UNIVERZITET UNION

RAČUNARSKI FAKULTET

Prenos podataka, bezbednost i energetska efikasnost internet stvari (Internet of things) rešenja zasnovanih na tehnologiji bežičnih senzorskih mreža

Doktorska teza

Mentor:

Dr Đorđe Babić

Kandidat:

Nemanja Radosavljević

Beograd, 2020.

APSTRAKT

Glavni doprinos ove doktorske teze jeste model procene potrošnje energije bežičnih senzorskih mreža za zadati skup protokola topologije i kriptografskih algoritama male računske i implementacione zahtevnosti. Predložena metoda procene potrošnje energije analizira se u simulacionom scenariju za nekoliko protokola i kriptografskih algoritama. Matematički model se testira u simulacionom okruženju gde se bežične senzorske mreže primenjuju kao tehnologija za komunikaciju na otvorenom području između senzorskih čvorova i uređaja zasnovanih na tehnologiji internet stvari. Rezultati simulacije potvrđuju formulu predloženu za procenu snage. Model predložen u ovoj tezi može se koristiti za odabir odgovarajućeg protokola topologije i kriptografskog algoritma. Na osnovu definisanog modela i rezultata simulacije predstavljena je sortirana lista kombinacija protokola topologije i algoritma šifrovanja. Najbolji rezultati u smislu potrošnje energije dobijaju se korišćenjem protokola A3 topologije i kriptografskog protokola *KATAN64*.

Ključne reči: topološki protokoli, kriptografski algoritmi male zahtevnosti, komunikacija na otvorenom, sigurnost bežične senzorske mreže, potrošnja energije u bežičnim senzorskim mrežama.

ABSTRACT

The main contribution of the doctoral thesis is a model for estimating the energy consumption of wireless sensor networks for a given set of topology protocols and cryptographic algorithms of low computational and implementation complexity. The proposed method of estimating energy consumption is analyzed in a simulation scenario for several protocols and cryptographic algorithms. We test a mathematical model in a simulation environment where wireless sensor networks are applied as an open field communication technology between sensor nodes and Internet of Things-based devices. The simulation results confirm the proposed power estimation formula. The proposed model can be used to select the appropriate topology protocol and cryptographic algorithm. Based on the defined model and simulation results, a sorted list of combinations of topology protocols and encryption algorithms is presented. The best results on energy consumption are obtained using the A3 topology protocol and the *KATAN64 cryptographic protocol*.

Keywords: topological protocols, low-complexity cryptographic algorithms, outdoor communication, wireless sensor network security, power consumption in wireless sensor networks.

Sadržaj

Akronimi

Akronim	***Engleski naziv***	***Srpski naziv***
AES	*Advanced Encryption Standard*	Napredni standard enkripcije
BSM	*Wireless sensor networks*	Bežične senzorske mreže
CDS	*Connected dominating set*	Dominantno povezan skup
CPU	*Central processor unit*	Centralna procesorska jedinica
EECDS	*Energy efficient connected dominating set*	Energetski efikasan dominantno povezan skup
GPRS	*General packet radio service*	Servis za prenos podataka
GSM	*Global system for mobile telecomunications*	Globalni sistem mobilne komunikacije
IMEI	*International mobile equipment identity*	Internacionalna identifikacija mobilnih uređaja
IoT	*Internet of Things*	Internet stvari
IT	*Information technologies*	Informacione tehnologije
LEACH	*Low energy adaptive clustering hierarchy*	Niskoenergetska prilagodljiva hijerarhija klasterovanja
LoRa	*Long range*	Veliki domet
MAC	*Media access control*	Kontrola pristupa mediju
MIS	*Maximal independent set*	Maksimalni nezavisni skup
OHC	*One-way Hash Chains*	Jednosmerni heš lanac
OT	*Operation technologies*	Operativne tehnologije
RF	*Radio-frequency*	Radio-frekvencija
RFID	*Radio-frequency identification*	Radio-frekvencijska identifikacija
SRIDR	*Source-initiated tree-based routing*	Rutiranje inicirano od izvora zasnovano na stablu
TTL	*Time to leave*	Vreme trajanja

Indeks slika

Indeks tabela

1. Uvod

Bežične senzorske mreže (BSM) već duže vreme privlače pažnju brojnih istraživača prvenstveno zbog onih njihovih karakteristika koje im omogućavaju realnu primenu, a pogotovo primenu u uređajima čije je funkcionisanje zasnovano na tehnologiji internet stvari. Zbog korišćenja u raznim uređajima, bežične senzorske mreže nalaze sve širu primenu u različitim oblastima, u saobraćaju, medicini, vojnoj industriji i u mnogim drugim privrednim granama, pri čemu se one prilagođavaju zadatku koji je potrebno izvršiti u aplikacijama. U narednom periodu očekuje se da će ova tehnologija naći veću primenu i u mnogim drugim oblastima. Među najznačajnije zadatke koje mreže obavljaju spadaju prikupljanje određenih informacija, beleženje podataka, njihovo očitavanje, nadgledanje i prepoznavanje događaja, a posebnu pažnju potrebno je posvetiti prenosu podataka kroz mrežu.

Budući da se tehnologija internet stvari zasnovana na komunikaciji putem bežičnih senzorskih mreža svrstava u savremene aktuelne tehnologije, ovoj problematici treba pristupiti iz različitih perspektiva, počevši od njihovog dizajna pa sve do razvoja i primene, posebno vodeći računa o potrebi za oslobađanjem neograničenog potencijala bežičnih senzorskih mreža i rešavanjem postavljenih izazova. Osnovni izazovi sa kojima se susrećemo prilikom komunikacije uređaja čije je funkcionisanje zasnovano na tehnologiji interneta stvari jesu energetska efikasnost senzorskih čvorova [1] i bezbednost prenosa podataka [2], [3], [4].

U ovom radu proučavamo principe bezbednosti bežičnih računarskih mreža sa stanovišta primene protokola i šifarskih algoritama. Naš cilj je da predložimo mehanizam za izbor optimalnog rešenja koje zadovoljava zadati nivo sigurnosti uz optimizovanu potrošnju energije.

Naučni doprinos disertacije ogleda se, pre svega, u matematičkom modelu procene potrošnje energije bežičnih senzorskih mreža za zadati skup protokola topologije i kriptografskih algoritama male računske i implementacione zahtevnosti. Predložena metoda procene potrošnje energije analizira se u simulacionom scenariju za četiri protokola za konstrukciju topologije i devet kriptografskih algoritama. Model je testiran u simulacionom okruženju gde se bežične senzorske mreže primenjuju kao tehnologija za

komunikaciju na otvorenom području između senzorskih čvorova i uređaja zasnovanih na tehnologiji interneta stvari. Pored matematičkog modela predstavljena je i njegova algoritamska implementacija koja je korišćena za nadogradnju simulacijonog okruženja *Atarraya*. Predloženi matematički model ima jednostavnu i efikasnu implementaciju. Analizom rezultata simulacije potvrdili smo predloženu formulu za koeficijent podobnosti protokola topologije i šifarskog algoritma. Na osnovu definisanog modela i rezultata simulacije predstavljena je sortirana lista koeficijenata podobnosti kombinacije protokola topologije i algoritma šifrovanja.

Jedan od važnih doprinosa teze jeste predložena arhitektura BSM. Predložena arhitektura BSM predlaže podelu po slojevima na kojima se posmatraju tipični napadi na BSM. Arhitektura bi trebalo da bude osnov za celovito rešenje pitanja bezbednosti i ne bavi se pojedinačnim napadima. Na osnovu analize količine sadržaja u BSM, kao i ponašanja senzorskih čvorova, potrebno je odabrati neki od već poznatih energetski efikasnih šifarskih algoritama koji su prilagođeni ponašanju senzorske mreže i na taj način rasteretiti čvorove i produžiti životni vek trajanja baterije a da pritom bezbednost ostane na zadovoljavajućem nivou.

Pored optimalnog izbora protokola topologije i šifarskog algoritma rezultati istraživanja prikazuju i najoptimalnija rešenja za najučestalije napade na BSM, bolje razumevanje postojećih problema u oblasti bezbednosti sa aspekta energetske efikasnosti, kao i pronalaženje efikasnih smernica za dalji razvoj sistema zasnovanih na BSM.

Članak u kome je razmatrana ova problematika i izložen glavni doprinos teze publikovan je u časopisu *Journal of internet technology* pod nazivom: *"Power Consumption Analysis Model in Wireless Sensor Network for Different Topology Protocols and Lightweight Cryptographic Algorithms*" [5].

2. Tehnologija internet stvari

2.1. Uvod

Razvojem interneta, koji danas suvereno vlada poslovnom i privatnom komunikacijom te samim tim nesporno igra veliku ulogu u svakodnevnoj komunikaciji između ljudi, došlo se do potrebe za komunikacijom različitih uređaja koje svakodnevno srećemo i koristimo u svome okruženju. Te uređaje koji međusobno komuniciraju nazivamo stvarima koje su povezane preko interneta – internet stvari (IoT – engl. *Internet of Things*). Internet kao skup uređenih protokola i servisa koji međusobno komuniciraju osnov je za ostvarivanje komunikacije između „stvari". IoT tehnologija služi da omogući povezivanje bilo čega što ima pristup internetu. Povezivanjem na internet, uređaj dobija svojstva pametnog uređaja. U svakodnevnom životu koristimo pametne uređaje kao što su pametni telefoni, pametni satovi, međutim, pod pojmom IoT smatramo i pametne kuće, pametne frižidere, pametne usisivače, kao i mnoge druge kućne uređaje i industrijske aparate sa softverskim upravljanjem. Razvojem IoT-a dolazimo do graničnog slučaja gde svaki uređaj može da ima pametna svojstva i da se celo okruženje pretvori u internet svega (engl. *Internet of Everything*) [6], [7], [8].

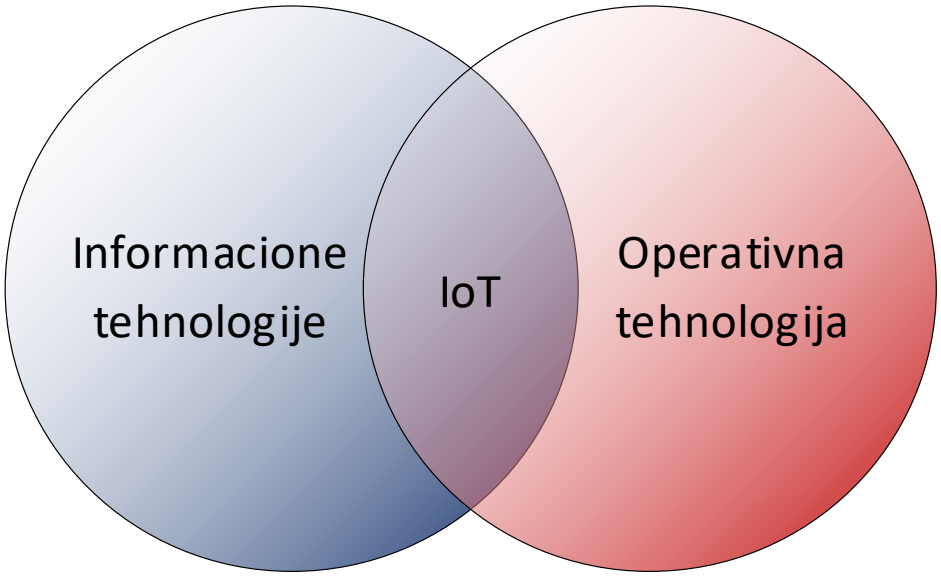
Ukoliko bi IoT trebalo predstaviti matematičkom formulom, ona bi prema navodima Alhafida (Alhafidh) i Alena (Allen) u radu iz 2016 [9] glasila :

IoT= Senzor + Mreža + Podaci + Servisi.

U svom radu oni su prikazali [9] glavna polja primene, međutim, od tada su se polja primene raširila i na mnoge druge oblasti, gde ova tehnologija ima značajnu primenu [7], [10]. Neki od primera primene prikazani su na slici 2.1.

Slika 2.1. Polja primene IoT

IoT su fizički entiteti koji komuniciraju na digitalni način. Njihova komunikacija ima za cilj da donese neku novu vrednost u međusobnoj razmeni informacija. IoT se može jednostavno objasniti Venovim dijagramom [7], kao što je prikazano na slici 2.2, gde se dva različita skupa, nazvana informaciona tehnologija (IT) i operativna tehnologija (OT), ujedinjuju kako bi formirali internet stvari. IT predstavlja skup pravila za komunikaciju, dok OT obuhvata senzore, mikrokontrolere i drugu opremu.

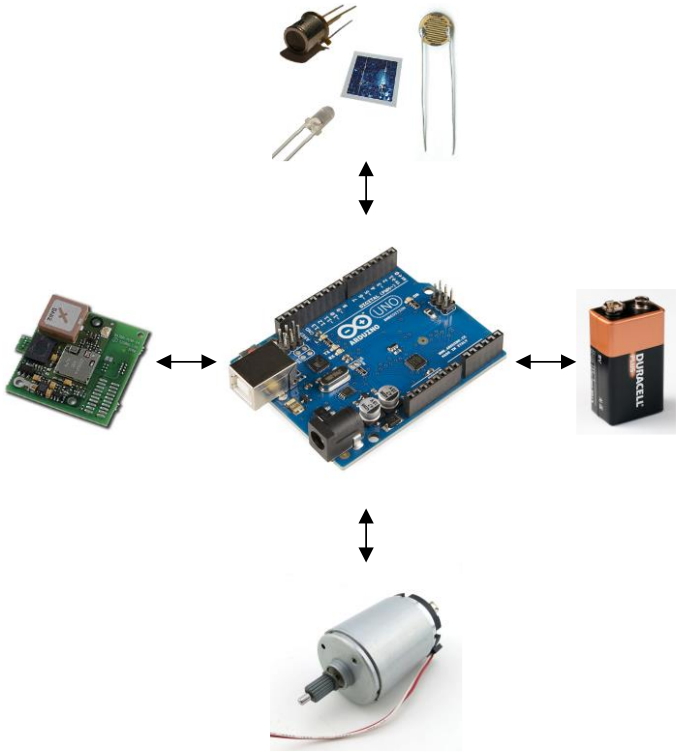


Slika 2.2. Venov dijagram koji opisuje prirodu IoT

Postoji mnoštvo tehnologija koje se koriste za komunikaciju između uređaja kao sto su wi-fi, Bluetooth, RFID i bežična senzorska mreža. U daljem radu ćemo se detaljnije baviti tehnologijom bežičnih senzorskih mreža kao jednom od tehnologija na kojoj je zasnovana komunikacija pametnih uređaja ili stvari [7].

2.2. Pametni uređaji

Pametni uređaji moraju da imaju mogućnost donošenja neke odluke. Da bismo za neki uređaj rekli da je pametan, on mora da obradi ulazne vrednosti, da na osnovu njih donese neku odluku, i na osnovu te odluke pokrene neki mehanizam koji će preduzeti neku akciju [7]. Da bi ovo mogao da uradi, pametni uređaj mora da se sastoji od: senzora, CPU, memorije, aktuatora, komunikacionog modula i napajanja. Na slici 2.3. možemo da vidimo prikaz arhitekture jednog pametnog uređaja [7].



Slika 2.3. Komponente pametnog objekta

2.2.1. Prikupljanje podataka

Senzori u pametnim uređajima imaju ulogu da prate promene u okolini. Promene koje senzori prate mogu da budu od jednostavnih kao što su temperatura, vlažnost, pH vrednost, dim, zvuk ili kompleksni, kao, na primer, termalna kamera i slični. Senzori koji se nalaze u uređajima služe da detektuju promene posmatranih vrednosti u svojoj okolini. Senzori, bili analogni ili digitalni, očitanu vrednost u digitalnom obliku do ostalih komponenata prenose IoT uređaju na dalju obradu [7].

2.2.2. Obrada podataka

IoT uređaji imaju CPU koji služi za obradu prikupljenih podataka. Mikrokontroleri se koriste za obradu i procesuiranje podataka. U zavisnosti od kompleksnosti operacija koje treba obraditi, zavisi i sam izbor mikrokontrolera. Zahvaljujući implementiranoj i programiranoj logici, mikrokontroler je u stanju da donosi odluke na osnovu podataka koje dobija očitavanjem na senzoru [7], [11]

2.2.3. Skladištenje podataka

Da bi jedan IoT uređaj mogao nezavisno da funkcioniše, mora da ima integrisan i neki memorijski prostor. Količina i brzina memorije za IoT uređaje ne poredi se sa savremenim sistemima za skladištenje podataka. Osnovna uloga memorije je da sadrži

program koji se izvršava. Pored sistemskog programa koji kontroliše rad uređaja, zadatak memorije je da sačuva očitane vrednosti sa senzora, privremeno ili trajno čuva vrednosti koje se koriste tokom računskih operacija, kao i da uskladišti vrednosti koje su dobijene nakon obrade [7], [11].

Pored navedenih stvari memorija služi i da čuva jedinstveni identifikacioni broj uređaja, koji bi bio nešto poput MAC adrese u mrežnim tehnologijama, ili kao IMEI broj u mobilnoj telefoniji. Ukoliko bi se povukla analogija sa primerima iz svakodnevnog života, to bi bio matični broj za čoveka, ili registarski broj za auto. Jedinstvena identifikacija svakog uređaja na internetu osnov je za bezbednu i nedvosmislenu komunikaciju [7], [8]..

2.2.4. Prenos podataka

Svi IoT uređaji komuniciraju međusobno, jer im je razmena informacija jedna od osnovnih uloga – da komuniciraju radi razmene informacija. Komunikacija se odvija između dva IoT uređaja ili preko interneta sa nekim drugim spoljnjim servisom. Urađaj može kablom da bude povezan na mreži, međutim, povezivanje uređaja bežičnim putem donosi veću fleksibilnost, mobilnost, jednostavnost u implementaciji i druge prednosti. Jedan od glavnih preduslova bežične komunikacije jeste dobro postavljena i uvek dostupna bežična infrastruktura. Bez obzira koji je vid komunikacije, žični ili bežični, među učesnicima moraju da se održavaju uspostavljeni protokoli o komunikaciji [7], [12].

2.2.5. Pokretanje sistema (akcije)

Senzori prikupljaju podatke, deo za obradu podataka analizira podatke i donosi neku odluku, nakon čega je potrebno tu odluku sprovesti u delo. Aktuator je deo uređaja koji na osnovu signala dobijenog od mikrokontrolera pokreće neku akciju koja je vidljiva u stvarnom (fizičkom) svetu. Jedan uređaj, u zavisnosti od zahteva, može da ima više aktuatora. Aktuator kontroliše rad nekog drugog uređaja [7], [13]

2.2.6. Napajanje sistema

Da bi ovi pametni uređaji mogli da rade, potrebno je obezbediti adekvatno napajanje. U tu svrhu mogu da se koriste sledeći izvori napajanja: električna mreža, baterije, obnovljivi izvori energije (vetar, voda, solarna energija) itd. U zavisnosti od oblasti primene konkretnog uređaja, zahtevi za energijom su različiti, ali uvek moramo voditi računa o energetskoj efikasnosti datog uređaja, na šta posebno moramo obratiti pažnju u slučajevima ograničenog izvora napajanja, kao što je baterijsko napajanje.

Dodatni slučaj u kojem je vrlo važno povesti računa o potrošnji energije jeste kombinacija baterijskog napajanja sa pozicijom uređaja koja nam ne dozvoljava laku, a katkad ni nikakvu zamenu baterija. Ovo je posebno bitno u tehnologiji bežičnih senzorskih mreža na kojima se jednim velikim delom zasnivaju IoT [7], [14].

2.2.7. Izazovi u savremenim IoT sistemima

Glavni izazov sa kojim se susrećemo u savremenim IoT sistemima jeste svakodnevno unapređenje svake od pojedinačnih komponenata sistema. Kako su glavne komponente zadužene za prikupljanje, obradu, skladištenje i prenos podataka, kao i pokretanje akcija i napajanje sistema, svaka od ovih komponenata ima svoje specifičnosti. Tok operacija prikazan je na slici 2.4. [7].

Slika 2.4. Veza između senzora i aktuatora

Prikupljanje podataka – senzori zaduženi za prikupljanje podataka imaju osnovni izazov da budu što manji a da pritom zadrže karakteristike koje garantuju preciznost očitanih podataka. Senzori koje danas možemo sresti mogu da budu toliko mali da nisu ni vidljivi golim okom, međutim ovo se još ne sreće u svakodnevnoj praksi [7].

Obrada podataka – procesorska snaga koja je zadužena za obradu podataka i donošenje odluka u svim sferama računarstva povećava se iz dana u dan pa tako i u IoT uređajima operacije koje treba izvršiti svakim danom sve su kompleksnije [7].

Skladištenje podataka – memorija potrebna za skladištenje podataka mora da bude što veća, u zavisnosti od konkretne namene uređaja ali ne sme imati preveliki uticaj na veličinu samog uređaja, kao ni na njegovu energetsku efikasnost [7].

Prenos podataka – komunikacioni moduli su najveći potrošači te na njih treba posebno obratiti pažnju. Logično bi bilo da nam je potreban što veći propusni opseg i veći komunikacioni radijus, ali ovo nije uvek slučaj. Potrebno je u zavisnosti od namene uređaja optimalno odrediti ove parametre i odabrati adekvatnu tehnologiju za prenos [7].

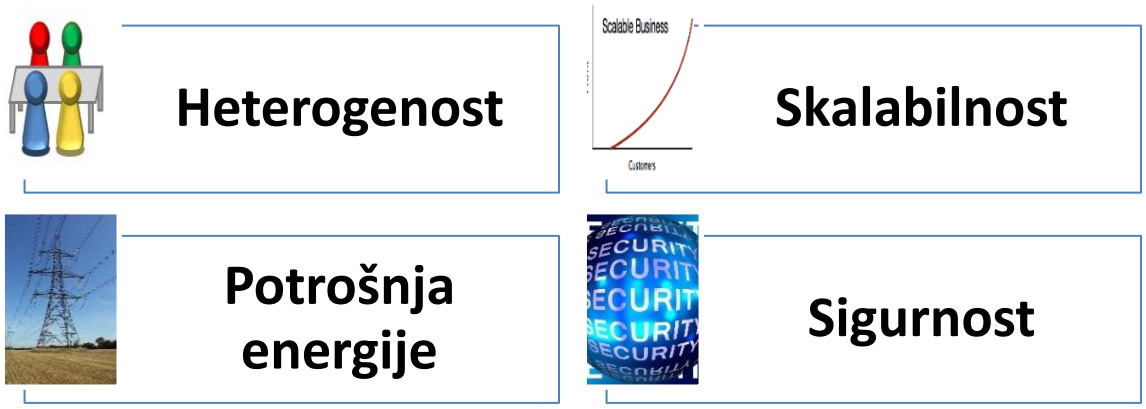
Pokretanje sistema (akcije) – aktuatori treba da budu optimalni u smislu potrošnje energije, a opet adekvatni da izvrše sve potrebne aktivnosti [7].

Napajanje sistema – pre svega je baziran na baterijskom napajanju jer je ono najčešći primer upotrebe u IoT uređajima, i ima najviše ograničenja kao što su snaga i životni vek. Jedan od glavnih problema je potreba da se obezbedi dovoljno dugo napajanje za sve operacije koje IoT uređaj treba da izvrši. Senzori koje danas koristimo mahom se napajaju preko samog mikrokontrolera tako da im je neophodno obezbediti dug životni vek [7].

Pored ovih zahteva, neophodno je imati u vidu i bezbednost prenosa podataka između IoT uređaja, koja je izuzetno važan faktor u savremenim sistemima. Budući da u ovom radu komunikaciju IoT uređaja zasnivamo na tehnologiji bežičnih senzorskih mreža, ovom temom ćemo se detaljnije baviti u nekom od sledećih poglavlja. Pored razmatranja sigurnosnih pretnji, bavićemo se i tipičnim napadima i mehanizmima odbrane.

2.3. Izazovi IoT-a

Svaki IoT uređaj ima neke specifičnosti koje ga čine jedinstvenim. Pored toga svi ovi uređaji imaju potrebu za međusobnom komunikacijom, pri čemu se srećemo sa velikim izazovima. Da bi uređaji mogli međusobno da komuniciraju, mogu da formiraju mrežu uređaja, kao i da rade samostalno. Na osnovu komunikacione tehnologije koju koriste imaju protokole prema kojima se konfigurišu, te shodno tome i komuniciraju. Neke od glavnih karakteristika i izazova koji se odnose na IoT predstavljeni su na slici 2.5, a detaljno su opisani u jednom od preglednih radova [7].



Slika 2.5. Izazovi tehnologije zasnovane na IoT

2.3.1. Heterogenost

IoT uređaji se razlikuju po nameni, dimenzijama, komunikacionim modulima i mogućnostima. IoT uređaji se međusobno povezuju u jednu mrežu. Tipovi podataka koje generišu mogu biti različiti. Da bi se ove razlike između uređaja prevazišle, potrebno je poštovati komunikacione protokole koji se koriste u komunikaciji IoT uređaja [7].

