Zbornik radova Fakulteta tehničkih nauka, Novi Sad

UDK: 621.38
DOI: https://doi.org/10.24867/01BE16Stanic

WI-FI KRIPTOPROCESOR

WI-FI CRYPTOPROCESSOR

Stefan Stanić, *Fakultet tehničkih nauka, Novi Sad*

Oblast – ELEKTROTEHNIKA I RAČUNARSTVO

Kratak sadržaj –*Ovaj rad ima za cilj da upozna čitaoca sa CCMP kriptografskim protokol korišćenim u mnogim WiFi sistemima, kao i da dokumentuje i opiše projektovani sistem koji implementuje ovaj protokol u hardveru i softveru.*

Abstract –*This paper strives to provide the reader with a basic introduction to the CCMP cryptographic protocol, in addition to documenting the accompanying system that implements the protocol in question in hardware and software.*

Ključne reči: *AES, CCMP, HLS, WiFI.*

1. UVOD

CCMP-a (Counter mode Cipher block chaining Message authentication code Protocol, alternativno Counter mode CBC-MAC Protocol ili jednostavno CCM mode Protocol) je prvobitno imao namenu da unapredi i nasledi WEP (Wired Equivalent Privacy) enkripcioni protokol korišćen u 802.11 sistemima pre dodatka 802.11i amandmana. CCMP definiše metod enkripcije kao AES, mod operacije kao CCM (Counter mode CBC-MAC) i enkapsulaciju paketa kao dodavanje CCMP zaglavlja i MIC vrednosti.

Implementacija sistema je izvršena koristeći se Cadence-ovim Stratus HLS paketom alata.

2. PROCES DIZAJNA

Proces dizajna koristeći Stratus paket alata je dat na slici 1. Početni model na višem nivou apstrakcije se piše ili modifikuje tako da se sastoji samo od sintezibilnog podskupa SystemC naredbi [1], ovo predstavlja bihevioralni model sistema koji se dizajnira [2]. Dodatno se konstruiše Testbench koji ce se dalje koristiti za validaciju i debagovanje svih generisanih modela uključujući i trenutni bihevioralni model.

Sledeći korak je definisanje HLS parametara kao sto su frekvencija clock-a, clock slack i tip memorija koje se koriste. Dodatno ovo podrazumeva definisanje ASIC tehnologije koja će se koristiti kao i dodavanje HLS direktiva u SystemC kod bihevioralnog modela [2]. Ove direktive služe da definišu željene osobine konačnog RTL modela, primer često korišćene direktive je direktiva koja definiše vreme potrebno za izvršavanje određenog dela koda ili direktive koje definišu koji nizovi predstavljaju memorije a koji registre ili registarske banke.

NAPOMENA:
Ovaj rad proistekao je iz master rada čiji mentor je bo prof. dr Rastislav Struharik.

Pri definisanju potrebnih direktiva pokreće se Stratus HLS alat za sintezu koji na osnovu SystemC koda, HLS direktiva kao i specificirane ASIC tehnologije sintetiše RTL model. Na osnovu karakteristika generisanog RTL modela se odlučuje da li je RTL dizajn zadovoljavajući, u slučaju da nije direktive se modifikuji i siteza se pokreće ponovo. Ovaj korak se ponavlja sve dok se ne postigne RTL model zadovoljavajućeg kvaliteta. Bitno je spomenuti da se prilikom ocene kvaliteta RTL modela može koristiti pre konstruisan Testbench za bihevioralni model u nepromenjenoj formi kako Stratus ima mogućnost korišćenja odgovarajućih wrapper-a za povezivanje SystemC testbench-a i sintetisanog RTL koda [2].

