A., An Introduction to Cryptography, Second Edition, Taylor & Francis, 2010
[23] National Institute of Standards and Technology (http://csrc.nist.gov/)
[24] Nover H., Algebraic Cryptanalysis of AES: An Overview, dostupno na: http://www.math.wisc.edu/~boston/nover.pdf, (april 2013)
[25] Ooi K. S. , Chin Vito B., Cryptanalysis of S-DES, IACR Cryptology ePrint Archive 2002: 45, (2002)
[26] Schaefer E., A Simplified Data Encryption Standard Algorithm, Cryptologia 96, (1996)
[27] Schneier B., A Self-Study Course In Block-Cipher Cryptanalysis, Cryptologia, Volume 24 Issue 1, Jan. 2000 , Pages 18 - 33 (2000)
[28] Sharma L., Pathak B. K., Sharma R, Breaking of Simplified Data Encryption Standard Using Genetic Algorithm, Global Journal of Computer Science and Technology Volume XII Issue V Version I, (2012)
[29] Simmons S., Algebraic Cryptanalysis of Simplified AES, Cryptologia, Volume 33, Issue 4, (2009)
[30] Specification for the Advanced Ecryption Standard (AES), Federal Information Processing Standards Publication 197, dostupno na: http://csrc.nist.gov/publications/, (2001)

M. Sc. Boris Damjanović,
Kontakt: damjanovic@koledzprijedor.org
Oblasti interesovanja: zaštita računarskih sistema, kriptografija, informacioni sistemi, interakcija čovek-računar, programiranje i jezici, paralelizacija koda

Dr Dejan Simić, redovni prof., FON
Kontakt: dsimic@fon.bg.ac.rs
Oblasti interesovanja: elektronski sistemi plaćanja, primena kriptografije, zaštita podataka i računarskih sistema, PCI standardi, informacione tehnologije i njihova primena

## UPUTSTVO ZA PRIPREMU RADA

1. Tekst pripremiti kao Word dokument, A4, u kodnom rasporedu 1250 latinica ili 1251 ćirilica, na srpskom jeziku, bez slika. Preporučen obim – oko 10 strana, single prored, font 11.
2. Naslov, abstakt (100-250 reči) i ključne reči (3-10) dati na srpskom i engleskom jeziku.
3. Jedino formatiranje teksta je normal, bold, italic i bolditalic, VELIKA i mala slova (tekst se naknadno prelama).
4. Mesta gde treba ubaciti slike, naglasiti u tekstu (Slika1...)
5. Slike pripremiti odvojeno, VAN teksta, imenovati ih kao u tekstu, radi identifikacije, u sledećim formatima: rasterske slike: jpg, tif, psd, u rezoluciji 300 dpi 1:1 (fotografije, ekranski prikazi i sl.), vektorske slike – cdr, ai, fh,eps (šeme i grafikoni).
6. Autor(i) treba da obavezno priloži svoju fotografiju (jpg oko 50 Kb), navede instituciju u kojoj radi, kontakt i 2-4 oblasti kojima se bavi.
7. Maksimalni broj autora po jednom radu je 5.

Redakcija časopisa Info M