2.3.2. Skalabilnost

Kako veliki broj IoT uređaja formira veliku mrežu, suočavamo se sa opštepoznatim problemom skalabilnosti. Skalabilnost definišemo kao mogućnost uređaja da se prilagode promenama koje nastaju u njihovom okruženju. Kako se iz dana u dan povećava broj uređaja, neophodno je između ostalog obezbediti i mehanizam za njihovu jedinstvenu identifikaciju [7].

2.3.3. Potrošnja energije

Da bi se smanjila potrošnja energije, neki IoT urađaji naizmenično prelaze iz radnog u stanje mirovanja, dok se neki drugi uključuju isključivo po potrebi. Ovo je posebno značajno za senzore koji rade na baterijsko napajanje. Uređaji koji rade isključivo na baterije moraju da imaju životni vek koji zadovoljava potrebe njihove namene [7].

2.3.4. Sigurnost

Kako su IoT uređaji povezani na globalnu mrežu, svi su potencijalno ugroženi. Pri projektovanju IoT uređaja neophodno je voditi računa o bezbednosti prenosa podataka. Jedan od načina za povećanje bezbednosti podataka a da se ne naruši energetska efikasnost uređaja jeste upotreba šifarskih algoritama koji nisu računski zahtevni [7].

2.4. Područja primene

Kako IoT tehnologija skoro da nema ograničenja u primeni, to je njen spektar primene veoma širok, od različitih grana industrije, preko saobraćaja i transporta, do ekologiije i poljoprivrede pa sve do zdravstva. Kako raste broj postojećih rešenja, tako se svakodnevno otkrivaju i nova polja primene [15].

U slučajevima pametnih kuća IoT se uglavnom koristi za automatizaciju svakodnevnih rutina i kontrolu i upravljanje različitim uređajima u domaćinstvu. Još preciznija automatizacija IoT može da se postigne ukoliko senzori prikupljaju informacije iz okoline i na osnovu tih prikupljenih informacija donose odluke [15].

Analogno konceptu pametnih kuća možemo posmatrati i koncept pametnih gradova, koji je sličan pametnoj kući, samo što je mnogo veći broj mogućnosti za automatizaciju i mnogo je složeniji odnos između komponenata. U daljem tekstu ćemo objasniti svaku od primena koje smo pronašli u preglednom radu [15].

2.4.1. Pametan grad

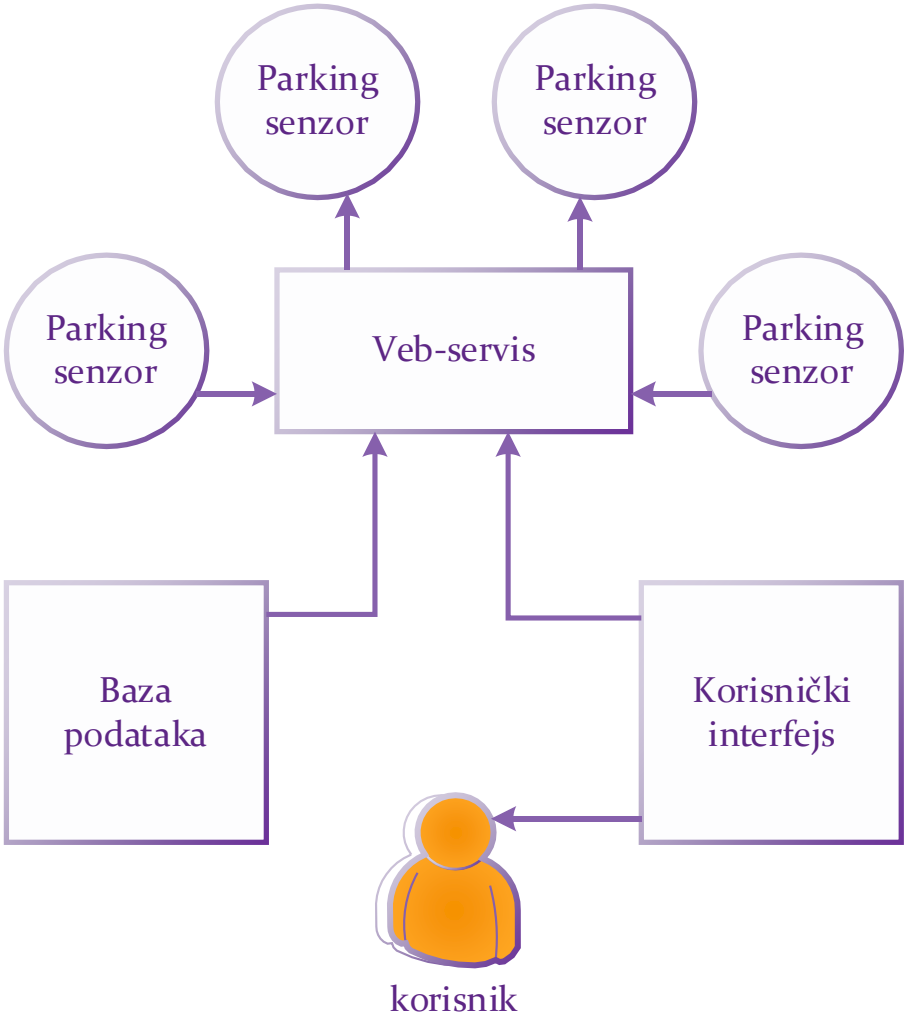
Pametan grad zasniva se na upotrebi senzora za prikupljanje podataka. Zahvaljujući obradi tih podataka pametan grad na racionalan način upravlja resursima i unapređuje životni standard [16].

Koncept pametnog grada nije zamišljen kao izolovana infrastruktura. To je kombinacija različitih sistema prikazanih na slici 2.6, kao što su sistemi za pametan parking, pametan saobraćaj, pametno osvetljenje, pametno upravljanje otpadom itd. [15].

Slika 2.6. IoT koncept pametnog grada

2.4.2. Pametan parking

Ideja o pametnom parkingu prikazanom na slici 2.7. javila se kao potreba da se smanje sve veće gužve u saobraćaju u gradovima širom sveta. Pored povećanja gužvi u saobraćaju, gubljenja vremena u vožnji zakrčenim ulicama i pronalaženju slobodnog parking mesta, bitan faktor koji je uticao na ideju o pametnom parkingu jeste i potreba da se smanji zagađenost vazduha, koja se javlja kao posledica povećane potrošnje goriva [15].



Slika 2.7. IoT koncept pametnog parkinga

Sisteme za pametno parkiranje možemo podeliti na sisteme za informacije o slobodnim parking mestima i na pametni sistem plaćanja. Svaki od ovih sistema primenjuje se na različite načine. Jedan od primera je snimanje video-zapisa, preko kojih se obradom video-materijala detektuju slobodna parking mesta, slično sistemu „oko sokolovo“ koje na osnovu video-zapisa detektuje neispravno parkirana vozila. Drugi način detekcije slobodnog parking mesta jeste pomoću senzora koji utvrđuju da li je neko parking mesto zauzeto [15].

Bez obzira koja od ovih dveju tehnologija se koristi za detekciju da li je mesto zauzeto ili nije, neophodno je te informacije proslediti korisniku. Da bi korisnik na samom mobilnom telefonu mogao da dobije informaciju, neophodno je da se informacija sa senzora prenese do nekog veb-servisa koji će preko interneta proslediti korisnicima informacije o slobodnim parking mestima u njihovoj okolini [15].

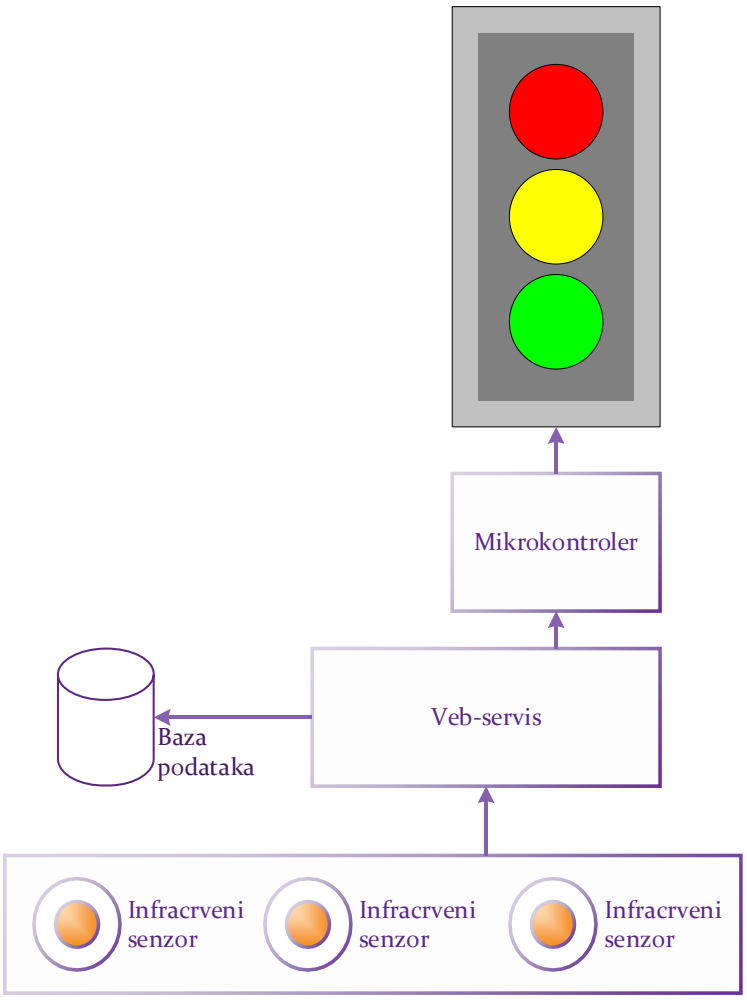
Kako je sistem postavljanja senzora za svako parking mesto relativno veliki posao, neophodno je obezbediti da senzori mogu da komuniciraju sa nekom baznom stanicom, ili nekim spoljnim servisom. Pošto senzori zahtevaju napajanje, a nije ih moguće povezati na električnu mrežu, moraju da budu energetski efikasni i da maksimalno iskoriste baterijsko napajanje. Da bi ova tehnika nesmetano funkcionisala, za komunikaciju između senzora koristi se tehnologija bežičnih senzorskih mreža [17].

2.4.3. Pametan saobraćaj

Koncept pametnog saobraćaja sličan je pametnom parkiranju, samo što je mnogo veći sistem, prikazan na slici 2.8. Problem gužvi i zagušenja u saobraćaju jeste problem sa kojim se susreće svaka urbana sredina, pogotovo u centralnim gradskim jezgrima koja nisu projektovana za veliki broj vozila. Izgradnja nove infrastrukture jedan je od načina za rasterećenje saobraćaja i smanjenje gužvi, mada to nije uvek moguće izvesti. Sistemi za upravljanje saobraćajem u obliku ustaljenih mera i propisa u saobraćaju, horizontalne i vertikalne saobraćajne signalizacije i semafori nisu dovoljni [15].

Problem funkcionisanja semafora je u tome što oni imaju unapred definisan sistem rada, bez obzira na okolnosti u saobraćaju. Rešenje za ove probleme bilo bi pametno upravljanje saobraćajem, smerovima kretanja vozila, semaforima, brojem traka i njihovim smerovima da bi se rasteretila mesta na kojima se stvaraju zagušenja i čepovi, kao i ulice kojima prolazi veliki broj ljudi [15].

Sistem pametnog saobraćaja prikazan u radu [18] zasnovan je na IoT i prati broj vozila u trakama pomoću infracrvenog senzora [18]. Dobijeni podaci šalju se servisu koji traži najoptimalnije rešenje na osnovu algoritama veštačke inteligencije. Kada algoritam utvrdi koji je optimalan vremenski interval za ponašanje semafora, on ga prosleđuje mikrokontroleru koji se nalazi u semaforu sa zadatkom da preduzme željenu akciju, kako za određen konkretan semafor, tako i za sve semafore u nizu čije ponašanje od njega zavisi [15].



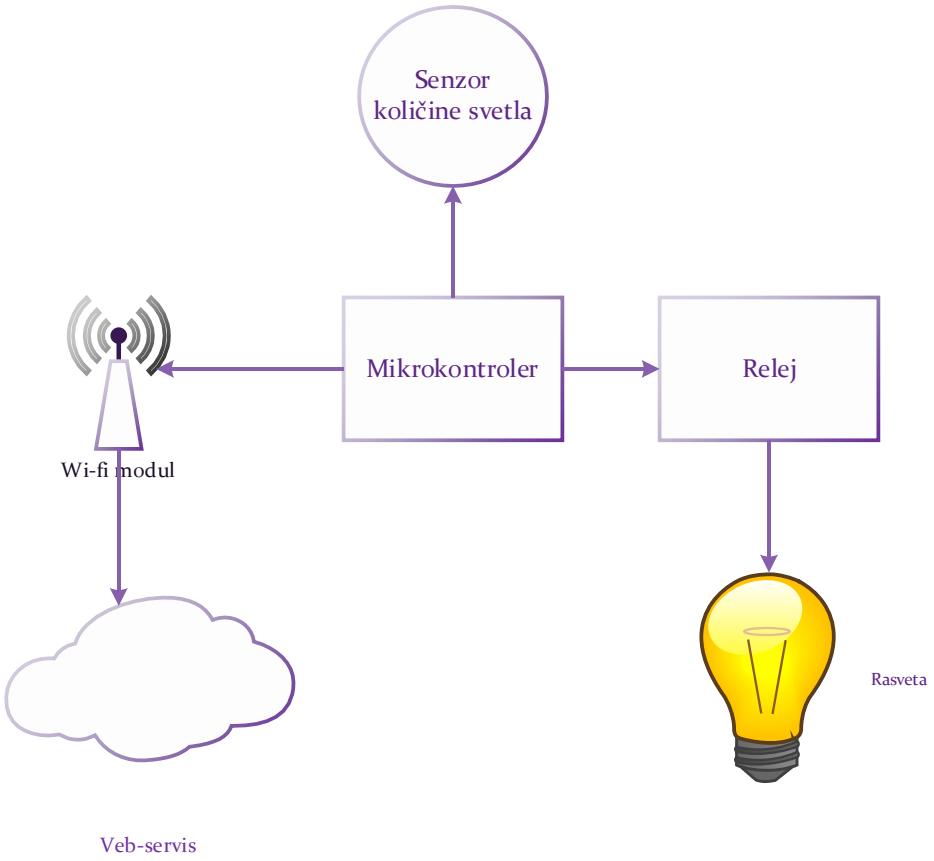
Slika 2.8. IoT koncept pametnog sistema za kontrolu saobraćaja

2.4.4. Pametna rasveta

Sistemi za upravljanje svetlom koji se danas koriste obično imaju samo inkorporiran element vremena, odnosno određeno vreme u koje se pale, i to je jedina pametna komponenta uobičajnog sistema za upravljanje javnom rasvetom [15].

Prednosti pametne rasvete prikazane na slici 2.9 su: ušteda energije, isplativost na duži period, lakša kontrola sistema, pouzdanije funkcionisanje od konvencionalnog sistema. Da bi se smanjio gubitak električne energije, uvodimo pojam pametne rasvete koja bi trebalo sama da se uključuje kada detektuje da nema dovoljno svetla na ulici, ili pak da se uključuje kada detektuje neki objekat u svojoj okolini. Jedan od takvih koncepata, prikazan je na slici 2.9, zasnovan je na tome da se svetla sama pale i gase kada se za to ustanovi potreba, ali i regulišu sopstveni intenzitet, tj. smanjuju se kada je noć ili nema objekata kojima je potrebna ulična rasveta, ili se pojačavaju kada detektuju neki pokret [19] [15].

Sistemi za pametna svetla sastoje se od senzora za količinu svetla koji nam služi za merenje osvetljenja, mikrokontrolera koji služi za učitavanje podataka i pokretanje releja, a relej nam služi kao prekidač koji pali i gasi svetlo, ili određuje njegov intenzitet. Pored ovih stvari mikrokontroler mora imati vezu sa nekim spoljnim servisom korišćenjem wi-fi [20] , gprs ili zahvaljujući tehnologiji bežičnih senzorskih mreža [15].

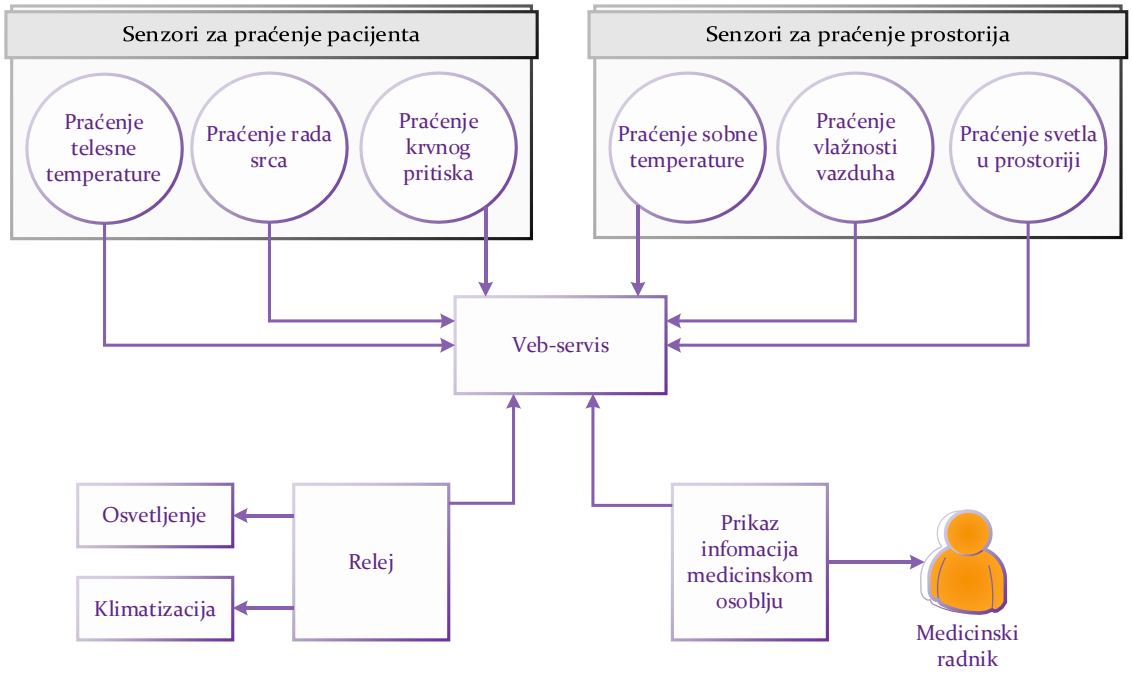


Slika 2.9. IoT koncept za upravljanje javnom rasvetom

2.4.5. Pametne bolnice

Koncept pametne bolnice prikazan na slici 2.10. ima osnovni cilj da prati zdravlje pacijenata i održavanje objekta. Da bi pacijenti bili adekvatno zbrinuti, potrebno je konstantno nadgledati njihovo stanje. Ako se ovaj postupak sprovodi neposrednom proverom zdravstvenog stanja pacijenata, to osoblju oduzima dosta vremena. IoT model se sastoji od mikrokontrolera koji sa senzora očitava vrednosti od značaja za pacijenta. Na taj način očitavaju se vrednosti kao što su temperatura, rad srca, krvni pritisak i druge [21]. Pored ovih senzora koji se nalaze na samom pacijentu i služe za praćenje njegovog zdravstvenog stanja, imamo i senzore koji mere temperaturu prostorije, vlažnost vazduha, količinu svetla ili bilo koje druge parametre od značaja. Svi prikupljeni podaci se

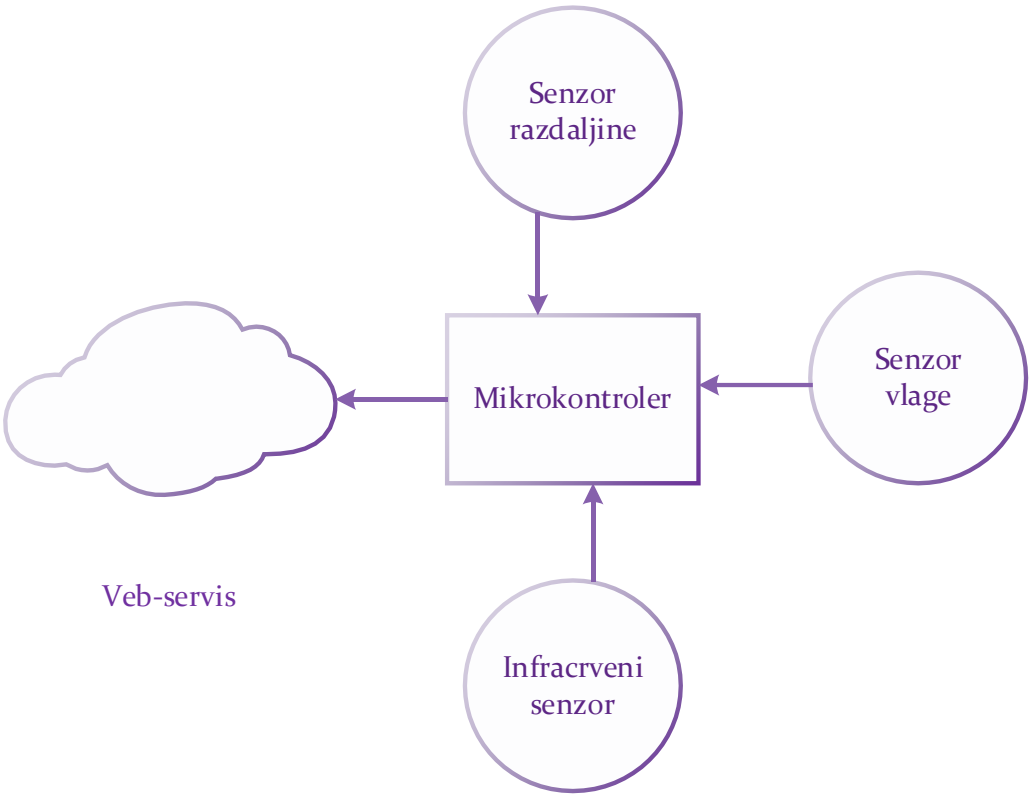
šalju na server, podatke može da nadgleda osoblje u realnom vremenu, ili da snima podatke za dalju proveru. Očitani podaci mogu i pokretati neku vrstu alarma ukoliko su prešli granične vrednosti [15].



Slika 2.10. IoT koncept pametne bolnice

2.4.6. Pametno odlaganje otpada

Sistem za odnošenje otpada prikazan na slici 2.11. zasnovan je na IoT sistemu koji rešava veliki problem sa kojim se suočavaju pre svega urbane sredine ali i ruralni krajevi. Predlog rešenja ovog problema zasnovan je na objedinjenom delovanju mikrokontrolera, senzora za detekciju razdaljine, infracrvenog senzora i senzora vlage koji su povezani na spoljni servis [22]. Po dobijenim potrebnim informacijama mikrokontroler pomoću ultrazvučnog senzora razdaljine proverava stepen popunjenosti kontejnera na osnovu razdaljine i te informacije prosleđuje centralnom servisu. Centralni servis na osnovu prikupljenih informacija zaključuje da li neki kontejner treba da bude ispražnjen [15].

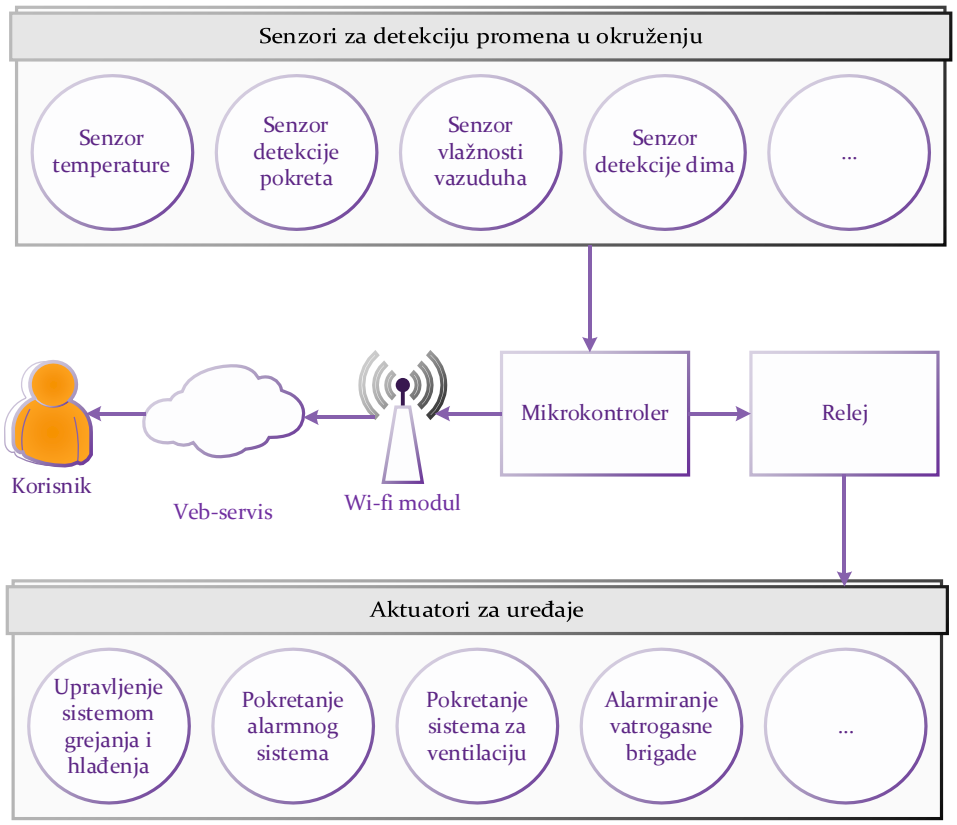


Slika 2.11. IoT koncept pametnog odlaganja otpada

2.4.7. Pametne kuće

Pametne kuće su primer gde je IoT veoma rano našao svoju primenu u svakodnevnoj upotrebi. U zavisnosti od stepena kompleksnosti sistema, možemo automatizovati kontrolu sobne temperature, osvetljenja, električnih uređaja, detekciju pokreta, detekciju upada, kontrolu potrošnje energenata itd. [15].