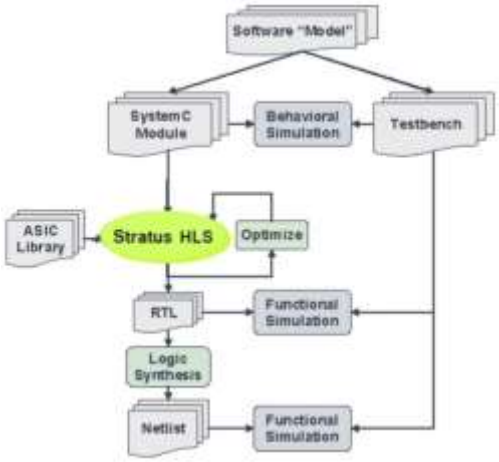
Generisani RTL kod se dalje koristi za logičku sintezu netliste. Ova netlista se takođe može validirati i debagovati koristeći početni Testbench kako i u ovom slučaju Stratus ima mogućnost generisanja potrebnih wrapper-a. Alternativno Stratus se može povezati sa eksternim RTL sintetizerima ili procesima dizajna [2]. U tom slučaju generisani RTL predstavlja kraj Stratusovog procesa dizajna i ovaj korak označava poziv nekog drugog RTL-to-GDSII procesa dizajna (GDSII predstavlja standardizovan format layout-a integrisanih kola).

Osim osnovnih prednosti HLS metodologije tj. značajno kraćem procesu dizajna kao i velikoj slobodi izmene arhitekture, Stratus paket alata pruža mogućnost korišćenja jedinstvenog Testbench-a na svim modelima nezavisno od nivoa abstrakcije koristeći odgovarajuće wrapper-e. Dodatno ovaj paket pruža gotove biblioteke interfejsa kao i floating point operacija veoma visokog stepena optimizacije [2].

Kako je deo CCMP algoritma implementiran u i softveru i hardveru postoji potreba za validacijom i testiranjem CCMP sistema u celosti. Ovo je izvedeno korišćenjem Cadence-ovog VSP (Virtual System Platform) alata. Ovaj alat ima mogućnost kosimulacije hardvera i softvera. SystemC kod hardvera se simulira u TLM (Transaction-Level Model) modu i to omogućava veoma brze simulacije, dovoljno brze da se ceo sistem, uključujući i softver, može efikasno simulirati [3].

3. AES

AES algoritam se bazira na principima permutacije, substitucije i linearne transformacije i prvobitno je bio opisan od strane NIST (National Institute of Standards and Technology) agencije Sjedinjenih Američkih Drzava u FIPS (Federal Information Processing Standards) 197 publikaciji[4]. Kako se ove 3 osnovne operacije veoma efikasno implementiraju u hardveru kao i u softveru ta činjenica predstavlja glavnu prednost AES algoritma nad njegovim konkurentima kao sto je DES.



Slika 1. *Stratus proces dizajna[2]*

Ovaj algoritam se sastoji iz iterativne primene određenih operacija nad trenutnim vrednostima state matrice. Ova matrica, prikazana na slici 2. podrazumeva stanje unutrasnjih podataka koji se menjaju izvrsavanjem svake od definisanih operacija.

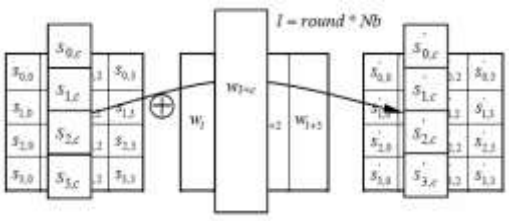
$$\begin{bmatrix} b_0 & b_4 & b_8 & b_{12} \\ b_1 & b_5 & b_9 & b_{13} \\ b_2 & b_6 & b_{10} & b_{14} \\ b_3 & b_7 & b_{11} & b_{15} \end{bmatrix}$$

Slika 2. *State matrica*

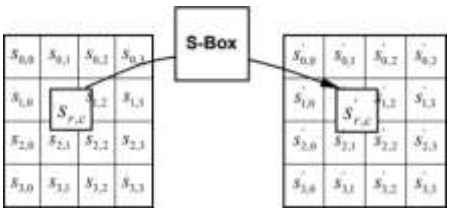
Kao sto se vidi sa slike 3 koja prikazuje pseudo kod AES algoritma primecuje se da se algoritam bazira na 4 osnovne operacije ilustrovane na slikama 4, 5, 6 i 7.

```
Cipher(byte in[4*Nb], byte out[4*Nb], word key_schedule[Nb*(Nr+1)])
begin
    byte state[4,Nb]
    state = in

    AddRoundKey(state, key_schedule[0, Nb-1])

    for round = 1 step 1 to Nr-1
        SubBytes(state)
        ShiftRows(state)
        MixColumns(state)
        AddRoundKey(state, key_schedule[round*Nb, (round+1)*Nb-1])
    end for

    SubBytes(state)
    ShiftRows(state)
    AddRoundKey(state, key_schedule[Nr*Nb, (Nr+1)*Nb-1])

    out = state
end
```