Arhitektura sistema, prikazana na slici 2.12, obuhvata veliki broj različitih senzora – u zavisnosti od stepena kompleksnosti projekta – mikrokontroler, komunikacioni modul, neki veb-servis i mobilnu aplikaciju za lakše upravljanje sistemom [15].



Slika 2.12. IoT koncept pametne kuće

Senzori koji se koriste u pametnim kućama imaju i mnoge druge namene. Kao što smo ranije pomenuli, jedan od bitnih aspekata bilo kog IoT rešenja jeste prikupljanje podataka. Pomoću senzora u pametnim kućama možemo prikupiti podatke kao što su količina svetla, temperatura, zasićenost vazduha vodom ili štetnim gasovima, stanje uređaja, detekcija pokreta, detekcija zvuka itd. [15].

Komunikacija između senzora i mikrokontrolera obavlja se korišćenjem neke od dobro poznatih tehnologija kao što su *Lora*, *ZigBee*, *Bluetooth*, *Wi-Fi* ili *GSM* signala [15].

Mikrokontroler obrađuje podatke dobijene od senzora koji su do njega stigli nekim od prethodno navedenih komunikacionih tehnologija i preko interneta ih prenosi do nekog veb-servisa koji pokreće dalje aktivnosti na osnovu dobijenih podataka [15].

Bitan deo sistema za kućnu upotrebu kao što je upravljanje pametnom kućom jeste korisnička aplikacija. Bilo da je u pitanju veb-aplikacija ili da je mobilna aplikacija, pozadina je uvek ista, veb-servis koji obrađuje podatke [15].

4. Bežične senzorske mreže

Poslednjih godina primetan je nagli razvoj tehnologija korišćenja bežičnih senzorskih mreža – BSM (engl. *Wireless Sensor Networks*), i to u sasvim različitim oblastima, kao što su vojna industrija, medicina, sport, saobraćaj, razne privredne grane, poljoprivreda, kao i nadgledanje raznih parametara u okruženju [23]. Komunikacija IoT uređaja zasnovana je na primeni tehnologije bežičnih senzorskih mreža.

Bežične senzorske mreže koriste se za posmatranje različitih tipova događaja, a to se postiže korišćenjem uređaja koji se nazivaju senzorski čvorovi. Bežične senzorske mreže sastoje se od senzorskih čvorova, procesora i radio-frekventnih (RF) modula i imaju baterijsko napajanje. Senzorski čvorovi imaju zadatak da uspostave bežičnu komunikaciju i, u zavisnosti od arhitekture, prikupljene podatke prosleđuju čvoru za komunikaciju ili baznoj stanici. Način komunikacije senzorskih čvorova zavisi pre svega od samog sklopa datog čvora, kao i od topologije na kojoj je zasnovana BSM [1].

Čvorovi koji se koriste mogu biti veoma kompleksni, u slučaju kada prate lokaciju ili analiziraju slike, ali mogu da budu i sasvim jednostavni, kada prate promene temperature, pritiska, vlažnosti, pH vrednosti, nivoa vode i nekih drugih pokazatelja. Svi ovi senzorski čvorovi omogućavaju nam da precizno pratimo razne promene parametara koji su od značaja za donošenje odluka. Primena tehnologije BSM za prikupljanje, čuvanje i obradu podataka pruža nam mogućnosti da konstantno i u realnom vremenu donosimo odluke na osnovu prikupljenih podataka, bez obzira koja je oblast primene u pitanju.

Jedan od ilustrativnih primera primene BSM jeste unapređenje poljoprivrede i način proizvodnje poljoprivrednih kultura, a samim tim i njihovog prinosa [24]. Osnovni razlozi za upotrebu BSM u preciznoj poljoprivredi jesu potreba da se povećaju prinosi a da se, s druge strane, smanji uticaj subjektivnog ljudskog faktora [25]. Naveli smo primer precizne poljoprivrede zato što u ovom slučaju pratimo ponašanje BSM na otvorenom polju, gde senzori nisu lako dostupni za servisiranje, a opet su dovoljno dostupni za bilo koji tip napada.

Prilikom primene bežičnih senzorskih mreža srećemo se sa sledećim problemima koje je potrebno rešiti; to su [26]: izrada optimalnih nacrta raspoređivanja čvorova, određivanje perioda merenja, izbor protokola za rutiranje, energetska efikasnost, način

prenosa podataka, skalabilnost, stepen tolerancije na greške, kao i bezbednost prenosa i tačnost podataka [1].

Pored problema sa energetskom efikasnošću, sve uočljivi je i problem bezbednosti prenosa podataka u BSM. Da bi se jedna bežična senzorska mreža smatrala bezbednom, neophodno je da se obezbede: dostupnost, poverljivost, integritet i autentifikacija podataka koji se prenose kroz mrežu.

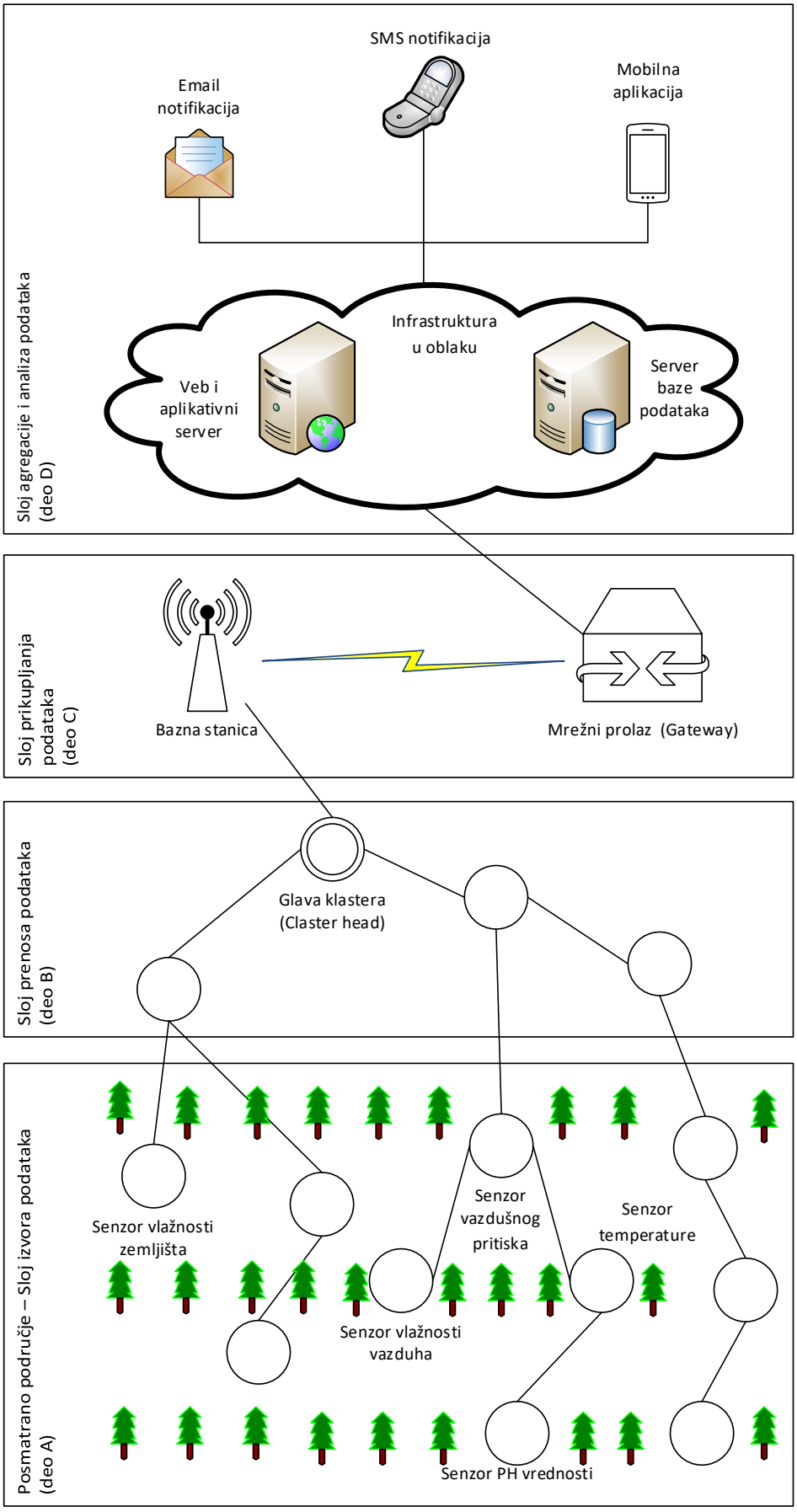
4.1. Tipična arhitektura bežičnih senzorskih mreža

U ovom poglavlju dajemo pregled tipične arhitekture senzorske mreže za potrebe prikupljanja podataka na otvorenom prostoru, potom razmatramo probleme bezbednosti prenosa podataka i predlažemo arhitekturu koja obezbeđuje najoptimalniju sigurnost, vodeći računa o energetskoj efikasnosti svakog čvora ponaosob.

Senzorski čvorovi u bežičnim senzorskim mrežama koriste se za prikupljanje informacija o fizičkim karakteristikama posmatranog područja, dok se aktuatori koriste za odgovor na povratne informacije kako bi kontrolisali ili izvršavali akcije u određenim uslovima. Aktuatori su uređaji koji se nalaze na posmatranom terenu koji autonomno (bez prisustva čoveka) preduzmaju neophodne mere za prevenciju ili smanjenje već nastalog problema.

Arhitektura sistema prikazana na slici 4.1. podeljena je u četiri sloja:

- Sloj izvora podataka;
- Sloj prenosa podataka;
- Sloj prikupljanja podataka;
- Sloj skladištenja i analize podataka.



Slika 4.1. Arhitektura sistema bežične senzorske mreže

4.1.1. Slojevi arhitekture

4.1.1.1. Sloj izvora podataka

Na ovom sloju senzorski čvorovi različitih namena prikupljaju podatke iz okruženja. Na slici 4.1, deo A, prikazani su neki od senzora koji mogu da prikupljaju informacije o temperaturi zemljišta, temperaturi vazduha, vlažnosti vazduha, vazdušnom pritisku, vlažnosti zemljišta, PH vrednosti zemljišta. Tu mogu da se nađu i drugi senzori koji su od značaja za konkretnu primenu. Kao što je prikazano na slici, između ovih senzorskih čvorova postoji direktna komunikacija.

4.1.1.2. Sloj prenosa podataka

Za komunikaciju BSM predviđen je protokol *ZigBee*. *ZigBee* je tehnologija bežičnog prenosa podataka koja ima nisku cenu implementacije i veliku energetsku efikasnost. *ZigBee* je baziran na IEEE 802.15.4 koji radi na frekvencijama od 2,4GHz, 900 MHz i 868 MHz [27]. Na slici 4.1, deo B, predstavili smo čvorove za agregaciju očitanih podataka. Neki od njih imaju ulogu *claster head* čvora, koji je zadužen za određeni broj čvorova u sistemu. Ovi čvorovi zajedno sa čvorovima koji su prikazani na slici 4.1, deo A, predstavljaju hibridnu topologiju BSM na kojoj je zasnovana naša arhitektura.

4.1.1.3. Sloj prikupljanja podataka

Na slici 4.1, deo C, prikazan je sloj prikupljanja podataka koji ima zadatak da sve podatke koji se prikupe kroz slojeve izvora i prenosa podataka prenese kroz baznu stanicu i pristupnu tačku (engl. *gateway*) do sloja za skladištenje i analizu podataka.

4.1.1.4. Sloj skladištenja i analize podataka

Funkcija ovog sloja je da već prikupljene podatke uskladišti i analizira na *cloud* arhitekturi kao što je to prikazano na slici 4.1, deo D. Analizu podataka može da obavlja aplikacija ili servis koji opslužuje celokupnu arhitekturu sistema. Izveštaji koji se generišu mogu biti ili pozivi nekih aktuatora u sklopu sistema ili u obliku *push* notifikacije, sms poruke ili imejl (*email)* obaveštenja, notifikcije za mobilne aplikacije.

4.2. Problemi bezbednosti u BSM

Sa povećanjem decentralizovanog distributivnog sistema prisustvo zlonamernog ponašanja više nije izuzetak, već postaje normalna pojava. Da bi bežične senzorske mreže mogle da se koriste u različitim aplikacijama, neophodni su jednostavni protokoli za

upravljanje topologijom, za bezbednost i komunikacije. Bezbednost se nameće kao jedno od najvažnijih pitanja u korišćenju bežične senzorske mreže [2]. BSM ima nekoliko karakteristika koje je čine podložnom različitim napadima [28]:

- Senzorski čvorovi u bežičnim senzorskim mrežama imaju ograničenu memoriju, energiju, sposobnost računanja, propusni opseg i opseg komunikacije.
- Nasumično *(Ad hoc)* raspoređivanje čvorova u senzorskoj mreži olakšava napadačima da pokrenu različite vrste napada koji se kreću od aktivnog ometanja do pasivnog prisluškivanja.
- BSM topologija je dinamična i nedostaje joj fiksna infrastruktura, zbog čega je neprekidan nadzor mreže otežan.
- Snažni sigurnosni protokoli mogu degradirati performanse aplikacija pošto troše više resursa na senzorskim čvorovima. Stoga se mora uspostaviti kompromis između performansi i bezbednosti. Međutim, napadači mogu lako da probiju slabe bezbednosne protokole.
- Svako može da učestvuje ili prati kanale komunikacije kroz bežičnu senzorsku mrežu sa radio-konfiguracijom na istoj frekvenciji. Tako napadači mogu jednostavno da probiju u bežične senzorske mreže.

4.3. Preduslovi za nesmetano funkcionisanje bežičnih senzorskih mreža

Bežične senzorske mreže (BSM) zahtevaju razmenu informacija među legitimnim korisnicima. Da bismo zaštitili bežične prenose od različitih vrsta napada, postoje dva osnovna zahteva koja je potrebno ispuniti u bežičnim senzorskim mrežama: bezbednost i potreba za preživljavanje (*survivality requirements*) bežičnih senzorskih mreža.

4.3.1. Zahtevi za bezbednost u bežičnim senzorskim mrežama

Postoji nekoliko zahteva za bezbednost koje je potrebno ispuniti da bi se bežični prenos zaštitio od napada kao što su DoS napad, napad ugrožavanja čvorova, napad prisluškivanja i tako dalje. Različite vrste zahteva za bezbednost bežičnih senzorskih mreža istražene su u nastavku rada [28].

4.3.1.1. Poverljivost

Poverljivost osigurava zaštitu osetljivih informacija tako da neovlašćeni korisnici ne dobijaju pristup osetljivim informacijama. Poverljivost štiti otkrivanje informacija u

senzorskom okruženju kada se paketi prenose između senzorskih čvorova ili između bazne stanice i čvorova. Kod poverljivosti, najveću opasnost predstavlja postojanje kompromitovanih čvorova, jer napadač može da eksploatiše ove čvorove da bi ukrao važne podatke kao što su kriptografski ključevi. Ovi ključevi mogu se koristiti za dešifrovanje poruka i dobijanje osetljivih informacija. Deo podataka prenetih paketa je šifrovan, a ponekad je i zaglavlje paketa takođe šifrovano, što u osnovi štiti identitet čvora [28].

4.3.1.2. Autentifikacija

Ova tehnika koristi se za verifikaciju identiteta korisnika i uglavnom razlikuje zlonamerne od legitimnih korisnika. U slučaju bežičnih senzorskih mreža, svaka bazna stanica i senzorski čvor moraju biti sposobni da ustanove da li im paket šalje napadački ili legitimni čvor. Na taj način moguće je izbeći situaciju da napadač prevari legitimni čvor i primora ga da prihvati lažne pakete podataka. Lažni podaci mogu u znatnoj meri da utiču na funkcionisanje senzorske mreže [28].

4.3.1.3. Integritet

Integritet sprečava izmenu informacija tokom procesa prenosa podataka u senzorskoj mreži. Upotreba netačnih ili pogrešnih podataka može dovesti do nesagledivih posledica, pa je nedostatak integriteta ozbiljan problem. Donošenje odluke u sistemima zasnovanim na BSM u potpunosti se oslanjaju na integritet informacija koje se prenose kroz mrežu. Zaštita podataka od izmene ili presretanja od izuzetne je važnosti za ovu primenu [28].

4.3.1.4. Upravljanje bezbednošću

Od posebne važnosti za primenu bežičnih senzorskih mreža jeste kontrola bazne stanice. Bazna stanica je jedna od najčešćih meta napada pa je neophodno šifrovanje komunikacije, kao i održavanje informacija o rutiranju. U tehnici koja podrazumeva postojanje klastera, svaki od klastera sastoji se od velikog broja čvorova, zbog čega je za bezbednu razmenu podataka potrebno sigurno upravljanje [28].

4.3.2. Zahtevi za preživljavanje bežične senzorske mreže

Postoje tri osnovna zahteva za preživljavanje bežičnih senzorskih mreža: pouzdanost, dostupnost i energetska efikasnost. Ovi zahtevi za preživljavanje opisani su u nastavku [28].

4.3.2.1. Pouzdanost

Pouzdanost je jedan od važnih aspekata u senzorskim mrežama, pogotovo ako posmatramo sisteme koji imaju malu toleranciju na grešku. Neki senzorski čvorovi mogu prouzrokovati neželjene probleme ili mogu uticati na pouzdanost celokupne mreže. Pouzdanost se odnosi na sposobnost mreže da nastavi sa svojom funkcijom čak i u slučaju da se mali broj čvorova suoči sa nekim uspehom u komunikaciji [28].

4.3.2.2. Dostupnost

Pod pojmom dostupnosti podrazumevamo da su nam informacije i usluge na raspolaganju u bilo koje vreme ako je to potrebno. Napadi odbijanja usluge ili kompromitovanje čvora mogu dovesti do toga da nekoliko servisa postane nedostupno, što može dovesti do pogubnih posledica za neke aplikacije koje rade u realnom vremenu. Protokoli bežičnih senzorskih mreža koji se koriste moraju biti robustni, tako da se mogu sprečiti bilo kakvi prekidi u radu [28].

4.3.2.3. Energetska efikasnost

Bežične senzorske mreže koje se sastoje od senzorskih čvorova koji imaju baterijsko napajanje imaju ograničenu energiju. Očuvanje energije je važan aspekt za senzorske mreže. Duži vek trajanja baterije povećava i dostupnost i pouzdanost kada su senzorske mreže u pitanju. Korišćeni protokoli rutiranja moraju biti energetski efikasni da bi se obezbedilo dovoljno energije za obradu podataka, računanje i komunikacione komponente [28].

4.4. Istaknuti napadi i njihove protivmere u bežičnim senzorskim mrežama

U tabeli 1. koja sledi dat je opis karakterističnih napada koje smo razvrstali po slojevima arhitekture BSM. Pored ove podele, data je i podela prema vrsti napada. Podela napada po slojevima sistema omogućuje nam da u poglavlju koje sledi detaljno obrazložimo mehanizme odbrane od napada na IoT zasnovanih na BSM [29], [30].

Tabela 1. Karakteristični napadi na BSM

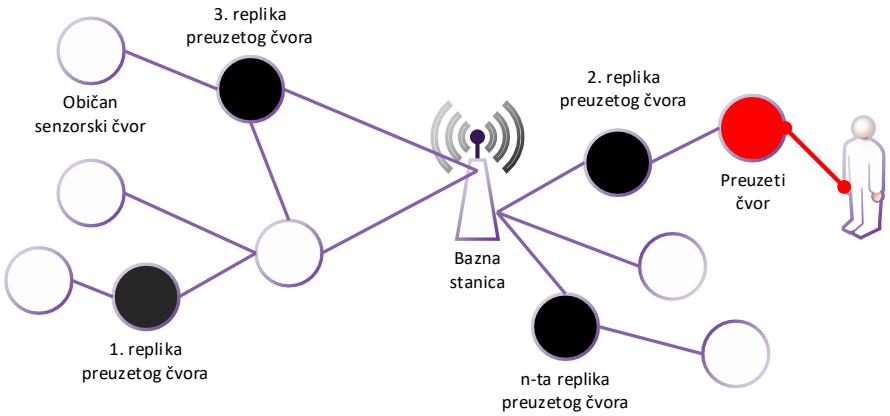
Vrsta napada	**Napad**	**Slojevi**		
		Izvor podataka	**Prenos podataka**	**Prikupljanje podataka**
Napadi koji se odnose na poverljivost podataka i integritet	Napad hvatanja čvora	Opasnost od prikupljanja kriptografskih ključeva i lažnog predstavljanja u BSM		
Napadi koji se tiču potrošnje energije	Napad odbijanja sna [31]	Opasnost od prevremenog prekida baterijskog napajanja		
Potrošnja propusnog opsega i napadi koji se odnose na dostupnost usluga	Napad preplavljivanja pozdravnim porukama [32]		Opasnost od prekida komunikacije u celoj mreži	
	Napad ometanja [32]		Opasnost od zauzimanja kanala za prenos podataka	
	Napad ponavljanja	Opasnost od preopterećenja senzorskog čvora		
	Napad selektivnog prosleđivanja		Opasnost od prosleđivanja nebitnih podataka	
Napadi u vezi s rutiranjem [33]	Napad crvotočine [34]		Opasnost po protokol rutiranja koji koristi geografsku lokaciju	
	Napad otvorene rupe [35]			Opasnost od preuzimanja uloge pristupne tačke (*gateway*) u mreži
Napadi koji se tiču identiteta	Napadi imitiranja [36]		Opasnost od imitiranja čvorova u mreži	
	Napad taloga [37]		Opasnost od lažnog identiteta paketa ili modifikacije paketa	
Napadi koji se odnose na privatnost [38]	Napad analize protoka [39] [40]		Opasnost od prikupljana informacija o mreži, čvorovima i komunikacionim protokolima	

Najistaknutiji napadi u bežičnim senzorskim mrežama su napad taloga, napad odbijanja sna, napad crvotočine, napad ometanja, napad selektivnog prosleđivanja i napada otvorene rupe, i opisani su u tekstu koji sledi.

4.4.1. Napad hvatanja čvora

Bežične senzorske mreže podložne su napadu hvatanja čvora *(*engl*. Node capture attack)* ukoliko čvorovi nisu raspoređeni tako da budu pod nadzorom koji je prikazan na slici 4.2. Kada protivnik uhvati senzorske čvorove, on može kroz taj kompromitovani čvor da pokrene različite vrste napada. Napadač uzima informacije o tajnim ključevima iz ugroženog čvora i širi (plasira) u BSM veliki broj replika, tj. lažnih čvorova koje imaju isti ID i tajni ključ kao originalni čvor.

Napad hvatanja čvora može da nanese veliku štetu energetskoj efikasnosti susednih čvorova, ali i samoj distribuciji netačnih informacija sa tog čvora, što može da ima za posledicu donošenje nekih pogrešnih odluka [41].



Slika 4.2. Napad hvatanja čvora

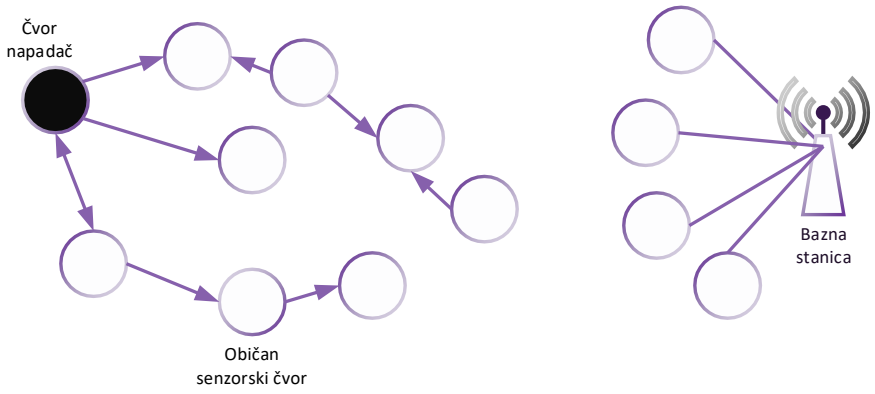
4.4.2. Napad odbijanja sna

Napad odbijanja sna (engl*. Denial of sleep attack*) jeste specifična vrsta napada čija je glavna meta baterijsko napajanje i iscrpljivanje njegovih resursa, a oni nisu neograničeni. Ovaj napad, prikazan na slici 4.3, pričinjava poteškoće zato što zamena senzora često nije nimalo jednostavna, već komplikovana ili skupa, a to isto odnosi se i na zamenu ili dopunjavanje baterijskog napajanja. Ukoliko BSM podlegne ovom napadu i ne uspemo da ga zaustavimo, to automatski znači da je time smanjen životni vek mreže.

Čvorovi su ugroženi uprkos tome što se autentičnost potvrđuje upotrebom *hash* algoritama i simetričnih ključeva. Zlonamerni čvor koji pristupa simetričnim ključevima može pristupiti i informacijama koje pripadaju baznoj stanici. Ukoliko dođe do kompromitovanja podataka koji se tiču bazne stanice, onda je i cela BSM kompromitovana. Ovaj problem moguće je rešiti upotrebom protokola pod nazivom izazvani odgovor (*challenge response*).

Metoda autentifikacije zasnovane na izazovima senzorskih čvorova zahteva da se pošiljalac predstavi i da otkrije originalni identitet, ali ako je čvor koji zahteva verifikaciju ranjiv, tj. pod napadom čovek-u-sredini (engl*. man in the middle*) ili kompromitovan nekim drugim napadom prisluškivanja, onda napadač dobija sve važne informacije. Ukoliko dođe do ovoga, napadač se može predstaviti kao čvor od autoriteta i samim tim preuzeti kontrolu nad mrežom i tako počinje napad odbijanja sna [42].

U arhitekturi bežične senzorske mreže najkritičnija tačka je bazna stanica, tako da svi podaci koji joj pripadaju moraju da budu maksimalno zaštićeni, tj. ne smeju da budu poznati nijednom drugom čvoru u mreži jer je to jedini način da budemo sigurni da bazna stanica nije kompromitovana.

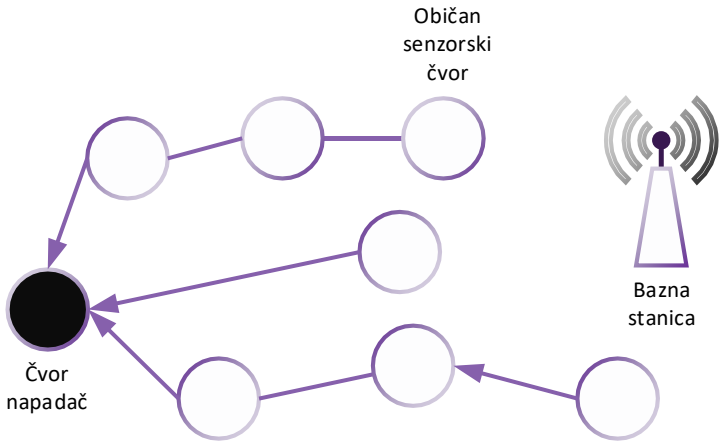


Slika 4.3. Napad odbijanja sna

4.4.3. Napad preplavljivanja pozdravnim porukama

Napad preplavljivanja pozdravnim porukama *(*engl*. Hello flood attack)* jeste opasnost od poplava porukama koje se koriste za otkrivanje suseda, a prikazan je na slici 4.4. Napadač velikom snagom prenosa šalje pozdravne pakete ili odgovara na njih tako da ga veliki broj čvorova u mreži vidi kao susedni čvor. Napadač emituje pakete tako velikom snagom da ga veliki broj čvorova u mreži bira kao roditeljski čvor. Kašnjenje se

povećava jer se sve poruke prenose preko čvora napadača. Na ovaj način deo mreže koji je pod napadom stiče utisak da je napadač bazna stanica, te napadač može da preuzme potpunu kontolu nad čvorovima i u potpunosti odseče deo mreže od ostatka BSM [43].



Slika 4.4. Napad preplavljivanja pozdravnim porukama

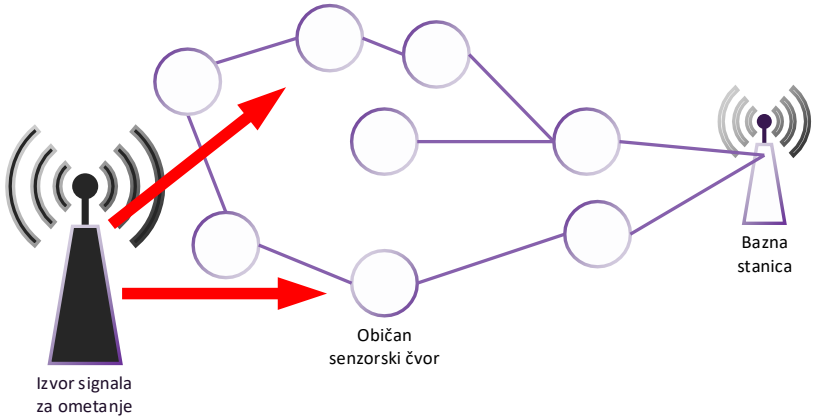
4.4.4. Napad ometanja

Napad ometanja *(*engl. *Jamming attack)* može se posmatrati kao poseban tip DoS napada i prikazan je na slici 4.5. Kroz preplavljivanje ovaj vid DoS napada sprečava normalno korišćenje komunikacija u mreži. Uređaj koji izvodi ometanje emituje signale radio-frekvencija koji su za čvorove senzora beskorisni ili neželjeni podaci. Ovaj signal može da podseća na mrežni protok ili može biti beli šum. Čin usmeravanja elektromagnetne energije namerno prema komunikacionom sistemu naziva se „ometanje“, kako bi se prekinuo prenos signala [44]. Napadi bežične senzorske mreže koji se ,,igraju“ radio-frekvencijama čvorova nazivaju se ometanje [2].

Izdvajamo tri vrste napada ometanja:

- **Ometanje tačke** *(*engl*. Spot jamming)* jeste najjednostavniji vid ovog napada pri kojem napadač cilja na jednu frekvenciju koju žrtva koristi trošeći mnogo snage kako bi nadglasala originalni signal.
- **Ometanje menjanjem** *(*engl*. Sweep jamming)* jeste napad zasnovan na brzom prebacivanju na različite frekvencije signala pune snage i na taj način nakratko ometa više frekvencija, ali ne u isto vreme sve njih.

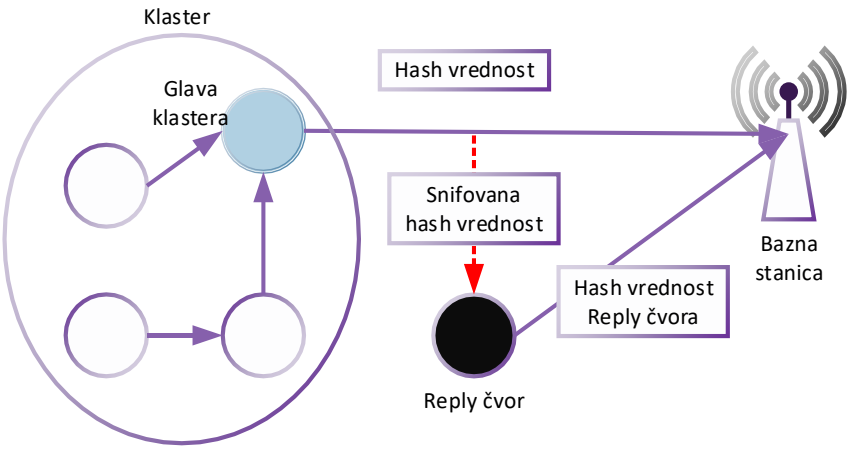
- **Baražno ometanje** *(*engl. *Barrage jamming)* jeste tip napad gde napadač u isto vreme ometa više frekvencija. Kako raste opseg ometanih frekvencija, tako se smanjuje moć ometanja na prethodnu frekvenciju.



Slika 4.5. Napad ometanja

4.4.5. Napad ponavljanja

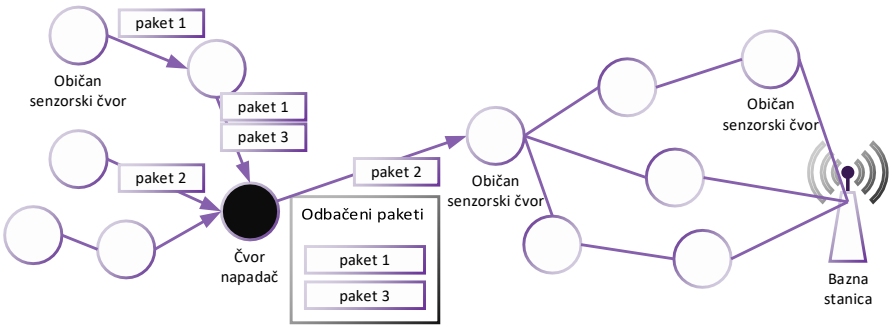
Napad ponavljanja (engl. *Replay attack*) – napad ponavljanja jedan je od podtipova napada odbijanja spavanja i prikazan je na slici 4.6. BSM je ranjiva i podložna ovom tipu napada koji se zasniva na prisluškivanju saobraćaja u mreži i njegovom ponovnom slanju kroz mrežu. Ako se na neki način ne branimo od ove vrste napada, saobraćaj koji napadač generiše smatraće se regularnim saobraćajem za ovu mrežu pa će čvorovi pokušati da ga prenesu do odredišta. Na ovaj način generiše se velika količina nekorisnog saobraćaja u mreži i tako se iscrpljuje napajanje svakog čvora ponaosob. Napad je zasnovan na principu da protivnik presreće podatke i da ih ponovo šalje [45].



Slika 4.6. Napad ponavljanja

4.4.6. Napad selektivnog prosleđivanja

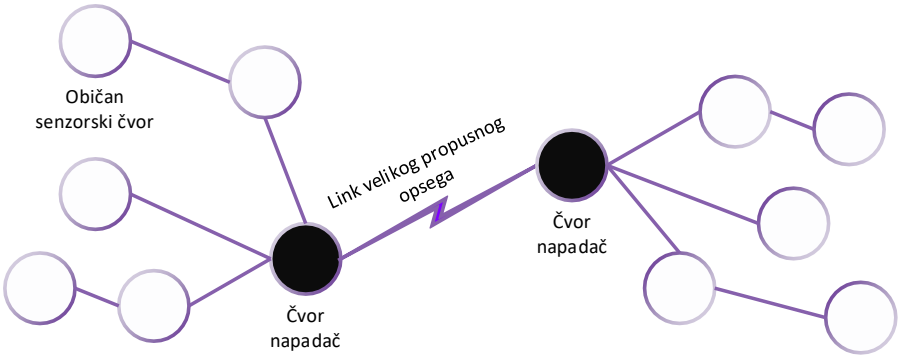
Napad selektivnog prosleđivanja *(*engl*. Selective forwarding attack),* prikazan na slici 4.7, jeste napad na mrežnom nivou u kojem zlonamerni čvor odbija neke legitimne pakete i odbacuje ih [33]. Čvor koji se ponaša kao crna rupa jednostavniji je oblik ovog napada u kojem takav čvor odbija sve pakete koji prolaze kroz njega. Međutim, u takvoj vrsti napada postoji mogućnost da čvorovi otkriju napad i na taj način isključe neprijatelja iz rutiranja. Tako je pokrenuta složena forma sličnog napada gde čvorovi odbacuju pakete, delimično otežavajući njihovo otkrivanje. Najefikasniji napad selektivnog prosleđivanja dešava se kada je protivnik eksplicitno uključen u putanju toka podataka. Selektivno prosleđivanje može se implementirati na dva različita načina u odnosu na pakete koji se odbijaju. Prvo, odbacivanjem paketa određenog tipa i, drugo, odbacivanjem paketa određenog porekla ili namene za određene čvorove [28].



Slika 4.7. Napad selektivnog prosleđivanja

4.4.7. Napad crvotočine

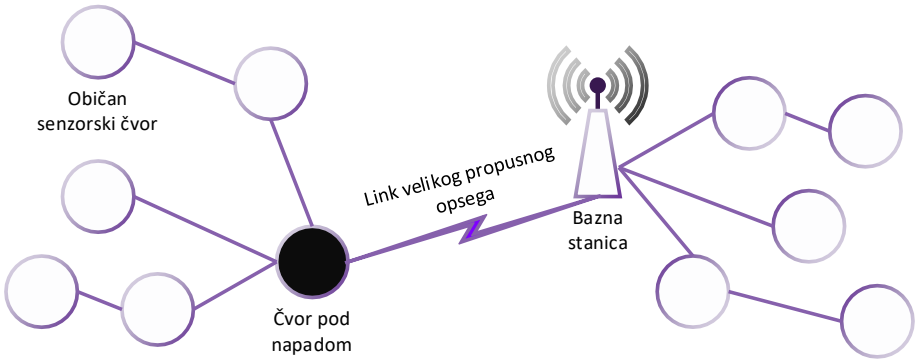
Napad crvotočine *(*engl*. Wormhole attack)* je napad gde neprijatelji putem tunela šalju paket kroz snop ili van snopa kanala između dve udaljene lokacije prikazanih na slici 4.8. Takvi crvotočni tuneli udaljenim čvorovima stvaraju iluziju da su oni mnogo bliži jedan drugome nego što zapravo jesu. Protivnik može sakupljati mrežni protok i manipulisati njime jer crvotočina može proći i privući ogromnu količinu protoka. Napadač ne poseduje validan mrežni identitet, on je autsajder koji može da prosledi komunikacijski tok duž crvotočine a da nema direktan uvid u sadržaj paketa. Koristeći takve veze sa crvotočinom, protivnik može da pokrene obrnuti inženjering protokola, napad čovek-u-sredini *(*engl*. man in the middle)*, prekidanje šifara itd. [28].



Slika 4.8. Napad crvotočine

4.4.8. Napad otvorene rupe

U napadu otvorene rupe (engl. *Sinkhole attack*), prikazanom na slici 4.9, kompromitovani čvor izgleda posebno privlačno za okolne čvorove u skladu sa algoritmom rutiranja u napadnutoj mreži. Kompromitovani čvor privlači okolne čvorove lažnim informacijama o rutiranju, a zatim menja podatke koji prolaze. Svojim selektivnim prosleđivanjem ili neprosleđivanjem određenih poruka kompromitovani čvor sprečava pristupnu tačku (*gateway*) da dobije potpunu i tačnu informaciju, što stvara poteškoće u procesu analize i obrade podataka [28].

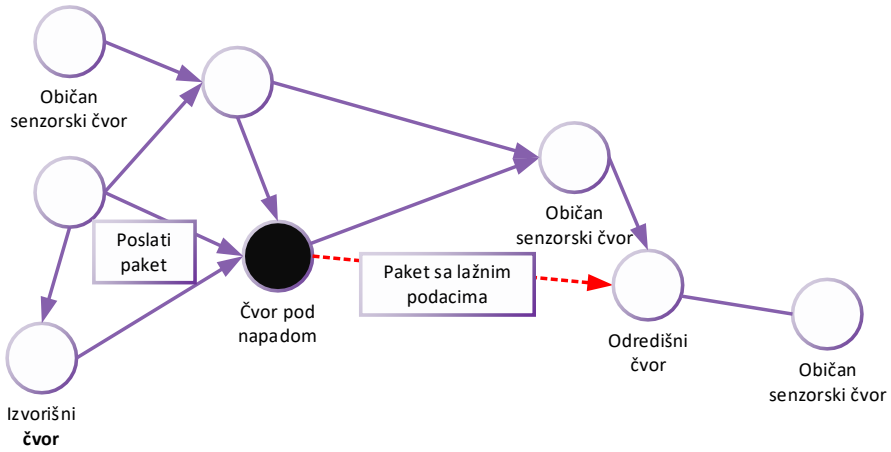


Slika 4.9. Napad otvorene rupe

4.4.9. Napadi imitiranja

Napad imitiranja *(*engl. *Impersonate attack*) zasnovan je na ugrožavanju autentičnosti izvorišnog čvora. Napad prikazan na slici 4.10. nije moguće izvesti ukoliko dva čvora koja komuniciraju nisu jedan drugom u dometu, a da bi mogli da komuniciraju međusobno, neophodno je da se komunikacija obavlja posredno, preko drugih čvorova koji se nalaze između njih. Ovaj način komunikacije čest je primer u arhitekturi predloženoj za primenu BSM u preciznoj poljoprivredi [46].

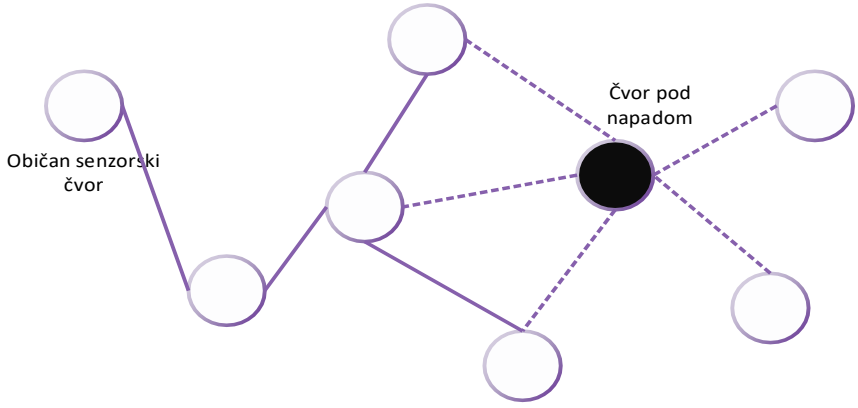
Napadi oponašanja, koji su od ključne važnosti za primenu na otvorenom prostoru, sprovode se ili tako što zlonamerni čvor preuzme ID izvorišnog čvora iz paketa koje prosleđuje iz relejnih paketa podataka ili pak tako što zlonamerni čvor šalje lažne podatke odredišnom čvoru, a u tim podacima je i prethodno ukradeni ID izvorišnog čvora [46].



Slika 4.10. Napadi imitiranja

4.4.10. Napad taloga

Za napad taloga *(*engl. *Sibil attack),* prikazan na slici 4.11*,* karakteristično je da iscrpljuje resurse i da predstavlja ozbiljnu pretnju protokolima rutiranja. U ovom napadu napadač preuzima identitet čvorova u sistemu i lažno se predstavlja drugim legitimnim čvorovima u sistemu. Takvi čvorovi nazivaju se taložnim *(*engl. *sibil)* čvorovima. Taložni čvorovi šalju veliki broj zahteva za pridruživanje pristupnoj tački, koristeći nasumične MAC vrednosti. Jednom kada taložni čvor zauzme kanale za pristup ili asocijativne slotove, onda je legitimnim klijentima pristup onemogućen [28].

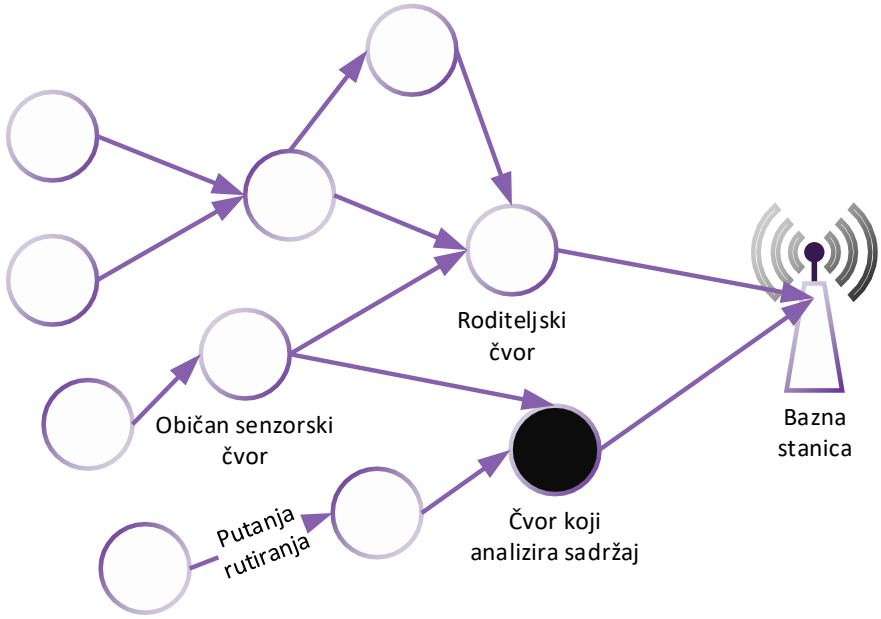


Slika 4.11. Napad taloga

4.4.11. Napad analize protoka

Napad analize protoka *(*engl*. Traffic analysis attack),* prikazan na slici 4.12, zasniva se na tome da bazna stanica služi za prikupljanje podataka i da poseduje sve osetljive informacije bitne za napad analize saobraćaja. Da bismo zaštitili baznu stanicu, neophodno je da ona ne bude jasno vizuelno uočljiva jer se tako dodatno olakšava njeno lociranje, a samim tim i povećava mogućnost za napad analize saobraćaja. Bazna stanica imitira rad čvorova da se ne bi lako otkrila njena lokacija u bežičnim senzorskim mrežama (26).

Napadač koristi antene u blizini područja za prisluškivanje radio-signala da bi na taj način došao do informacija o posmatranoj BSM. Presretnuti saobraćaj šifrovan je nekim od algoritama za čije je razbijanje neophodno mnogo vremena, pa to nije cilj ovog napada. Glavni cilj ovog napada je da se primenom nekih tehnika za određivanje lokacije utvrdi položaj izvora podataka [40]. Na osnovu ovoga pravi se struktura cele posmatrane mreže i na taj način je moguće odrediti poziciju čvora na nivou ćelije.



Slika 4.12. Napad analize protoka

4.5. Mehanizmi odbrane od napada na bežične senzorske mreže

Mehanizmi odbrane od navedenih napada prema arhitekturi sistema bežičnih senzorskih mreža prikazani su u tabeli 2.

Tabela 2. Mehanizmi zaštite od napada na bežične senzorske mreže

Vrsta napada	Napad	Slojevi		
		Izvor podataka	Prenos podataka	Prikupljanje podataka
Napadi koji se tiču poverljivosti podataka i integriteta	Napad hvatanja čvora	Detekcija napada na osnovu broja senzorskih čvorova preuzetih od strane napadača.		
Napadi koji se odnose na potrošnju energije	Napad odbijanja sna [31]	OHC (*One-way Hash Chains*) za sprečavanje širenja ponovljenih ili lažnih paketa.		
Potrošnja propusnog opsega i napadi u vezi sa dostupnošću usluga	Napad preplavljivanja pozdravnim porukama [32]		Mehanizam zasnovan na podeli BSM na klastere sa pragom broja čvorova koji pripadaju svakom od klastera.	
	Napad ometanja [32]		*Ultra Wide Band* tehnike modulacije	
	Napad ponavljanja	Autentifikacija svakog čvora koji se pojavi u BSM od strane glave klastera.		
	Napad selektivnog prosleđivanja		*Detection Using Acknowledgments* gde čvorovi koji se nalaze na ruti prosleđivanja sami detektuju napadača među sobom.	
Napadi koji se odnose na rutiranje [33]	Napad crvotočine [34]		Kombinacija principa zasnovanih na *Liteworp* konceptu i konceptu abnormalno visokih frekvencija.	
	Napad otvorene rupe [35]	Mehanizam zaštite koristi detekciju od odstupanja od očekivanog opterećenja procesora na čvoru.		
Napadi koji se odnose na identitet	Napadi imitiranja [36]		Zaštita zasnovana na autentičnosti paketa upotrebom više ruta.	
	Napad taloga [37]		Zaštita je zasnovana na sistemu deljenja ključnih delova za verifikaciju u senzorskim mrežama [47] poboljšanjem informacija o lokaciji senzorskih čvorova na posmatranom području [3].	
Napadi koji se tiču privatnosti [38]	Napad analize protoka [39] [40]		Bazna stanica imitira rad čvorova da se ne bi lako otkrila njena lokacija u BSM sa primenom u PA.	

U daljem tekstu je detaljan opis mehanizama zaštite od prethodno navedenih napada. Posebna pažnja posvećena je mehanizmima sa velikom energetskom efikasnošću senzorskih čvorova.