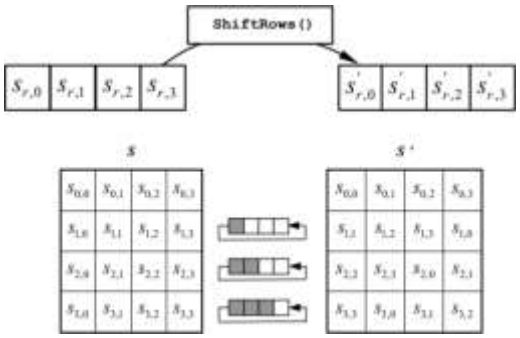
Slika 3. *Pseudo kod AES algoritma*



Slika 4. *Ilustracija AddRoundKeys operacije*



Slika 5. *Ilustracija SubBytes operacije*



Slika 6. *Ilustracija ShiftRows operacije*

$$S'_{0,c} = (\{02\} \bullet S_{0,c}) \text{ xor } (\{03\} \bullet S_{1,c}) \text{ xor } S_{2,c} \text{ xor } S_{3,c}$$

$$S'_{1,c} = S_{0,c} \text{ xor } (\{02\} \bullet S_{1,c}) \text{ xor } (\{03\} \bullet S_{2,c}) \text{ xor } S_{3,c}$$

$$S'_{2,c} = S_{0,c} \text{ xor } S_{1,c} \text{ xor } (\{02\} \bullet S_{2,c}) \text{ xor } (\{03\} \bullet S_{3,c})$$

$$S'_{3,c} = (\{03\} \bullet S_{0,c}) \text{ xor } S_{1,c} \text{ xor } S_{2,c} \text{ xor } (\{02\} \bullet S_{3,c}),$$

Slika 7. *Ilustracija MixColumns operacije nad c-tom kolonom*

Primecuje se da je za AES algoritam potreban key_schedule niz koji se generise putem KeyExpansion procedure opisane na slici 8. SubWord je naziv za SubBytes operaciju kojoj je operand jedna kolona dok RotWord predstavlja shift-ovanje bajtova reci za jedno mesto ka vise znacajnom bajtu. Rcon predstavlja niz konstantnih predefinisanih vrednosti prikazanih na slici 9.

```
KeyExpansion(byte key[4*Nk], word key_schedule[Nb*(Nr+1)], Nk)
begin
    word temp
    i = 0

    while (i < Nk)
        key_schedule[i] = word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])
        i = i+1
    end while

    i = Nk
    while (i < Nb * (Nr+1))
        temp = key_schedule[i-1]
        if (i mod Nk = 0)
            temp = SubWord(RotWord(temp)) xor Rcon[i/Nk]
        else if (Nk > 6 and i mod Nk = 4)
            temp = SubWord(temp)
        end if
        key_schedule[i] = key_schedule[i-Nk] xor temp
        i = i + 1
    end while
end
```

Slika 8. *Pseudo kod KeyExpansion procedure*

```
unsigned char Rcon[51] = {
    0x8d, 0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40,
    0x80, 0x1b, 0x36, 0x6c, 0xd8, 0xab, 0x4d, 0x9a,
    0x2f, 0x5e, 0xbc, 0x63, 0xc6, 0x97, 0x35, 0x6a,
    0xd4, 0xb3, 0x7d, 0xfa, 0xef, 0xc5, 0x91, 0x39,
    0x72, 0xe4, 0xd3, 0xbd, 0x61, 0xc2, 0x9f, 0x25,
    0x4a, 0x94, 0x33, 0x66, 0xcc, 0x83, 0x1d, 0x3a,
    0x74, 0xe8, 0xcb
}
```