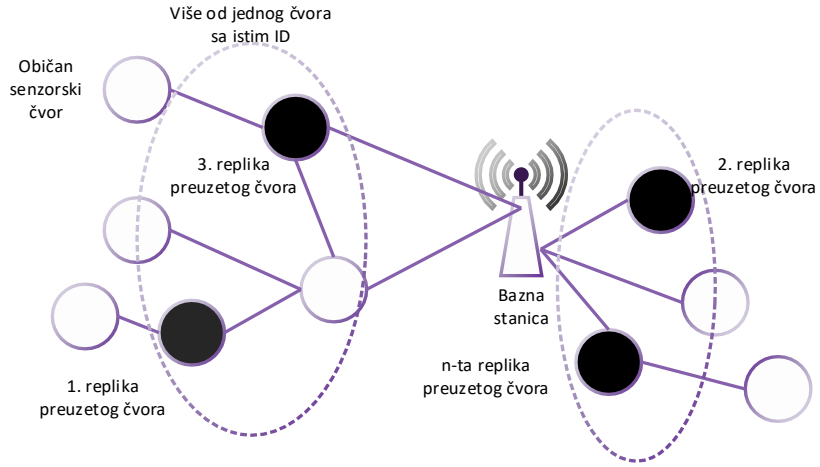
4.5.1. Napad hvatanja čvora – mehanizam detekcije i zaštite

Najbolji način za odbranu od napada hvatanja čvora *(*engl. *Node capture attack)* prikazan je na slici 4.13. i zasnovan je na principu da se posmatra stacionarna BSM, tako da senzori ne menjaju svoju poziciju. Pristup koji je predstavljen u literaturi [41] zasnovan je na sekvencijalnom testu verovatnoće (engl. *Sequential probability Ratio Test*).

Princip zaštite počiva na tome da se u bežičnim senzorskim mrežama lokacije senzorskih čvorova ne menjaju nakon implementacije, kao i da svaki senzorski čvor može da identifikuje izvore svih poruka od svojih suseda.

Primeri primene bežičnih senzorskih mreža zasnovani su na činjenici da znamo poziciju i gustinu postavljenih senzora, a za ovaj tip napada napadač mora fizički da preuzme senzor; možemo se pozvati na svojstvo ovog pristupa koje se zasniva na broju senzorskih čvorova koji mogu biti fizički uhvaćeni u određenoj regiji jer ako se poveća broj uhvaćenih senzora, povećava se i verovatnoća da se otkrije napadač. Pored ovoga, neophodno je određeno vreme za preuzimanje čvorova i za njihovo ponovno vraćanje u mrežu.

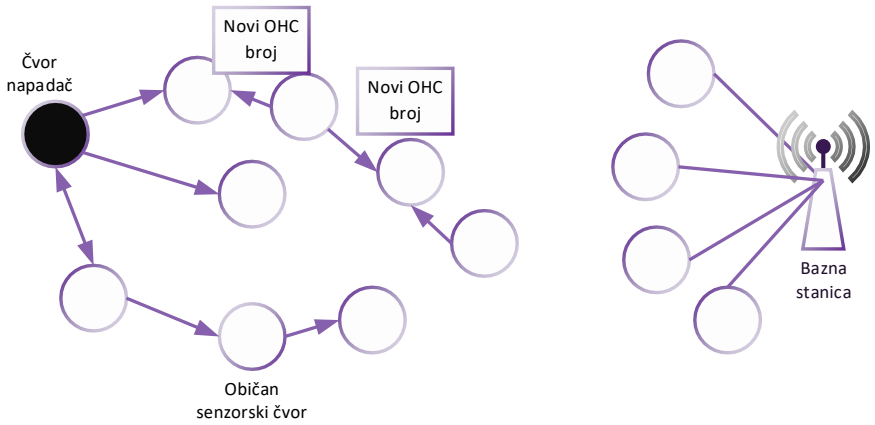
U statičkim mrežama senzora kao što je naša čvor se smatra kompromitovanim ukoliko se isti ID ponavlja na najmanje dve lokacije.



Slika 4.13. Napad hvatanja čvora – mehanizam detekcije i zaštite

4.5.2. Napad odbijanja sna – mehanizam detekcije i zaštite

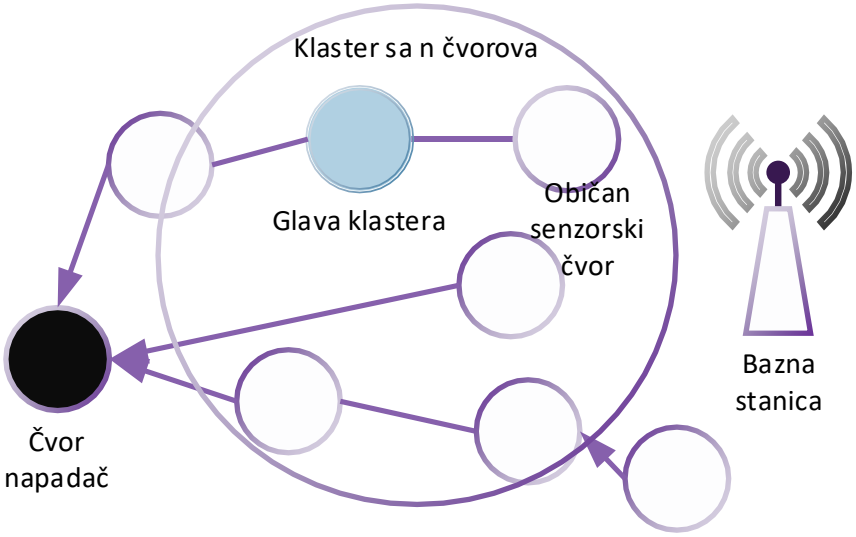
Napad odbijanja sna *(*engl*. Denial of sleep attack)* jeste napad na sloju izvora podataka i ugrožava same senzorske čvorove,a prikazan je na slici 4.14. [48]. Neprijatelj iz daljine preplavljuje senzorske čvorove tako što šalje lažne pakete ili ponavlja pakete koji su već preneseni kroz mrežu. U literaturi je predloženo rešenje pomoću OHC (*One-way Hash Chains*) kako bi se zaštitila komunikacija s kraja na kraj od permanentnog DoS-a. Svaki čvor je konfigurisan pomoću OHC-a, tako što srednjim čvorovima omogućuje permanentno otkrivanje DoS-a i sprečava širenje ponovljenih ili lažnih paketa. Ovde svaki paket uključuje novi OHC broj. Samo ako je OHC broj novi, srednji čvor predaje paket. Upotreba OHC broja sprečava neprijatelja da preplavi put ponovljenim paketima.



Slika 4.14. Napad odbijanja sna – mehanizam detekcije i zaštite

4.5.3. Napad preplavljivanja pozdravnim porukama – mehanizam detekcije i zaštite

Mehanizam zaštite od napada preplavljivanja pozdravnim porukama (engl*. Hello flooding attack)* počiva na velikoj energetskoj efikasnosti, spada u niskoenergetski prilagodljivu hijerarhiju klasterovanja (engl*. Low Energy Adaptive Clustering Hierarchy – LEACH*) i prikazan je na slici 4.15. Prema podacima koji se mogu naći u literaturi [49], ovaj mehanizam zaštite zasnovan je na podeli BSM na klastere. Na osnovu određivanja praga za broj čvorova koji pripadaju svakom od klastera određuje se glava (*head*) za svaki od klastera i na taj način pokreće mehanizam zaštite od ove vrste napada. Algoritam zaštite funkcioniše po principu detekcije glave klastera čiji broj članova je iznad praga koji je određen za svaki od klastera i na taj način se detektuje da li postoji uljez u mreži; međutim, potvrda da je zlonamerni čvor definitivno prisutan u mreži izvodi se na osnovu jačine primljenog signala i njegove udaljenosti.

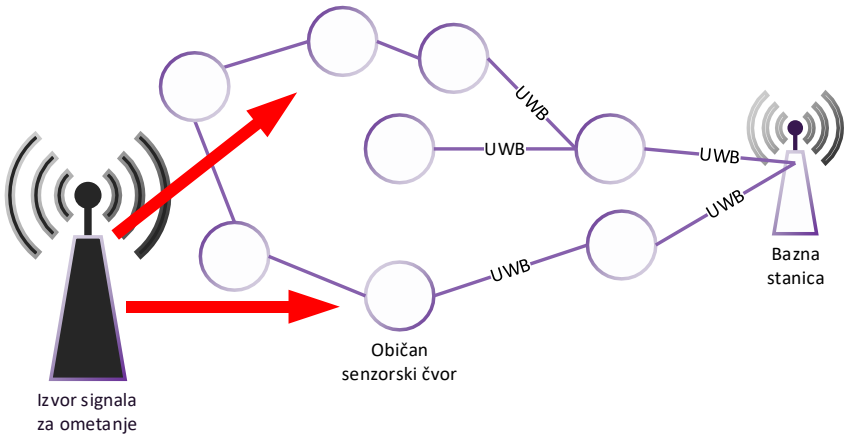


Slika 4.15. Napad preplavljivanja pozdravnim porukama – mehanizam detekcije i zaštite

4.5.4. Napad ometanja – mehanizam detekcije i zaštite

Protiv napada ometanja (engl. *Jamming attack*), koji je prikazan na slici 4.16., postoji nekoliko protivmera koje se koriste protiv ometanja, kao što su *Frequency Hopping Spread Spectrum* [50], *Direct Sequence Spread Spectrum*, *Ultra Wide Band Spectrum*, *Antenna Polarization*, *Directional Transmission* koje su opisane u preglednim radovima [28]. Neke od navedenih metoda su računski zahtevne i samim tim nisu energetski efikasne. *Direct Sequence Spread Spectrum* je računski zahtevna metoda, *Frequency Hopping Spread Spectrum* metoda zahteva konstantnu promenu frekvencija, dok su za *Antenna Polarization* i *Directional Transmission* potrebne suviše kompleksne antene.

Mehanizam zaštite koji privlači našu pažnju jeste upotreba tehnike modulacije ultraširokog opsega *(*engl*. Ultra Wide Band – UWB)* koja je zasnovana na emitovanju kratkih impulsa na veoma velikom frekvencijskom opsegu [51]. Ovo otežava prenošenje ili presretanje prenošenog signala i čini ga otpornim na efekte izazvane višestrukim prostiranjem (engl. *multipath*). Postoje istraživanja u kojima je predstavljen raspored senzorskih čvorova koji zahteva malo energije [52]. UVB takođe garantuje produženo trajanje baterije i preciznu lokalizaciju.



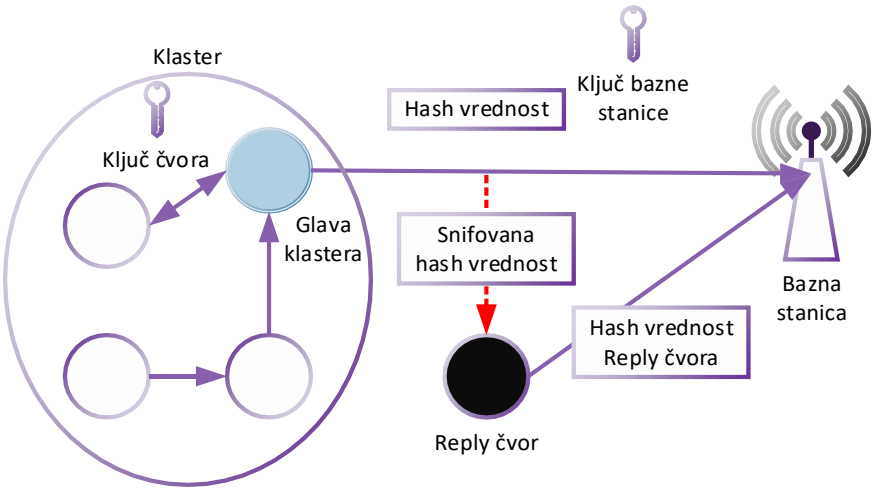
Slika 4.16. Napad ometanja – mehanizam detekcije i zaštite

4.5.5. Napad ponavljanja – mehanizam detekcije i zaštite

Protiv napada ponavljanja (engl. *Replay attack*) dobre rezultate u zaštiti dao je mehanizam na mreži koja je 100x100 metara i ima 100 čvorova nasumično raspoređenih [45]. Primena ovog modela prikazanog na slici 4.17. može lako da se skalira na veće površine, kao i na površine različitih proporcija a da ne ugrozi rezultate mehanizma zaštite koji su prikazani u literaturi. Polazimo od toga da nisu definisani klasteri, kao i da su heterogeni uslovi u mreži, te da nemaju svi čvorovi istu početnu energiju. Svaki čvor ima dva tajna (privatna) ključa. Jedan od ključeva koristi se za komunikaciju između čvorova. Drugi ključ koristi se za komunikaciju sa baznom stanicom, i to tek kada čvor preuzme ulogu glave klastera (*cluster head*).

Da bismo napadača odvratili od pristupa mreži, autentifikaciju treba izvršiti između glave klastera i čvorova koji treba da se pridruže mreži. Svaki čvor proverava autentičnost glave klastera (*cluster head*) pre nego što pošalje zahtev za pridruživanje mreži.

Autentičnost čvorova koji zahtevaju pridruživanje klasteru verifikuje se od strane glave klastera pre nego što oni postanu članovi tog klastera. Određeni čvor je glava klastera u jednom ciklusu. Izbor čvora koji treba da postane glava klastera u novom ciklusu vrši se pre kraja trenutnog ciklusa. Čvor koji je izabran za novu glavu klastera mora da se verifikuje na baznoj stanici preko postojeće glave klastera. Posle toga bazna stanica obaveštava čvorove o tome koji su čvorovi izabrani da budu glave klastera u sledećem ciklusu.



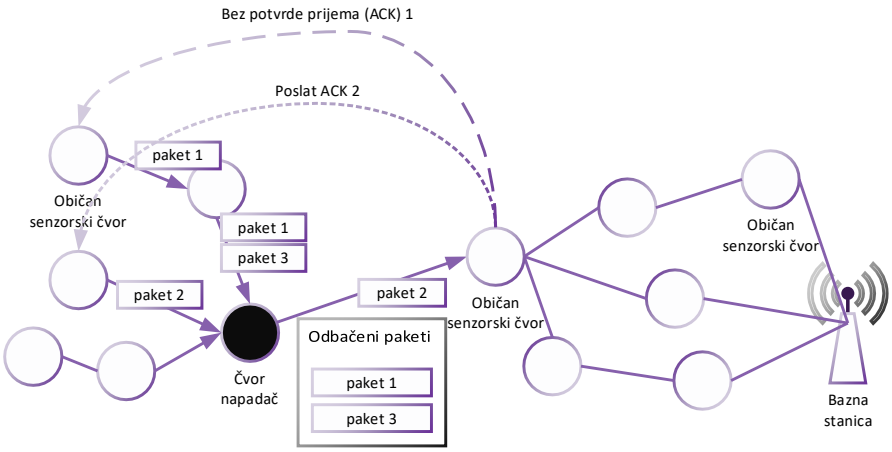
Slika 4.17. Napad ponavljanja – mehanizam detekcije i zaštite

4.5.6. Napad selektivnog prosleđivanja poruka – mehanizam detekcije i zaštite

Od napada selektivnog prosleđivanja *(*engl. *Selective forwarding attack)* u bežičnim senzorskim mrežama postoji više protivmera kao što su: detekcija priznanjem *(*engl. *Detection Using Acknowledgments*), detekcija lagane težine susednog čvora *(*engl. *Neighbour Node Light Weight Detection)*, šema otkrivanja više protoka podataka *(*engl. *Multi Data Flow Detection Scheme)*, otkrivanje u heterogenim mrežama *(*engl. *Detection in Heterogeneous Networks)* [53].

Zbog relativno malog opterećenja resursa kojima čvorovi raspolažu, najpogodniji mehanizam zaštite, prikazan na slici 4.18, zasnovan je na detekciji po osnovu priznanja. Za razliku od ovog mehanizma ostali mehanizmi daju podjednako dobre rezultate, ali su zahtevni u pogledu iscrpljivanja baterijskog napajanja čvorova [53].

Detekcija pomoću priznanja predstavlja šemu priznanja sa više skokova *(*engl. *multihop)* koja pokreće alarm, na osnovu odgovora iz drugih čvorova. Čvorovi koji se nalaze na ruti prosleđivanja imaju mogućnost da primete zlonamerni čvor unutar mreže. Srednji čvorovi prilikom otkrivanja zlonamernog čvora putem više skokova šalju poruku sa alarmom baznoj stanici. Postoje dva procesa detekcije. Prvi, nizvodno, označava da se vrši prenos podataka prema baznoj stanici iz izvora čvorišta, i drugi, uzvodno, označava prenos podataka prema izvornom čvoru od bazne stanice. Uključuje tri paketa za detekciju napada koji su nazvani paket priznanja, paket izveštaja i paket alarma [53].



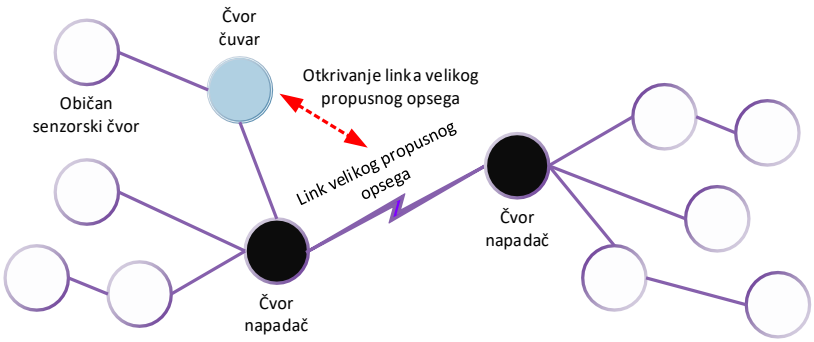
Slika 4.18. Napad selektivnog prosleđivanja poruka – mehanizam detekcije i zaštite

4.5.7. Napad crvotočine – mehanizam detekcije i zaštite

Najpogodniji mehanizam zaštite od napada crvotočine *(*engl. *Wormhole attack)* ne zahteva nikakva dodatna hardverska rešenja, kao ni potrebu da se unapred znaju lokacije čvorova i prikazan je na slici 4.19. Rešenje koje bi bilo adekvatno za primenu objedinjuje u sebi već dva dobro poznata principa – to je princip zasnovan na kombinaciji *Liteworp* koncepta i koncepta abnormalno visokih frekvencija na crvotočnom kanalu [54].

Liteworp koncept polazi od pretpostavke da topologija mreže pre samog inicijalnog pokretanja nije pod napadom crvotočine. Svaki čvor u fazi raspoređivanja zajedno sa svoja dva suseda prisluškuje nesusedne čvorove, koji biraju čvorove čuvare, te ukoliko se naruši taj poredak, oni detektuju da je došlo do napada [54].

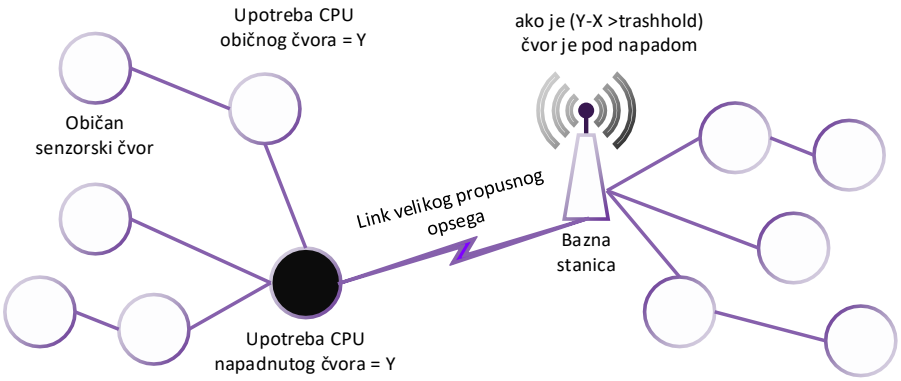
Koncept abnormalne učestalosti zasnovan je na osobini crvotočinih veza da imaju veliku učestalost slanja poruka. Na taj način se crvotočna veza razlikuje od normalne veze [47].



Slika 4.19. Napad crvotočine – mehanizam detekcije i zaštite

4.5.8. Napad otvorene rupe – mehanizam detekcije i zaštite

Jedna od karakteristika koja je uočena za čvor koji je pod napadom otvorene rupe *(*engl*. Sinkhole attack)* jeste povećanje opterećenja procesora, pa kao najadekvatniji predlog za detekciju ovog napada može poslužiti tehnika za izračunavanje razlike korišćenja CPU-a za svaki čvor praćenjem korišćenja procesora u fiksnom vremenskom intervalu. Korišćenjem ove razlike, bazna stanica određuje da li je čvor legitiman ili zlonameran, kao što je prikazano na slici 4.10. [55].

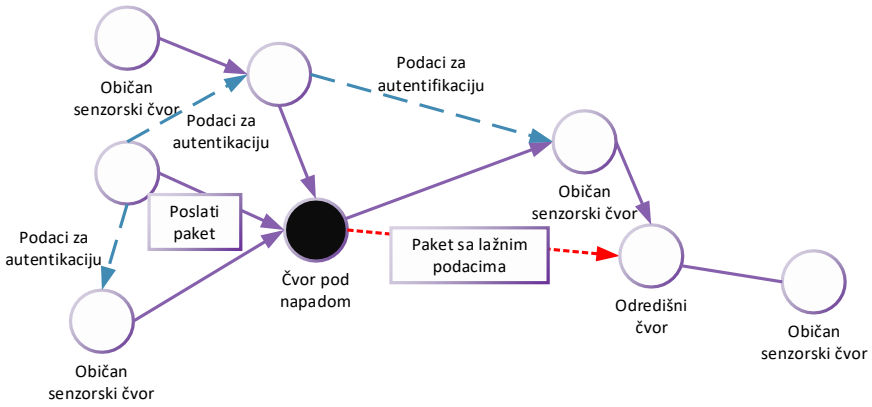


Slika 4.20. Napad otvorene rupe – mehanizam detekcije i zaštite

4.5.9. Napadi imitiranja – mehanizam detekcije i zaštite

Da bi se odbranili od napada imitiranja (engl. *Impersonate attack*), svi čvorovi BSM moraju da budu uvereni da su paketi autentični. Mehanizam prikazan na slici 2.21. daje dobre rezultate i zasniva se na metodu primene više ruta. Za registraciju *hash* vrednosti ID-jeva susednih čvorova na izvorišnom čvoru koristi se *Bloom* filter. Ovaj metod je nazvan autentični identitet *Bloom* filter (AIBF) i koristi se da obezbedi raštrkan prenos podataka za autentifikaciju paketa legitimnog izvorišnog čvora.

Pored toga, u predloženoj metodi koristi se *Source-initiated tree-based Routing* (SRIDR) [56], [57]. SRIDR se bazira na metodi rutiranja ID-jeva zasnovanoj na stablu i za svaki čvor je kreirana tabela rutiranja. Ovaj metod sastoji se od tri postupka: prenosa autentičnog identiteta, korišćenja autentičnog identiteta za otkrivanje falsifikata i otkrivanja napada lažnog predstavljanja.

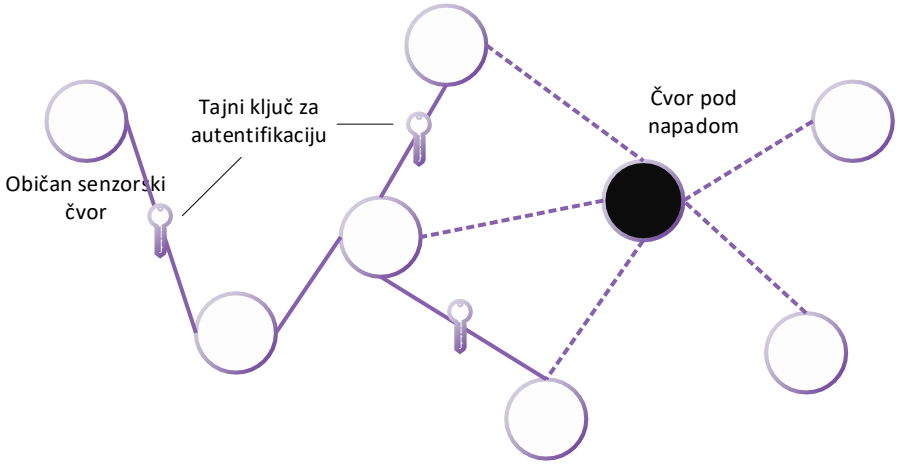


Slika 4.21. Napadi imitiranja – mehanizam detekcije i zaštite

4.5.10. Napad taloga– mehanizam detekcije i zaštite

Za sprovođenje napada taloga *(*engl. *Sibil attack)* napadači moraju da imaju više resursa kao što su procesorska snaga, jačina primopredajnika ili izvor napajanja, a ovi faktori su od presudnog značaja za sistem koji je zamišljen da obezbedi sigurnost i energetsku efikasnost.

Princip zaštite prikazan na slici 4.22. koncipiran je na tehnikama autentifikacije tajnim ključem. Nekoliko ključnih tehnika upravljanja predloženo je na osnovu deljenja ključnih delova za verifikaciju u senzorskim mrežama [[58], [59], [4]]. Kako bi se smanjila potreba za sistemskim zahtevima za šifrovanje, najpogodnije je koristiti informacije o lokaciji senzorskih čvorova na fizičkom sloju.



Slika 4.22. Napad taloga – mehanizam detekcije i zaštite

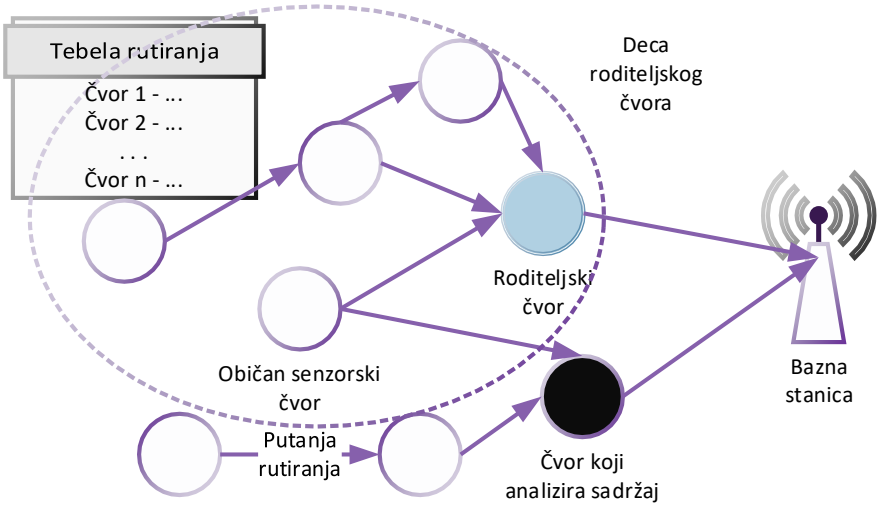
4.5.11. Napad analize protoka – mehanizam detekcije i zaštite

Kao mehanizam odbrane od napada analize protoka *(*engl. *Traffic analysis attack)* u literaturi [60] je prikazan mehanizam koji ima za cilj da reši problem otkrivanja

topologije i prenosa podataka. Mehanizam detekcije prikazan na slici 4.23. koristi otkrivanje topologije da ustanovi okolnosti do kojih je došlo usled nestanka energetskog napajanja nekog od čvorova i pokreće se periodično. Ukoliko je bežična senzorska mreža pod napadom analize sadržaja *(*engl. *traffic analysis)*, bazna stanica mora da poseduje informacije o stanju i poziciji čvorova, ali predloženi način za otkrivanje topologije ne ugrožava privatnost lokacije. Da bi se ovo postiglo, topologija se odvija u dve etape. Prva etapa obuhvata definisanje putanje rutiranja za svaki pojedinačni čvor, a potom se utvrđuje broj dece svakog od čvorova i svaki čvor informiše o broju dece čvorova koji se nalaze u blizini bazne stanice.

Kada senzorski čvorovi očitaju podatke, neophodno je da se ti podaci prenesu do bazne stanice. Kada se uspostavi protokol rutiranja u BSM, napadač je u mogućnosti da kreira stablo saobraćaja u mreži. Ovo je dobar način da se statističkom analizom saobraćaja u mreži odredi lokacija bazne stanice, jer ako posmatramo saobraćaj u mreži, čvorovi koji se nalaze neposredno uz baznu stanicu imaju najveći broj prenetih poruka, a bazna stanica je na samom vrhu tog stabla. Da bežična senzorska mreža ne bi bila podložna ovoj vrsti napada, neophodno je da svi čvorovi generišu podjednaku količinu saobraćaja a da se to ne odrazi na energetsku efikasnost samih čvorova. Da bi se ovo izbeglo, predlaže se mehanizam kojim se generišu poruke sa podacima od značaja za baznu stanicu, kao i poruke koje imaju zadatak da napadača dovedu u zabludu, tako da ne može statistički da obradi broj prenetih poruka i da sa sigurnošću odredi gde se nalazi bazna stanica.

Pored ostalih polja koja sadrže, ove poruke sadrže i TTL (engl. *Time to leave*) polje. TTL je u ovom pristupu polje koje definiše koliko je skokova do odredišta, i sa svakim skokom ova vrednost se smanjuje za jedan. Kada čvor inicira paket gde TTL ima vrednost 0, onda je to znak za susedni čvor da je taj paket lažan. Ukoliko paket koji treba proslediti ima TTL veći od 0, prijemni čvor dati paket smešta u bafer korisnih paketa koje treba proslediti dalje, te se na taj način kontroliše količina saobraćaja svakog čvora ponaosob.



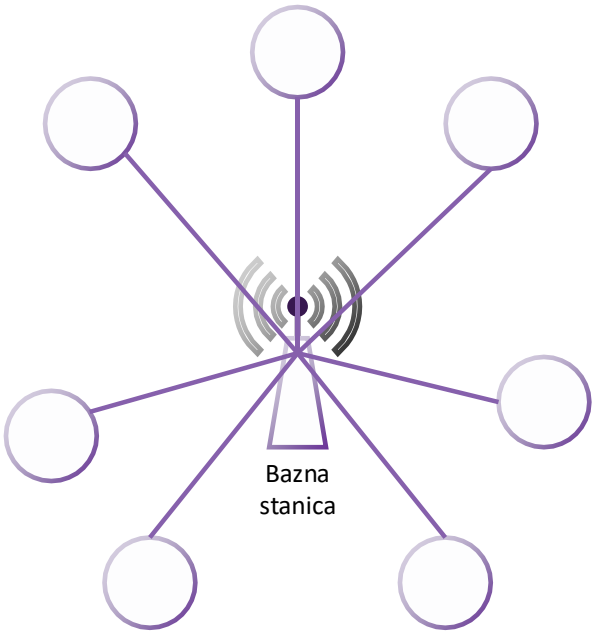
Slika 4.23. Napad analize protoka – mehanizam detekcije i zaštite

4.6. Komunikacione topologije u bežičnim senzorskim mrežama

Bežične senzorske mreže mogu da komuniciraju na različite načine i kroz različite topologije. U bežičnim senzorskim mrežama koriste se sledeće topologije: topologija zvezde, topologije mreže i hibridna topologija, koja predstavlja kombinaciju topologije zvezde i topologije mreže [61].

4.6.1. Topologija zvezde

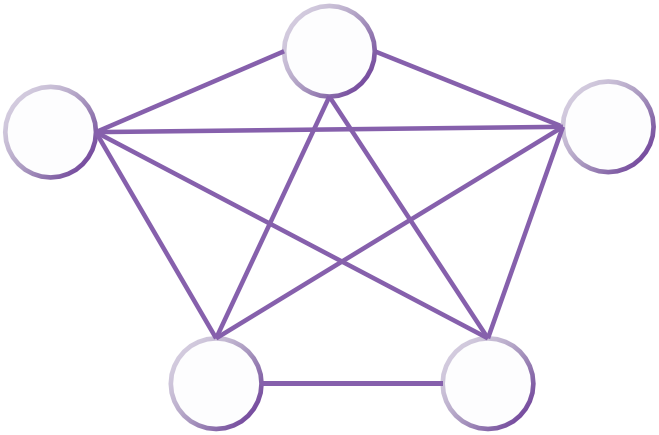
Topologija zvezde prikazana na slici 4. 24. jeste tako zasnovana komunikacija da jedna centralna bazna stanica može da komunicira, tj. da prima i šalje poruke senzorskim čvorovima koji su na određenoj udaljenosti od nje. Sva komunikacija čvorova ide direktno na baznu stanicu, tako da čvorovi ne mogu da komuniciraju između sebe. Kod ovakvog načina komunikacije obezbeđuje se maksimalni životni vek senzorskih čvorova, kao i direktan prenos podataka do bazne stanice, koja je centralni deo sistema. Takođe otkazom nekog od čvorova ne narušava se komunikacija unutar mreže i nema potrebe za ponovnim uspostavljanjem protokola za komunikaciju. Nedostatak ove topologije jeste to što je veličina senzorske mreže ograničena na domet bazne stanice, pa svi čvorovi koji se nađu van direktnog dometa bazne stanice nisu upotrebljivi u ovoj topologiji senzorske mreže [62], [61].



Slika 4.24. Bežična senzorska mreža – topologija zvezde

4.6.2. Topologija mreže – mesh

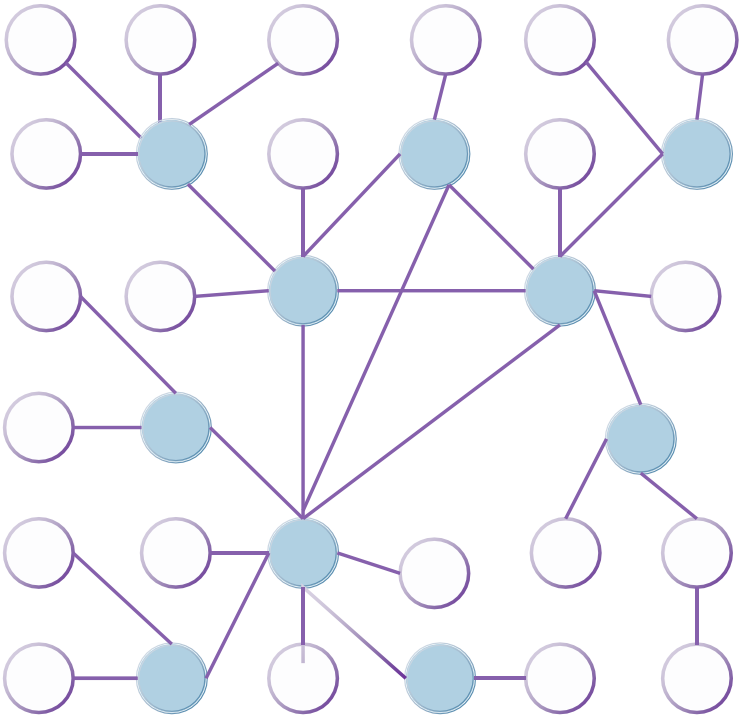
Topologija prikazana na slici 4.25. često se sreće u BSM i zasnovana je na međusobnoj komunikaciji čvorova koji se nalaze u komunikacionom opsegu. Svaki čvor može da kominicira sa svim čvorovima koji mu se nalaze u dometu, i na taj način preko susednih čvorova prenosi svoju poruku do bazne stanice koja mu je van dometa, ili ukoliko želi da komunicira sa nekim čvorom koji mu nije u neposrednom dometu, ta komunikacija mora da se izvrši posredno preko susednih čvorova. Ovim načinom komunikacije obezbeđuje se višestruka veza između čvorova, tj. redudantnost. U slučaju zakazivanja nekog čvora, komunikacija će se realizovati alternativnom rutom. Još je bitno napomenuti da u ovoj topologiji veličina posmatranog područja koje je pokriveno senzorskim čvorovima nije ograničena na domet bazne stanice. U slučaju otkaza velikog broja čvorova može doći do prekida komunikacije sa nekim delom mreže, ili će preostali čvorovi da budu izloženi velikoj količini saobraćaja koji treba da proslede i na taj način će brzo ostati bez napajanja [61], [62].



Slika 4.25 Bežična senzorska mreža – topologija mreže

4.6.3. Hibridna topologija

Hibridna topologija prikazana na slici 4.26. jeste kombinacija topologije zvezde i mreže. Glavna prednost ove topologije jeste to što produžava životni vek mreže i omogućava minimalnu potrošnju energije. Pored toga što je potrošnja energije minimalna, obezbeđene su robusnost mreže i skalabilnost bežične senzorske mreže. Čvorovi koji iscrpu svoje resurse služe samo za prikupljanje podataka, dok čvorovi koji imaju više resursa imaju ulogu čvorova za prosleđivanje poruka [62].



Slika 4.26. Bežična senzorska mreža – hibridna topologija

4.7. Struktura senzorskog čvora

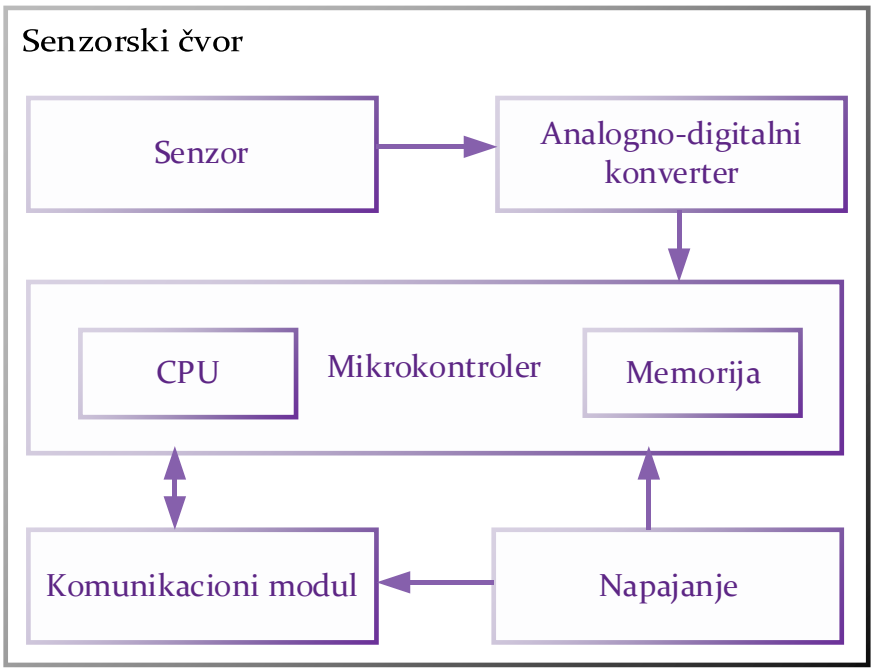
Svaki senzorski čvor u bežičnoj senzorskoj mreži da bi bio kompletan, mora da ima potpuno funkcionalne četiri osnovne komponente. Komponente senzorskog čvora su [61]: senzor za očitavanje vrednosti, mikrokontroler, komunikaciona komponenta i napajanje.

Bitno je napomenuti da su senzori koji služe za očitavanje vrednosti najčešće analogni, i da bi te očitane vrednosti mogle da se koriste za dalju obradu, neophodno je da uz analogne senzore ide i analogno-digitalni konvertor [63].

Mikrokontroler se u osnovi sastoji od procesorke jedinice koja obrađuje podatke i memorije koja služi za privremeno ili trajno skladištenje podataka [61]:

Komunikacioni modul služi za međusobnu komunikaciju između čvorova, kao i za komunikaciju sa baznom stanicom [61].

Napajanje senzorskih čvorova je po pravilu baterijsko; međutim, mogu da budu i drugi izvori napajanja [61]. Na slici 4.27. dat je šematski prikaz senzorskog čvora.



Slika 4.27 Komponente senzorskog čvora

4.8. Protokoli topologije u BSM

Zadatak protokola topologija u BSM je da optimizuje broj poruka i da smanji potrošnju energije, te na taj način produži životni vek posmatrane BSM. Protokoli koje posmatramo u ovom radu su: A3, A3 *coverage*, *Energy efficient connected dominating set* (EECDS) i CDS sa pravilom K.

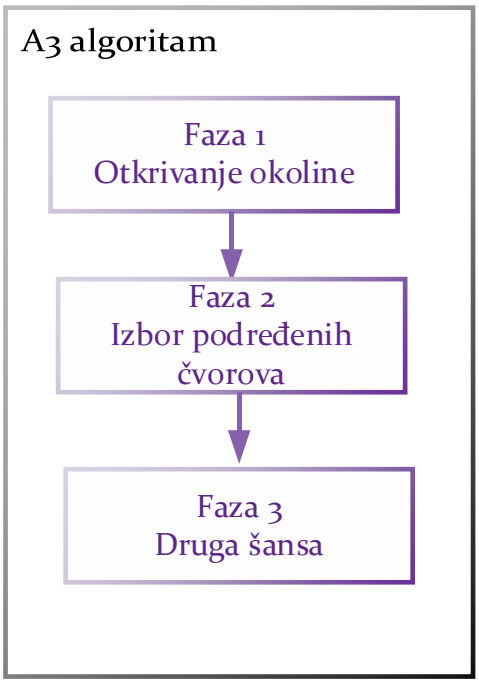
4.8.1. A3 protokol

A3 algoritam se zasniva na tome da nema saznanja o položaju ni o međusobnoj orijentaciji čvorova. Kako čvorovi nemaju informaciju o geometriji topologije, oni komunikaciju topologije zasnivaju na osnovu jačine primljenog signala susednih čvorova. Na osnovu informacije o jačini signala formiraju optimalno stablo CDS-a [64].

A3 algoritam se sastoji od tri faze [65]:

- otkrivanje okoline;
- izbor podređenih čvorova;
- druga šansa.

Grafički prikaz toka faza možemo da vidimo na slici 4.28:



Slika 4.28. Grafički prikaz faza uspostavljanja protokola topologije A3

4.8.1.1. Otkrivanje okoline

Otkrivanje okoline je proces koji počinje definisanjem unapred izabranog čvora za sinhronizaciju *(*engl. *sink node)* sa ciljem da se formira CDS. Protokol se aktivira tako što čvor za sinhronizaciju šalje pozdravne poruke. Na osnovu ove poruke okolni čvorovi treba da saznaju koji od njihovih suseda je njima nadređeni čvor. Pozdravne poruke dobijaju samo čvorovi koji su u komunikacionom radijusu čvora koji inicira konunikaciju [65].

Čvorovi koji prime pozdravnu poruku, a ne nalaze se unutar komunikacionog radijusa nekod drugog čvora, po automatizmu prihvataju inicijalni čvor kao roditeljski čvor i odgovaraju porukom o prepoznavanju i prihvatanju, a prihvataju i metriku zasnovanu na parametrima jačine signala čvora koji je poslao pozdravnu poruku, kao i energiju samog čvora koji prima poruku. Na osnovu vrednosti metrike nadređeni čvor pravi sortiranu listu susednih čvorova. Ako je čvor već ranije primio pozdravnu poruku, on zanemaruje sve pozdravne poruke koje stignu posle inicijalne. Ako nadređeni čvor ne primi poruku o prepoznavanju nadređenog čvora od susednih čvorova, on prelazi u stanje mirovanja [65].

4.8.1.2. Izbor podređenih čvorova

Roditeljski čvor prikuplja odgovore od svih suseda koji odgovore na njegovu poruku. Nakon što istekne vreme za prijem odgovora formira se sortirana lista sa

parametrima koje je roditeljski čvor dobio od podređenih čvorova. Na osnovu tih podataka roditeljski čvor šalje poruke svojoj deci, koje uključuju sortiranu listu metrike svih čvorova. Po prijemu liste kandidati određuju vreme pauze, u zavisnosti od njihove pozicije na listi, i čekaju poruku o spavanju od susednih čvorova. Pošto je uređena lista poruka o spavanju, najbolji čvor prvi šalje poruku i na taj način blokira sve svoje okolne čvorove. U slučaju da prime poruku o spavanju, oni se isključuju, a čvor koji je poslao poruku o spavanju postaje kvalifikovani čvor u stablu [65].

4.8.1.3. Druga šansa

Kod prethodno opisanog algoritma nastaje problem kada čvorovi koji su u režimu spavanja formiraju usko grlo u odrađenom delu bežične senzorske mreže. Svaki čvor koji primi poruku o spavanju spava određeno vreme, nakon čega on započinje proces formiranja stabla, jer će pronaći sve nepokrivene čvorove u svom okruženju i tako osigurati potpunu pokrivenost. Postupak kreiranja CDS-a u drugoj šansi ne razlikuje se od prethodno opisanog procesa, ali radi samo sa čvorovima koji nisu članovi nekog CDS-a. Proces se u potpunosti završava kada poslednji čvor okonča sopstveni proces kreiranja [65].

4.8.1.4. Proračun i upotreba metrike izbora

Metriku za odabir glave klastera kandidata za roditeljski čvor daje jednačina (4.1) [66], dajući prednost čvorovima s većom energijom i onima koji su dalje od roditeljskog čvora, očekujući da će izgraditi drvo sa manje čvorova i boljom pokrivenošću [66].

$$M_{x,y} = W_E * \frac{E_x}{E_{max}} + W_D * \left(\frac{RSSI_y}{RSSI_*}\right) \tag{4.1}$$

gde je M oznaka za metriku, x – regularan čvor, y – kandidat za glavu klastera, W_E je preostala energija mase čvora, E_x je preostala energija čvora x, E_{max} je maksimalna početna energija, W_D je težina udaljenosti od glave klastera, $RSSI_y$ je jačina signala čvora y u odnosu na čvor koji je glava klastera, a $RSSI_*$je minimalni $RSSI$ koji je neophodan da bi se dva čvora povezala [66].

Jednačina za računanje metrike daje prioritet čvorovima koji imaju veću energiju, ali i veću udaljenost od roditeljskog čvora. Cilj ove jednačine i određivanja ovakvih prioriteta jeste kreiranje stabla sa minimalnim brojem čvorova i većim stepenom pokrivenosti. Podešavanjem parametara možemo optimizovati pokrivenost

komunikacionog radijusa i smanjiti broj posredničkih čvorova u komunikaciji. Ovo postižemo tako što veći prioritet dajemo parametru udaljenosti. Glavna mana ovog pristupa jeste ta što u stablu mogu da se nađu i čvorovi sa niskim nivom energije, i tako smanje životni vek stabla. Pored parametrizovanja za vrednosti rastojanja, može da se parametrizuje i energija čvora, i na taj način da se unapredi pouzdanost stabla [65].

4.8.2. A3 coverage protokol

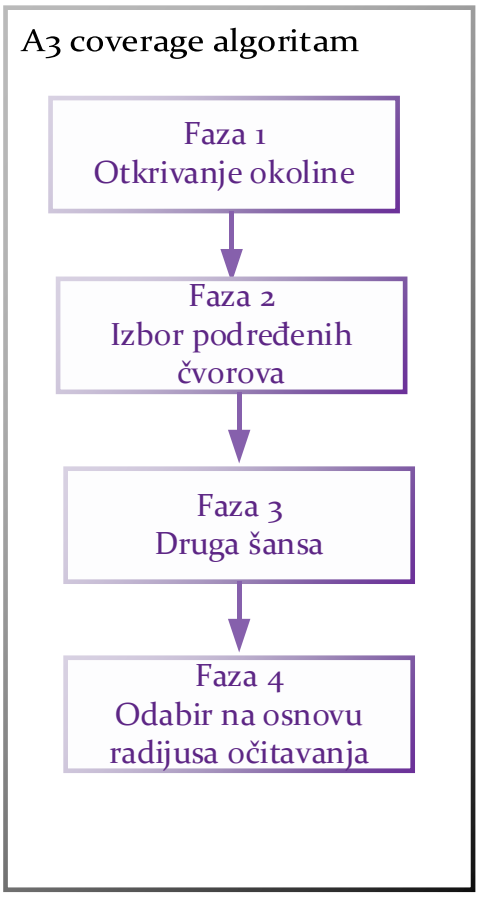
A3 *coverage* protokol je modifikacija A3 protokola i bazira se na radijusu očitavanja na osnovu koga se formira komunikaciono stablo [67]. Protokol zahteva da svi senzorski čvorovi u mreži budu u radijusu očitavanja makar jednog aktivnog susednog senzorskog čvora [68].

Čvor je sastavni deo mreže ako se nalazi u radijusu očitavanja nekog drugog aktivnog čvora. Na prvi pogled ovo izgleda kao da je opterećenje mreže, kao i broj potrebnih čvorova za komunikaciju značajno veći nego kod drugih protokola, međutim, u radu [69] prikazano je da je zbog preklapanja površina maksimalno povećanje od 50%. U ovom radu tvrdi se da je potrebno prvo odabrati čvorove koji su udaljeniji od aktivnog čvora, a nalaze se u njegovom opsegu pokrivenosti radijusa očitavanja. Kako su čvorovi u neposrednoj blizini aktivnog čvora i podaci koji bi se dobili njihovim očitavanjem zanemarljivi, jer za njih to već radi aktivan čvor, stoga oni odlaze u stanje u kome čuvaju energiju [67].

Za razliku od A3, A3 *coverage* protokol maksimalno koristi aktivne čvorove, da bi maksimalno proširio posmatrano područje, ali dobar deo čvorova u mreži u stanju je mirovanja, i mogu da zamene čvorove čija se energija potroši, bez uticaja na tačnost i preciznost očitanih podataka [68].