Slika 9. *Vrednosti Rcon niza*

4. CCMP

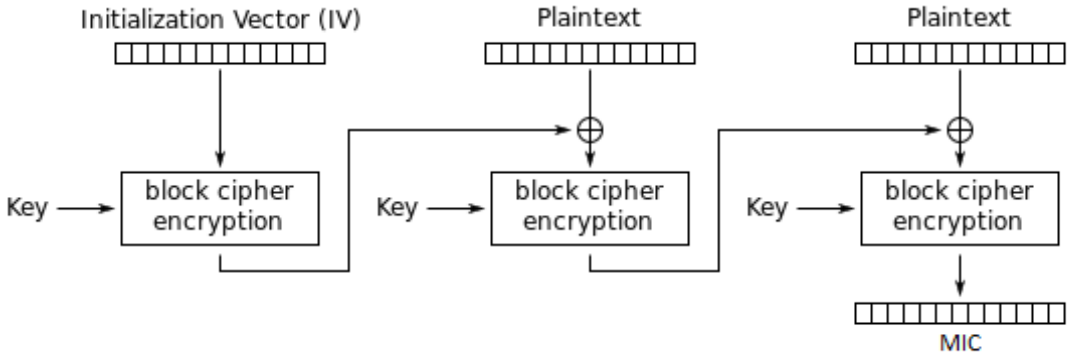
CCMP bazira se na CCM modu operacije AES enkripcionog metoda. Ovaj mod zahteva definiciju dva parametra kojima se definišu dužine određenih polja i vrednosti. Ovi parametri, L i M su definisani CCMP-om. Odabirom dužine kljuca AES enkripcionog metoda se definisu dve najčešće varijante CCMP-a tj CCMP-128 i CCMP-256 koje takodje predstavljaju odabir M parametra kao 8 ili 16 respektitivno dok L parametar ima fiksiranu vrednost 2 [5].

CCM mod operacije se sastoji iz dva dela. Prvi deo je zadužen za operaciju autentikacije tj provere integriteta poruke i koristi CBC-MAC mod operacije prikazan na slici 10 dok drugi deo pruža mogućnost enkripcije i dekripcije putem CTR moda operacije prikazanom na slici 11a i 11b respektitivno [5].

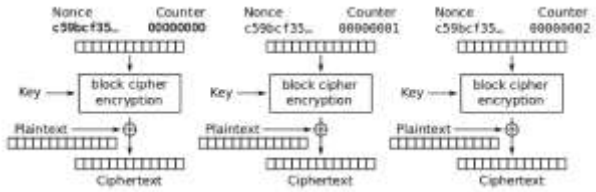
CCMP definise AAD (Additional Authentication Data) koji predstavlja podatke koji ce biti uzeti kao plaintext za CBC-MAC deo CCM moda ali ne i u CTR delu. Rezultat ovoga je da se ovi podaci samo autentikuju. AAD se konstruiše izbacivanjem Duration ID polja iz 802.11 MAC zaglavlja, kako je ovo polje promenljivo i nema potrebe da se autentikuje [5,6]. Konstrukcija AAD-a je prikazana na slici 12.

CCMP definise svoje zaglavlje koje je prikazano na slici 13. Ovo zaglavlje se smešta na početak rezultujućeg CCMP paketa i služi za transmisiju PN (Packet Number) vrednosti [5].

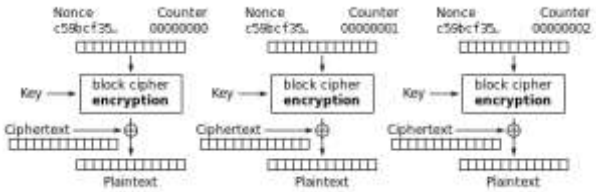
Nonce vrednost takođe definisana CCMP-om se koristi pri konstrukciji CTR brojača i CBC-MAC inicijalizacionog vektora [5]. Sadržaji nonce vrednosti i CBC-MAC inicijalizacionog vektora su prikazani na slikama 14 i 15.