Algoritam A3 *coverage*, isto kao i A3, polazi od pretpostavke da senzorski čvorovi nemaju informacije o lokaciji susednih čvorova, već da se položaj određuje na osnovu jačine signala. S obzirom na polaznu postavku da što su čvorovi više udaljeni, to je veća pokrivenost, bitna nam je jačina signala kao metrika za određivanje udaljenosti [67].

Protokol se sastoji iz tri faze koje obuhvataju i A3, a to su: otkrivanje okoline, odabir podređenih čvorova, druga šansa i još jedna dodatna faza odabira na osnovu radijusa očitavanja [67]. Grafički prikaz toka faza možemo da vidimo na slici 4.29:



Slika 4.29. Grafički prikaz faza uspostavljanja protokola topologije A3 coverage

4.8.2.1. Otkrivanja okoline

Sam početak ove faze isti je kao i kod A3 protokola, tako da ovde nećemo zalaziti u detalje. Ukratko rečeno, unapred odabran čvor, *sink* čvor, svoje stanje menja u aktivno i šalje pozdravnu (*hello*) poruku. Na osnovu ove poruke susedni čvorovi prepoznaju roditeljski čvor. Ako čvor primi samo jednu pozdravnu poruku, postavlja čvor koji je slao poruku kao nadređeni čvor, a ako je već pokriven, onda zanemaruje ovu poruku. Poruka o prepoznavanju uključuje podatke o metrici na osnovu jačine signala i trenutne energije čvora. Da bi formirao CDS stablo u protokolu A3 *coverage*, potrebno je uzeti informaciju i o radijusu očitavanja; u slučaju da se ne nalazi u radijusu očitavanja roditeljskog čvora, čvor definiše vreme pauze da bi sačekao poruku o senzorskoj pokrivenosti susednih čvorova, u suprotnom se sam isključuje [67].

4.8.2.2. Proces odabira podređenih čvorova

Tokom pauze za dobijanje odgovora o prepoznavanju od svojih roditeljskih čvorova, potrebno je sačekati da se primi makar jedna poruka o prepoznavanju. Po prijemu takve poruke nadređeni čvor menja svoje stanje u aktivno. Posle isteka vremena

za pauzu, svaki nadređeni čvor prema metrici pravi opadajuću listu čvorova. Posle toga nadređeni čvor šalje poruku sa tom kompletnom sortiranom listom čvoru koji je unutar nejgove senzorske pokrivenosti. Kada čvor primi ovu poruku, menja svoj status u čvor kandidat i postavlja vremensku pauzu čije je trajanje proporcionalno njegovoj poziciji na listi. I tako čeka blokirajuće poruke. Blokirajuće poruke šalju između sebe čvorovi koji nisu roditeljski čvorovi. Ako čvor primi poruku u kojoj vidi da se neki od susednih čvorova bolje kotira od njega, tada menja svoj status u stanje spavanja. Čvor sa najboljom metrikom prvo šalje poruku o blokadi braće, i na taj način blokira sve druge čvorove u svom opsegu očitavanja. Ovaj proces ponovo pokreću čvorovi koji nisu u području pokrivenosti nekog od prethodnih čvorova [67].

4.8.2.3. Proces druge šanse

Postoje i slučajevi gde čvor koji je u stanju spavanja može da blokira određeni deo mreže. Kada čvor primi poruku o blokadi, on postavlja vremensko ograničenje za stanje blokade, i posle isteka tog vremena ponovo započinje proces otkrivanja okoline [67].

4.8.2.4. Izbor na osnovu radijusa očitavanja

Ova faza je jedna od bitnih razlika protokola A3 *coverage* u odnosu na regularni protokol A3. Čvorovi koji tokom prve faze nisu pripali CDS stablu i koji se ne nalaze u radijusu očitavanja ostaju budni dodatni vremenski period. Čvorovi koji nisu deo početnog stabla a prime poruku o pokrivenosti radijusom očitavanja pre isteka perioda budnosti prelaze u režim spavanja. Ako je čvor pod komunikacionim radijusom ali ne i u opsegu radijusa očitavanja susednih čvorova, on još nema uslov da bude osnov za proširenje površine područja radijusa očitavanja. Kad vreme predviđno za pauzu istekne a senzor se i dalje ne nalazi u radijusu očitavanja susednog senzora, on prelazi u aktivno stanje. Senzor u aktivnom stanju započinje proces slanja poruke o pokrivenosti radijusom očitavanja [67].

A3 *coverage* je u potpunosti distribuiran protokol i ne zahteva bilo kakva pravila za sinhronizaciju. Čvor može da počne da obavlja svoju primarnu ulogu odmah pošto bude odabran da bude sastavni deo CDS stabla [67].

4.8.3. Energetski efikasan povezani dominantni skup (EECDS)

Svaki pojedinačni čvor u bežičnoj senzorskoj mreži ima isti komunikacioni radijus. Algoritam pod nazivom Energetski efikasan povezani dominantni skup *(*engl. *Energy*

efficient connected dominating set – EECDS) sastoji se iz dve faze. Prva faza je računanje maksimalno nezavisnog skupa (MIS) na osnovu postojećeg rasporeda čvorova, dok je u drugoj fazi neophodno odabrati minimalni broj čvorova da bi se dobio povezani CDS [70]. Grafički prikaz toka faza možemo da vidimo na slici 4.30:

Slika 4.30 Grafički prikaz faza uspostavljanja protokola topologije EECDS

4.8.3.1. Računanje minimalnog nezavisnog skupa (MIS)

Algoritam koristi boje da označi da li čvor pripada ili ne pripada MIS-u. Čvor koji u bežičnoj senzorskoj mreži inicira komunikaciju i menja svoje inicijalno stanje u crno na taj način sam sebe proglašava za deo maksimalno nezavisnog skupa (*Maximal independent set* – MIS). Stanje ostalih čvorova ostaje bele boje, tj. još nemaju nikakvu ulogu u mreži. Posle promene svoga stanja crni čvor šalje poruku kojom obaveštava čvorove u svom komunikacionom radijusu da oni pripadaju istom MIS-a. Svi čvorovi koji dobiju poruku od čvora koji je inicirao komunikaciju menjaju svoje stanje u sivu boju, te poruku o svom stanju šalju svim svojim susednim čvorovima. Čvorovi koji nisu promenili svoje inicijalno stanje, beli čvorovi, a primili su sivu poruku, sada treba da odrede da li treba da promene svoje stranje u crno i da kreiraju zaseban MIS ili da se priključe nekom postojećem MIS-u. Da bi odredili da li u svom komunikacionom radijusu imaju neki crni čvor, oni šalju upitnu poruku o stanju susednih čvorova. Ukoliko ne dobiju odgovor od nekog čvora koji je već svoje stanje prebacio u crno, onda oni postaju crni čvor i ovo se ponavlja u više iteracija [70].

4.8.3.2. Konstrukcija CDS-a

MIS koji je kreiran u prethodnoj fazi potrebno je povezati u CDS da bi mogao da se obavlja prenos kroz bežičnu senzorsku mrežu. Algoritam se sastoji od toga da svaki MIS čvor odabere susedni ne-MIS čvor sa najvećim ponderom koji bi povezao dva ili više maksimalno nezavisnih skupova.

MIS čvor ima tri moguća stanja: crno, crno-prelazno i plavo, dok čvor koji nije MIS može biti u jednom od tri stanja: sivom, sivom-prelaznom i plavom. Po izvršavanju algoritma čvorovi u mreži moraju da budu ili sive ili plave boje, plavi čvorovi su oni cvorovi koji čine CDS [70].

Konstrukcija CDS se vrši na osnovu tri vrste poruka [70]:

- Plava poruka – potvrda da je neki čvor sastavni deo CDS čvora.
- Pozivna poruka – šalje je MIS čvor susedu da promeni svoje stanje u čvor za povezivanje.
- Ažuriranje poruka – čvor koji nije deo MIS-a obaveštava susede o svom trenutnom ponderu. Ponder čvora se računa na osnovu stanja baterije i broja njegovih MIS suseda koji su trenutno u crnom stanju.

Nakon inicijalne faze čvor ima status crnog čvora, što znači da pripada MIS-u, ili sivog čvora, što znači da ne pripada MIS-u. Čvor započinje sledeću fazu algoritma kada svi susedni čvorovi pređu u crno ili sivo stanje. Čvor započinje drugu fazu slanjem plave poruke i na taj način označava sebe kao CDS čvor [70].

Drugu fazu algoritma može da započne sivi čvor koji nije MIS. Sivi čvor koji primi PLAVU poruku treba opet da preračuna svoj ponder i da pošalje poruku o ažuriranju susednim čvorovima. Čvor, nakon slanja poruke za ažuriranje, menja svoje stanje u prelazno sivo. Ako je čvor koji ima sivo stanje primio poruku za ažuriranje, on će je samo odbaciti. Čvor u *crno-prelaznom* stanju ispituje mrežu da bi utvrdio da li može da postane čvor za povezivanje [70].

Čvor koji je u sivom prelaznom stanju može da primi tri vrste poruka: poruku za ažuriranje, plavu i pozivnu poruku. Postoje tri scenarija da čvor u *crno-prelaznom* stanju primi [70]:

- Poruku ažuriranja – ovu poruku ignoriše.

- Plavu poruku – poruka obaveštenja da je drugi čvor bolji kandidat za ovo povezivanje, i da on treba da vrati svoje stanje u sivo.
- Pozivnu poruku – on postaje čvor za povezivanje, menja svoje stanje u plavo, postaje CDS čvor i emituje plavu poruku.

Odgovorom na primljene poruke čvor u crnom stanju započinje drugu fazu algoritma. Postoji više tipova poruka koje crni čvor može da primi [70]:

- Plava poruka – kada ovu poruku dobije od povezujućeg čvora menja svoje stanje u plavo i postaje CDS čvor.
- Poruka ažuriranja – kada primi ovu poruku od susednih čvorova koji nisu MIS, definisaće vreme pauze i menja svoje stanje u *crno-prelazno* dok ne istekne vreme za pauzu.

Senzorski čvor u crno-prelaznom stanju tokom vremena za pauzu može da primi dve vrste poruka [70]:

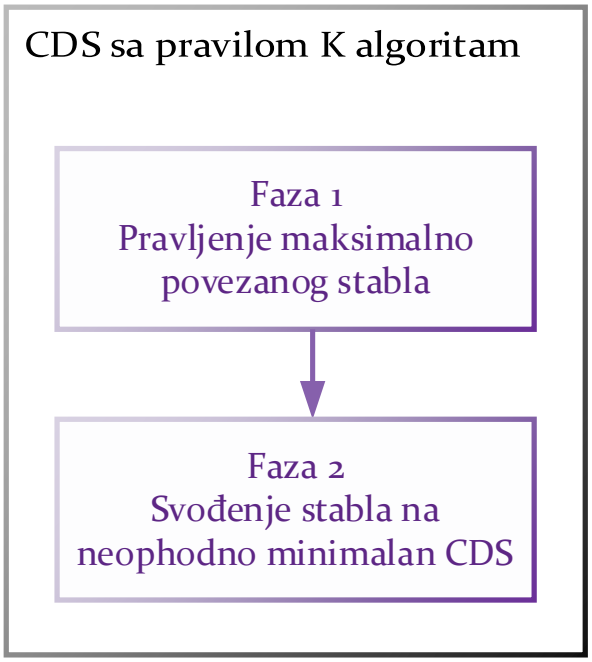
- Poruku ažuriranja – ovu vrstu poruka odbacuje.
- Plavu poruku – ovo stanje nastaje kada čvor već ima povezan čvor, menja svoje stanje u plavo, i šalje plavu poruku da bi se označio kao CDS čvor.

U suprotnom, kada pauza u *crno-prelaznom* stanju istekne, čvor bira ne-MIS suseda sa najvećim ponderom koji je dobio u porukama ažuriranja, šalje mu pozivnu poruku i menja svoje stanje u crno [70].

Algoritam konstrukcije CDS-a nastavlja se sve dok svi čvorovi koji su deo MIS-a ne završe algoritam ili čvorovi koji nisu deo MIS-a promene svoje stanje u plavo ili pak svi njegovi susedi budu u plavoj i sivoj boji (59).

4.8.4. CDS sa pravilom K

Algoritam CDS sa pravilom K sastoji se od dve faze, gde je inicijalna faza pravljenje maksimalno povezanog stabla, a druga je svođenje stabla na neophodno minimalan CDS [66]. Grafički prikaz toka faza možemo da vidimo na slici 4.31:



Slika 4.31. Grafički prikaz faza uspostavljanja protokola topologije CDS-a sa pravilom K

Prva faza definisana je jednačinom (4.2) koja sledi, a predstavljena je u [66]:

$$S = (\forall v \in S: x, y \in N(v), \neg\exists(x, y) \in V). \tag{4.2}$$

U formuli je prikazan *N(v)* kao skup susednih čvorova *v*. Formula definiše da se čvor *v* koji ima dva ili više suseda, koji nisu međusobno povezani, dodaje na početno stablo. Čvorovi prenose *hello* poruke koje sadrže listu susednih čvorova. Po prijemu *hello* poruke, čvor poredi podatke iz poruke sa listom svojih suseda. Čvor se dodaje u stablo ako je broj suseda u njegovoj listi veći od broja iz primljene liste [66].

Na osnovu K pravila skraćuje se maksimalizovano početno stablo. Sinhronizacioni čvor je čvor najnižeg novoa u početnom stablu, a susedni čvorovi označeni su za po jedan sloj više. Slojevi nižeg nivoa su čvorovi sa većim prioritetom [66].

Pravilo X obezbeđuje da se minimalizuje komunikaciono stablo po pravilu da čvor prelazi u stanje mirovanja ako su svi susedni čvorovi u komunikacionom radijusu čvora većeg prioriteta, ili ako su svi susedni čvorovi pokriveni komunikacionim radijusom drugih dvaju povezanih čvorova [66].

Ovaj protokol je osetljiv na promene topologije. Svaki čvor može da detektuje 4 stanja: pojavljivanje novog suseda, nedostatak susednog čvora, približavanje čvora, udaljavanje susednih čvorova. Ovi problemi rešavaju se lokalizovano postupkom

označavanja i pravilom K. Po otkrivanju neke od promena u topologiji primenjuju se postupak označavanja i ograničeno pravilo K koji služe za proračun novog statusa čvora [66].

Čvor pokreće proces označavanja na osnovu 1 skoka *(*engl. *hop)*, gde pokazuje da li je čvor uključen ili isključen, ili na osnovu 2 skoka, gde pokazuje da li je veza uključena ili isključena [66].

Ograničenje na osnovu pravila K može se pokrenuti samo promenama u okviru:

- 1 skoka – gde detektuje da li je čvor uključen ili isključen;
- 2 skoka – gde detektuje da li je veza između čvorova uključena ili isključena;
- 3 skoka – gde detektuje promenu statusa susednih čvorova.

Opseg širenja bilo koje promene na topologiji bežične senzorske mreže ne može da ima više od 3 skoka [66].

Ograničenja gornje granice pravila K na 2 skoka dovodi do toga da zavisi samo od statusa veze između suseda udaljenosti do 2 skoka. Ovo ograničenje dovodi do povećanja računarske zahtevnosti, ali ima niži ukupan trošak komunikacije [66], [71].

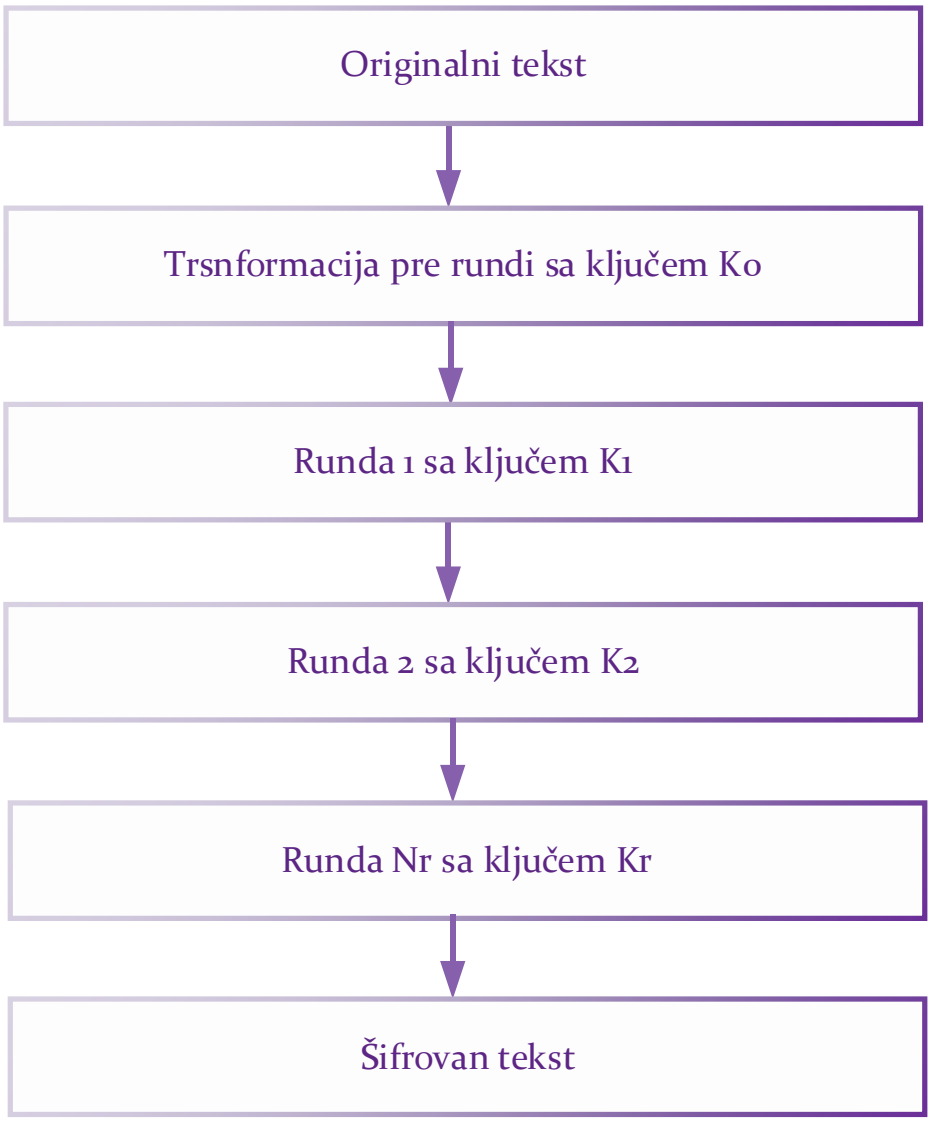
4.9. Šifarski algoritmi koji nisu računski zahtevni (lagani)

Pod pojmom lagani šifarski algoritam podrazumevamo algoritam koji je pogodan za upotrebu na nekoj platformi koja ima ograničene resurse. Uvidom u literaturu koja se bavi temom *Lightweight Block Cipher Algorithams*, sveobuhvatan pregled načinili su Hadživasilis (Hatzivasilis) i ostali [72], koji su između ostalog obrađivali šifarske algoritme čije osobine su osnov za simulacioni scenario i rezultate koje ćemo prikazati u ovom radu. Od glavnog značaja za naš rad su sledeći šifarski algoritmi: *AES*, *NOEKEON*, *PRESENT*, *LED*, *Piccolo*, *TWINE*, *KATAN*, *KATANAN*, *PRINCE*, *SIMON*. Nabrojane šifarske algoritme predstavljamo u kratkoj formi i prikazujemo samo osnovne parametre koji su nam od značaja za dalje istraživanje.

4.9.1. Napredni standard enkripcije

Napredni standard enkripcije (*AES – Advanced Encryption Standard*) je šifarski algoritam gde podaci imaju dužinu od 128 bitova, i može da koristi ključeve dužine 128, 192 i 256 bitova. AES otvoren tekst od 128 bitova posmatra kao matricu 4x4 koja je

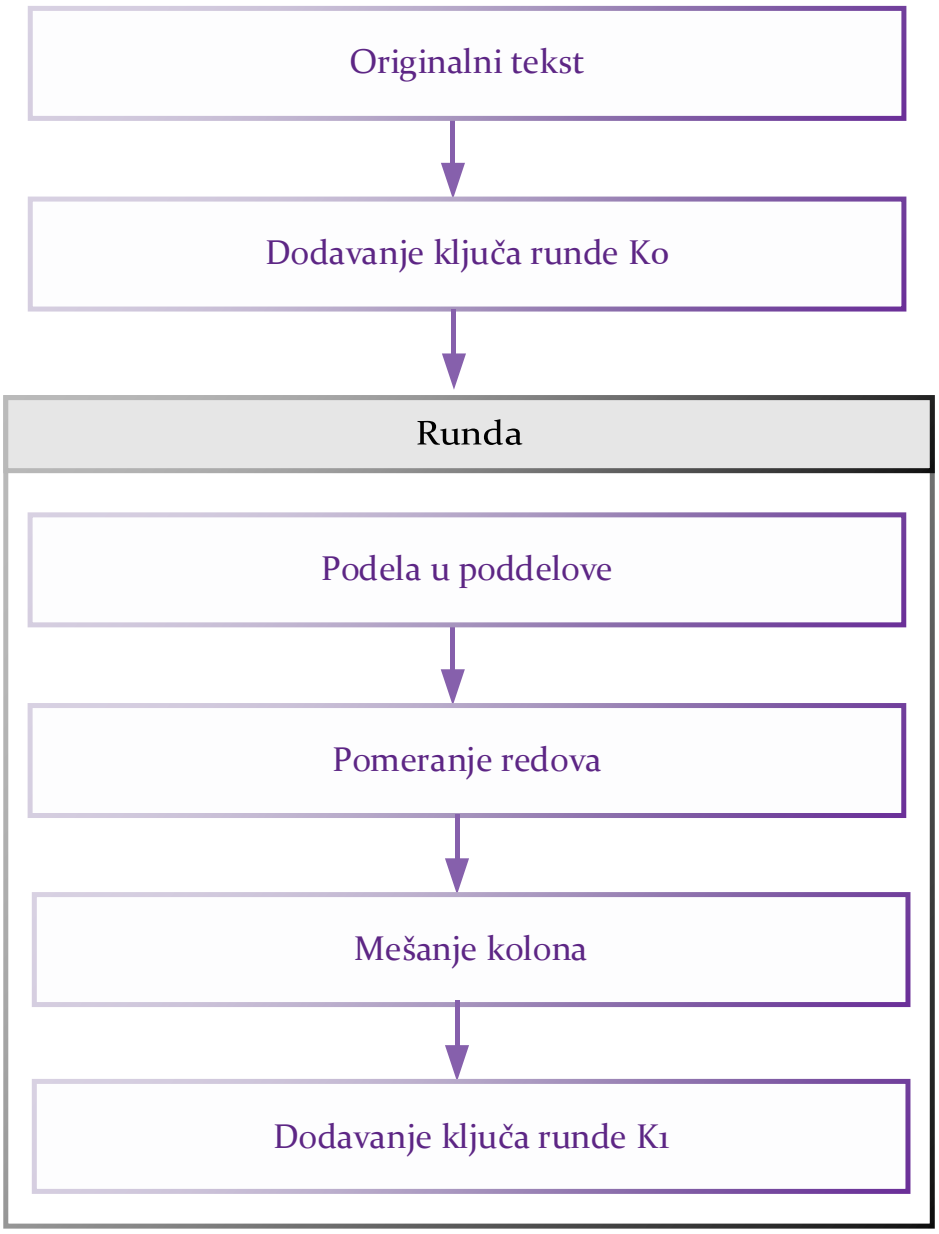
popunjena vrednostima dužine 16 bajtova. U zavisnosti od dužine ključa, ovaj šifarski algoritam koristi 10 rundi za ključeve dužine 128 bitova, 12 rundi za ključeve dužine 192 bita i 14 rundi za ključeve dužine 256 bitova. U svakoj od rundi koristi se ključ od 128 bitova, koji se računa iz originalnog AES ključa. Postupak šifrovanja prikazan je na slici 4.32. [73].



Slika 4.32. Postupak šifrovanja AES algoritmom

4.9.1.1. Proces šifrovanja

Svaka runda šifrovanja sastoji se od četiri potprocesa. Proces prvog kruga prikazan je na slici 4.33. [73].