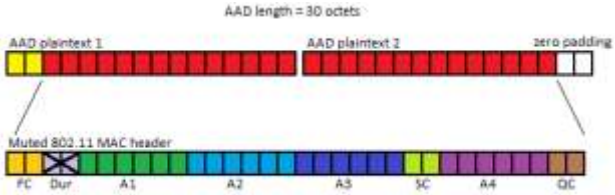
Slika 10. *CBC-MAC mod operacije[7]*



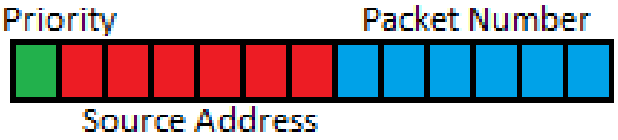
Slika 11a. *CTR mod operacije, enkripcija[7]*



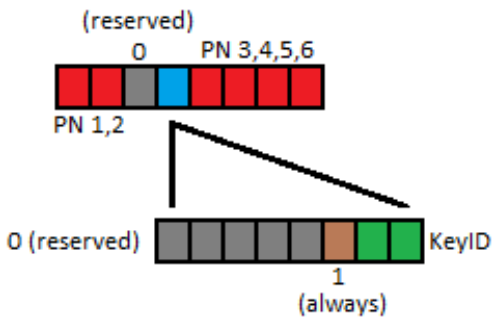
Slika 11b. *CTR mod operacije, dekripcija[7]*



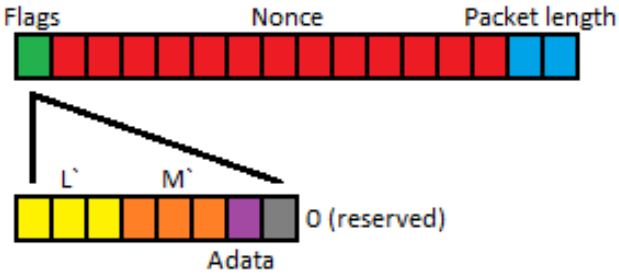
Slika 12. *Konstrukcija AAD-a*



Slika 14. *Konstrukcija nonce vrednosti*



Slika 13. *Konstrukcija CCMP zaglavlja*



Slika 15. *Konstrukcija CBC-MAC inicijalizacionog vektora*

5. IMPLEMENTACIJA

AES modul tj. AES 256 je sistem koji izvršava AES enkripciju nad dva ulazna podatka tj. operanda. Sistem takođe ima mogućnost izvršavanja procedure razvijanja ključa tj KeyExpansion procedure, za veličine ključa od 128 ili 256 bitova. Ova procedura će interno sačuvati raspored kljuceva tj key_schedule, i kasnije, pri operaciji enkripcije, koristiti jednistveni key_schedule za oba operanda, što podrazumeva jedinstvenu key_schedule RAM memoriju.

Glavni izazov implementacije ovog modula sistema je bio paralelizacija potrebnih pristupa memorijama. Tokom pre spomenute KeyExpansion procedure u key_schedule RAM se upisuju vrednosti. Za AddRoundKeys operaciju je potrebno izvršiti 4 citanja iz tog RAM-a sto znaci da je za jednu rundu AES-a minimalno potrebno 4 takta. Osim pristupa key_schedule RAM-u SubBytes operacija se sluzi pristupom S-box ROM memoriji tj memorijama. Sadrzaj ovih ROM memorija je konstantan i moze se paralelizovati pod pretpostavkom da su nam sve potrebne adrese ovih pristupa poznate što jesu. SubBytes operacija je implementirana tako da tokom pristupa key_schedule RAM-u stigne da izvrši svojih 16 pristupa S-Box ROM memorijama. Kao je pristup RAM-u 4 takta potrebno su 4 instance ROM memorija po operandu AES operacije.

WiFi kriptoprocesorski sistem je namenjen da uz pomoć softvera za formatiranje izvrsi CCMP operaciju nad podacima koji se nalaze u memoriji na koju je povezan, rezultat ove operacije smesti u tu memoriju kao i određenim Interrupt signalom javi sirem sistemu da je operacija završena.

Interfejs ka softverskom delu predstavlja AXI4 Lite magistrala putem koje se, pre pokretanja operacije, upisom u interne registre sistema definišu određeni parametri CCMP operacije kao sto su enkripcioni ključ, AAD, nonce, veličine i adrese ulaznih i izlaznih podataka [8]. Pri završetku ovih upisa operacija se pokreće upisom na određenu adresu ove magistrale.

Kao sto je pre spomenuto hadver ovog sistema ne izvršava CCMP u celosti iz ovog razloga je kontrolnom softveru potrebno dodati određene operacije kao što su formatiranje AAD podataka i upis CCMP zaglavlja na odgovarajuće mesto u memoriji.

Interfejs sa memorijom je implementiran putem AXI3 magistrale. Ne uzimajući u obzir kašnjenje pristupa sistemskoj memoriji kašnjenje ovog sistema je u velikoj meri određeno kašnjenjem AES operacija CCM moda. Činjenica da su kašnjenja prilikom pristupa sistemskoj memoriji daleko od zanemarljivih je podržala ideju da se pristupi memoriji tako vremenski rasporede da se uticaj ovih kašnjenja na kašnjenje cele operacije umanji. Ovo je izvršeno koristeći burst funkcionalnost AXI3 magistrale kao i implementacijom ovih pristupa tako da se oni dešavaju između AES operacija koje imaju fiksno kašnjenje definisano AES 256 DUAL modulom [8]. Kako je kašnjenje AES operacije značajno veće od kašnjenja idealnog pristupa sistemskoj memoriji, ovaj pristup ima olakšano minimalno vreme pristupa za postizanje minimalnog kašnjenja. Ovo znači da ce se i pri manjim zastojima usled zauzetka pristupa sistemskoj memoriji postići idealno kašnjenje CCMP operacije.

Ocekivana povrsina sistema je pri pocetku projekta iznosila 35.000 um^2. Vremensko ograničenje je definisano nad podatkom dužine 512 bajtova, podrazumevajuci frekvenciju sistemskog takta od 32 MHz i iznosi 160 us. Konacna površina sistema od 41.600 um^2 je zadovoljavajuća iako u maloj meri ne odgovara početnoj proceni dok su zahtevi brzine u potpunosti ispunjeni maksimalnim vremenom operacije od 73.25 us.

Kako je vreme izvršavanja značajno manje od zahtevanog, može se postaviti pitanje optimalnosti sistema jer je površina prevazišla procenu. Razlog iza ovog disbalansa jeste da su delovi sistema koji najviše utiču na ove karakteristike različiti. Na vreme izvršavanja skoro ekskluzivno utice AES sistem dok oko 75% površine sistema zauzimaju ostali delovi sistema. Upravo zbog toga je dalja optimizacija nemoguća kako bi smanjenjem površine bilo uvedeno neprihvatljivo kašnjenje.

6.ZAKLJUČAK

Kriptoprocesor je razvijen u cilju da bude integrisan u WiFi komunikacionim rešenjima kompanije Methods2Business. Kao deo ovih rešenja on je u potpunosti validiran kao deo WPA2 (WiFi Protected Access 2) sistema.

Pre opisanim optimizacijama su uspešno postignuti projektni ciljevi brzine rada kao i površine sistema. Očekuje se da ovo rešenje bude unapređeno podrškom za GCMP (Galois Counter Mode Protocol) kako postoji mogućnost da on bude prihvaćen kao brža alternativa CCMP-a u nasledniku WPA2 standarda tj. WPA3.

7. LITERATURA

[1] Ashfaq Khan, i drugi (Januar 2015). SystemC Synthesizable Subset, Version 1.4 Draft. Accellera Systems

[2] Stratus HLS User Guide (2017). Cadence

[3] Virtual System Platform User Guide (2016). Cadence

[4] NIST (November 26, 2001). FIPS Publication 197: Announcing the Advanced Encryption Standard (AES). Arhiviran original iz Marta 12. 2007.

[5] Whiting, i drugi (2003). Counter with CBC-MAC (CCM). The Internet Society

[6] IEEE Std 802.11ah-2016 (2016). IEEE Standards.

[7] Block cipher mode of operation (2013). Wikipedia. Dostupno na: https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation. Pristupljeno 17. Juna 2018.

[8] AMBA AXI and ACE Specification (2011). ARM.

Kratka biografija:

Stefan Stanić rođen je 22.8.1994. u Novom Sadu. Diplomirao je na Fakultetu tehničkih nauka u Novom Sadu, na Katedri za elektroniku u septembru 2017. godine.