Slika 4.33. Prva runda šifrovanja AES algoritmom

4.9.1.2. Zamena bajtova (podbajtova)

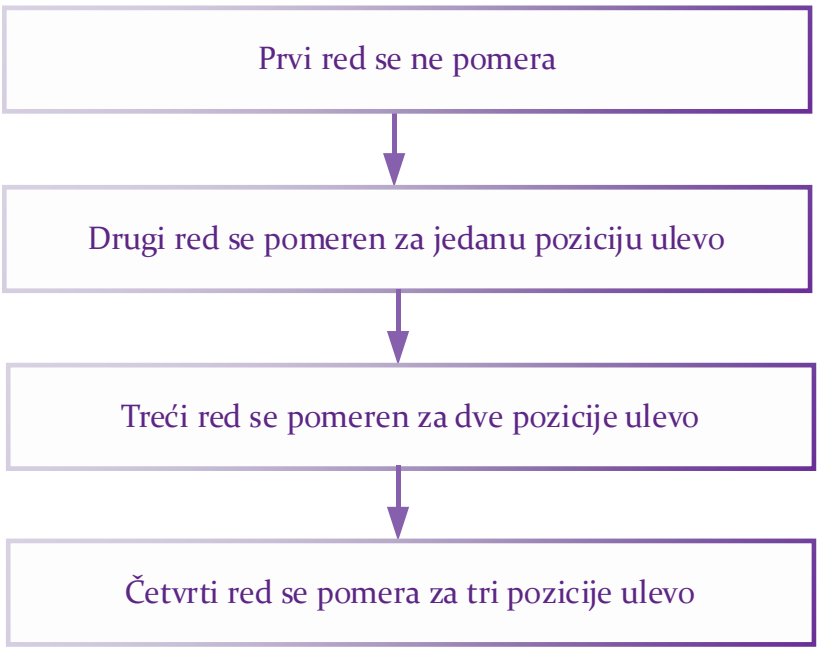
Svaki ulaz dužine 16 bitova menja se pomoću S-kutije koja je prikazana u tabeli 3. Kao rezultat dobija se matrica od četiri reda i četiri kolone [73].

Tabela 3. S kutija AES algoritma

S(rs)		s															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
r	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

4.9.1.3. Pomeranje redova

Svaki od četiri reda matrice pomera se ulevo. Pomeranjem u levu stranu sve ono što preostane prebacuje se na desnu stranu. Šematski prikaz pomeranja redova dat je na slici 4.34. Rezultat je nova matrica koja se sastoji od 16 bajtova koji su pomereni [73].



Slika 4.34. Postupak pomeranja redova u AES algoritmu

4.9.1.4. Mešanje kolona

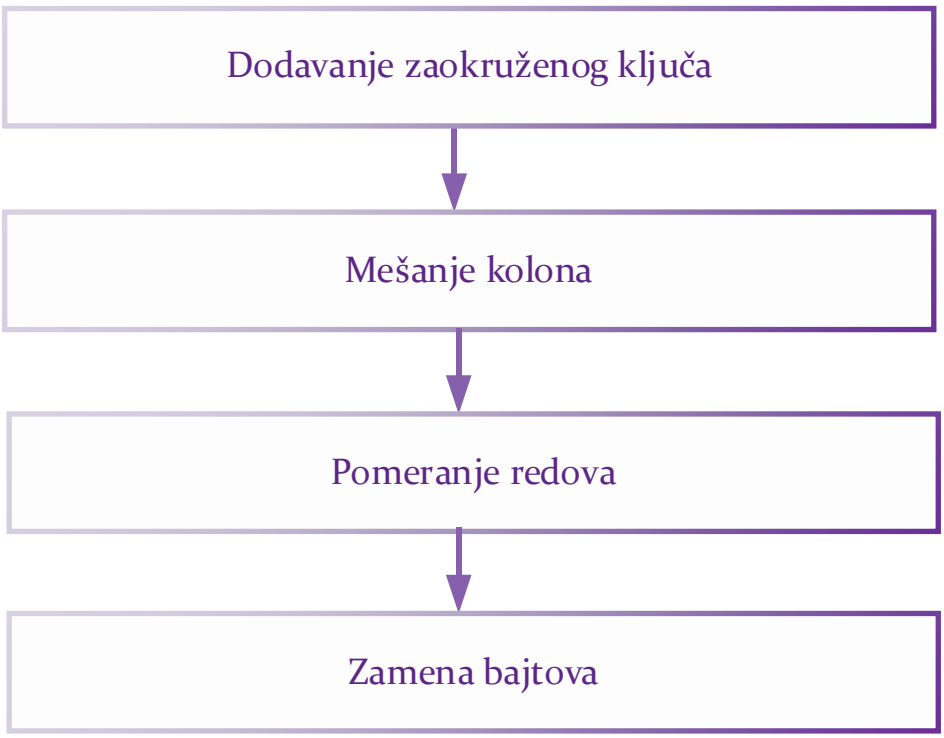
Svaka kolona od četiri bajta transformiše se pomoću posebne matematičke funkcije. Kao ulaz ove funkcije uzimaju se četiri bajta jedne kolone i izbacuju četiri potpuno nova bajta. Kao rezultat ove operacije dobija se nova matrica veličine 16 bajtova. Poslednja runda razlikuje se po tome što se ovaj korak u njoj ne primenjuje [73].

4.9.1.5. Dodavanja ključa runde

Nad matricom veličine 16 bajtova izvršena je operacija XOR sa ključem zaokruženim na dužinu od 128 bitova. Ako se ovo dešava u poslednjoj rundi, onda je proces šifrovanja završen i njegov izlaz je tekst koji je šifrovan. Runde šifrovanja se ponavljaju sve do poslednje runde [73].

4.9.1.6. Proces dešifrovanja

Proces dešifrovanja AES šifarskog algoritma ima obrnut redosled operacija od procesa šifrovanja. Svaka runda sastoji se od četiri procesa koja su izvedena kao na shemi sa slike 4.35. Za dešifrovanje je potrebno koristiti inverznu S-kutiju koja ima obrnuti redosled, obrnutu linearnu transformaciju i obrnuti redosled potključeva [73].

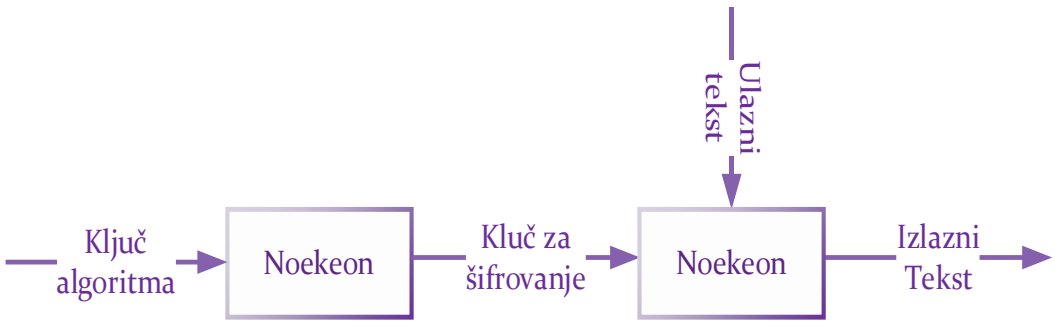


Slika 4.35. Proces dešifrovanja AES algoritma

4.9.2. NOEKEON

NOEKEON je šifarski algoritam čija je dužina bloka ista kao i dužina ključa koji iznose po 128 bitova. Koristi 16 rundi, a u svakoj rundi izvode se tri transformacije. Transformacije se zovu: teta, pomeranje ofseta pomoću transformacija Pil1 i Pil2, i gama [74].

Raspoređivanje ključa od 128 bitova zasnovano je na režimu direktnog ključa (*direct key mode*) da bi eliminisao napade zasnovane na povezanim ključevima, kao što je prikazano na slici 4.36. [74].

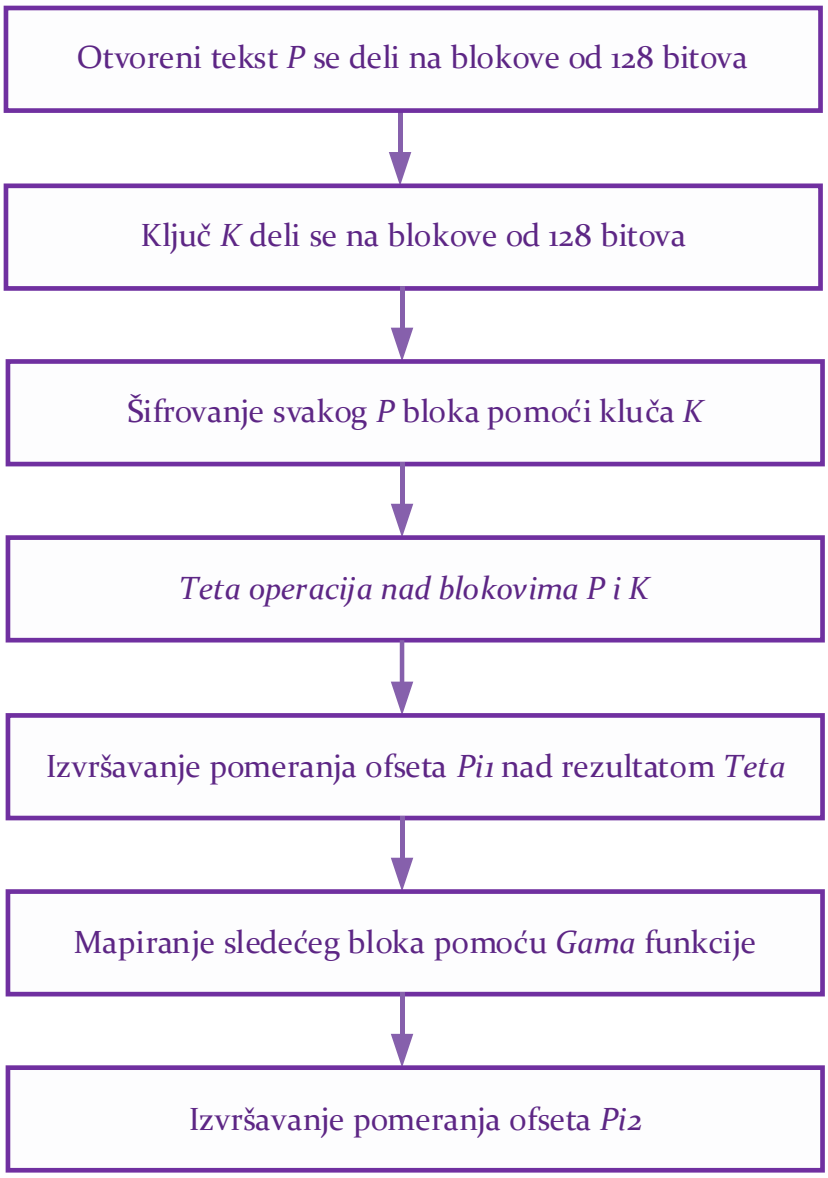


Slika 4.36. Režim indirektnog ključa NOEKEON algoritma

Pre nego što se ključ primeni na poruku potrebno ga je pretvoriti u drugi ključ, tek posle toga se primenjuje na poruke kroz ostale transformacije u narednih 16 rundi [74].

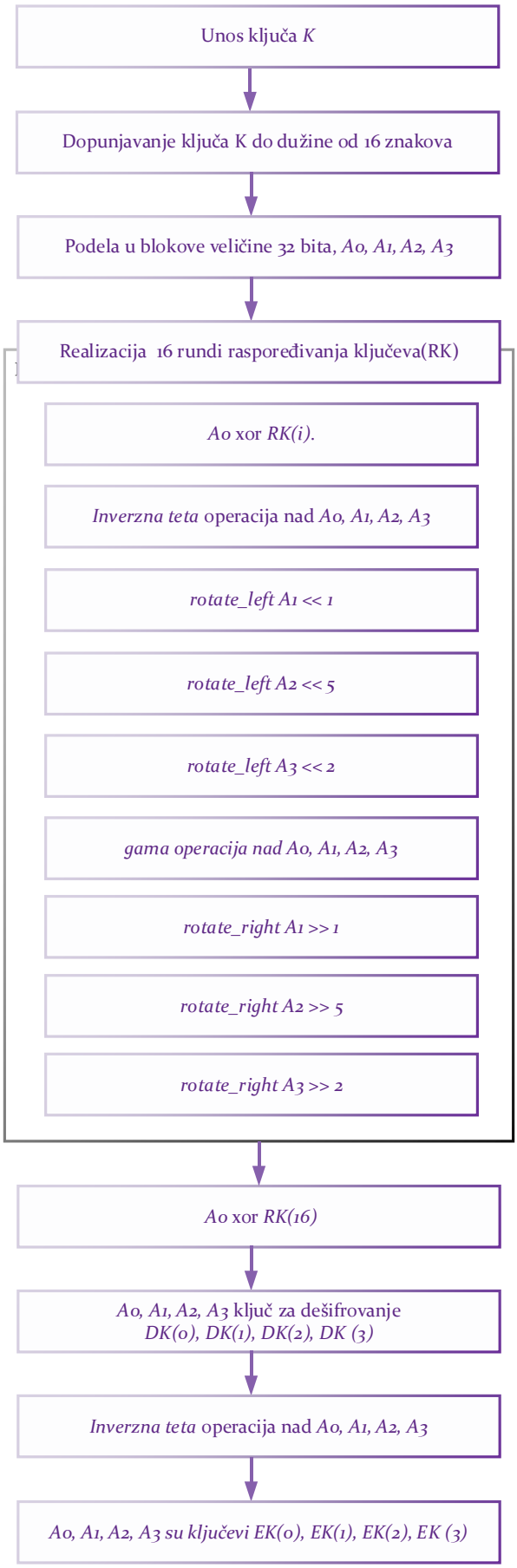
Proceš šifrovanja blokova otvorenog teksta dužine od 128 bita u binarnom zapisu započinje njihovom podelom na 4 reči od po 32 bita. Originalni, ne šifrovani, tekst označavamo sa [a_0, a_1, a_2, a_3], dok ključ označavamo sa [k_0, k_1, k_2, k_3]. Ukoliko imamo slučaj da je blok kraći od 128 bitova, ostatak koji je prazan popunjavamo tako da dodajemo onoliko nula koliko nam nedostaje do punog bloka [74].

NOEKEON šifarski algoritam je prilično jednostavan za razumevanje, i sam proces šifrovanja i dešifrovanja je najjednostavnije prikazati šematski. Proces šifrovanja predstavljen je na slici 4.37. [74].



Slika 4.37. Postupak šifrovanja NOEKEON algoritma

Proces šifrovanja i dešifrovanja algoritma *NOEKEON* odvija se kroz postupak generisanja ili raspoređivanja ključeva; ovi ključevi koriste se u svakoj od rundi kako šifrovanja tako i dešifrovanja. Faze generisanja ključeva *NOEKEON* šifarskog algoritama prikazane su na slici 4.38. [74].



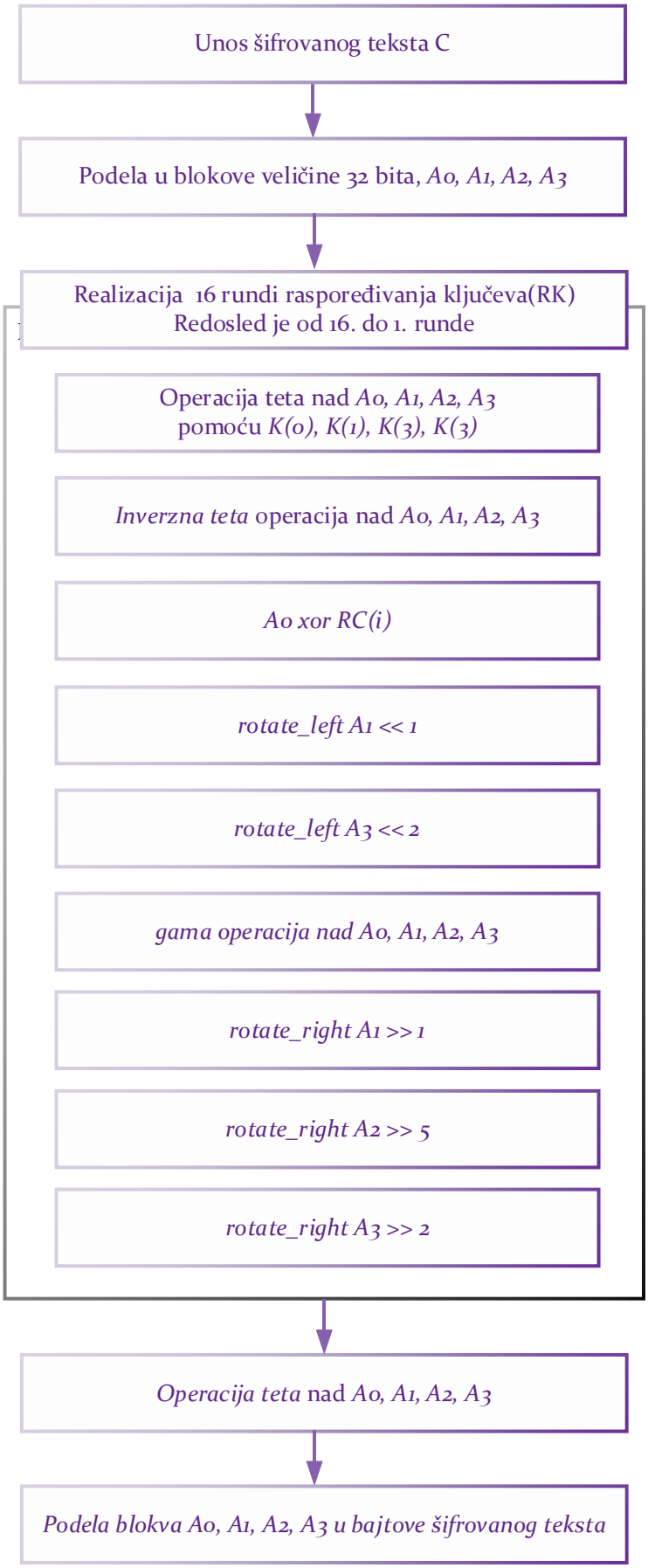
Slika 4.38. Faze generisanja ključeva NOEKEON šifarskog algoritma

Dešifrovanje *NOEKEON* šifarskog algoritma je isto kao i šifrovanje, jedina je razlika u *teta* operaciji. Za šifrovanje se koristi *teta(k,a)*, a za dešifrovanje se koristi *teta (NullVector,a)* [74].

Faze operacije dešifrovanja na *NOEKEON* algoritmu mogu se opisati pomoću šeme predstavljene na slici 4.39. [74].

Slika 4.39. Faze operacije dešifrovanja na NOEKEON algoritmu

Proces dešifrovanja algoritma *NOEKEON* koristi ulazne blokove od po 32 bita. Detaljan opis algoritma za dešifrovanje šematski je prikazan na slici 4.40. [74].



Slika 4.40. Detaljan prikaz dešifrovanja NOEKEON algoritma

4.9.3. PRESENT

PRESENT šifarski algoritam ima dužinu bloka od 64 bita, a podržane su dve dužine ključa od 80 i 128 bitova. Ukoliko je posredi primena lakih šifarskih algoritama, onda je preporučljiva verzija sa ključem dužine 80 bitova. Šifrovanje se obavlja u 32 runde [75].

Na svaku od 31 runde primenjuje se XOR operacije da bi se dobio ključ runde K_i, gde je $1 \leq i \leq 32$. K_{32} se koristi nakon beljenja, kao i za linearnu *bitwise* permutaciju i nelinearni sloj zamene. Nelinearni sloj koristi jednostruku 4-bitnu S-kutiju, koja se primenjuje 16 puta paralelno u svakoj rundi [75]

4.9.3.1. Raspored ključeva

PRESENT šifarski algoritam koristi ključeve dužine od 80 ili 128 bitova. Ako uzmemo za primer da se upotrebljava ključ dužine 80 bitova, on se smešta u registar ključeva *K* i predstavljen je kao $k_{79}, k_{78}, ... k_0$. U prvoj rundi šifrovanja uzima se ključ dužine 64 bita, gde se ključ runde $K_i = k_{63}, k_{62}, ... k_0$, sastoji od 64 krajnja bita sa leve strane trenutnog sadržaja registra *K* koji je prikazan jednačinom (4.3). Tako u rundi *i* imamo da [75]:

$$K_i = k_{63}, k_{62}, ... k_0 = k_{79}, k_{78}, ... k_{16}. \tag{4.3}$$

Nakon izvlačenja ključa runde K_i, registar ključa $k_{79}, k_{78}, ... k_0$ se ažurira kao što je prikazano u jednačini (4.4) [75].

1. $[k_{79}, k_{78}, ... k_1, k_0] = [k_{18}, k_{17}, ... k_{20}, k_{19}]$
2. $[k_{79}, k_{78}, k_{77}, k_{76}] = S[k_{79}, k_{78}, k_{77}, k_{76}]$ (4.4)
3. $[k_{19}, k_{18}, k_{17}, k_{16}, k_{15}] = [k_{19}, k_{18}, k_{17}, k_{16}, k_{15}] \oplus$ broj runde.

Registar ključeva se rotira za 61 mesto ulevo, krajnja četiri bita s leve strane prolaze kroz *PRESENT* S-kutiju, a vrednost broja runde *i* ekskluzivno se podešava bitovima $k_{19}, k_{18}, k_{17}, k_{16}, k_{15}$ od *K* sa najmanje značajnim bitom broja runde na desnoj strani [75].

Ako uzmemo za primer da se upotrebljava ključ dužine 128 bitova, on se smešta u registar ključeva *K* i predstavljen je kao $k_{127}, k_{7126} ... k_0$. U prvoj rundi šifrovanja uzima se ključ dužine 64 bita, gde se ključ runde $K_i = k_{63}, k_{62}, ... k_0$, sastoji od 64 krajnja bita sa leve strane trenutnog sadržaja registra *K* prikazanog jednačinom (4.5). Tako u rundi *i* imamo da [75]:

$$K_i = k_{63}, k_{62}, ... k_0 = k_{127}, k_{126}, ... k_{64}. \tag{4.5}$$

Nakon izvlačenja ključa runde K_i, registar ključa k_{127}, k_{126}, ... k_0 ažurira se kao u jednačini (4.6) [75].

1. $[k_{127}, k_{126} \ldots k_1, k_0] = [k_{66}, k_{65}, \ldots k_{68}, k_{67}]$
2. $[k_{127}, k_{126}, k_{125}, k_{124}] = S[k_{127}, k_{126}, k_{125}, k_{124}]$
3. $[k_{123}, k_{122}, k_{121}, k_{120}] = S[k_{123}, k_{122}, k_{121}, k_{120}]$
4. $[k_{66}, k_{65}, k_{64}, k_{63}, k_{62}] = [k_{66}, k_{65}, k_{64}, k_{63}, k_{62}] \oplus$ broj runde.

(4.6)

Registar ključeva rotira se za 61 mesto ulevo, krajnjih osam bitova sa leve strane prolaze kroz sve *PRESENT* S-kutije, a vrednost broja runde *i* ekskluzivno se podešava bitovima k_{66}, k_{65}, k_{64}, k_{63}, k_{62} od *K* sa najmanje značajnim bitom broja runde na desnoj strani [75].

4.9.3.2. Sloj permutacije

Da bi postavili sloj za permutacije *PRESENT* šifarskog algoritma, bio je dovoljan minimalni broj elemenata za računanje, pa su se autori odlučili za permutacije na nivou bitova. Izabrana je uobičajena bitna permutacija [75].

4.9.3.3. S-kutija

Koristimo jednu S-kutiju veličine 4x4 bita koja je definisana kao $S: F_2^4 \rightarrow F_2^4$. Glavni izbor za ovu kutiju jeste potreba za niskom hardverskom zahtevnošću. S obzirom na upotrebu permutacije bitova na linearnom difuzionom sloju S, kutija mora da zadovolji kriterijume prikazane u jednačini (4.7) [75]:

$$S_b^W(a) = \sum_{x \in \mathrm{F}_2^4} (-1)^{\langle b, S(x) \rangle + \langle a, x \rangle}. \tag{4.7}$$

1. Za bilo koju fiksnu nenultu ulaznu razliku $\Delta_I \in \mathrm{F}_2^4$ i bilo koju fiksnu nenultu izlaznu razliku $\Delta_o \in \mathrm{F}_2^4$ zahtevamo da bude zadovoljena jednačina (4.8) [75]:

$$\#\{x \in \mathrm{F}_2^4 | S(x) + S(x + \Delta_I) = \Delta_o\} \leq 4. \tag{4.8}$$

2. Za bilo koju fiksnu nenultu ulaznu razliku $\Delta_I \in \mathrm{F}_2^4$ i bilo koju fiksnu nenultu izlaznu razliku $\Delta_0 \in \mathrm{F}_2^4$ tako da $wt(\Delta_I) = wt(\Delta_o) = 1$ važi jednačina (4.9) [75]:

$$\#\{x \in \mathrm{F}_2^4 | S(x) + S(x + \Delta_I) = \Delta_o\} = 0. \tag{4.9}$$

3. Za svaku nenultu $a \in F_2^4$ i svaku nenultu $b \in \mathrm{F}_2^4$ važi da $|S_b^W(a)| \leq 8$ [75].

4. Za svaku $a \in F_2^4$ i svaku nenultu $b \in F_2^4$ tako da $wt(a) = wt(b) = 1$ važi da $S_b^W(a) \pm 4$ [75].

4.9.4. LED

LED koristi 64-, 80-, 96- i 128-bitne ključeve sa 64-bitnim blokovima kroz 32 i 48 rundi [76]. To je šifra nalik AES-u, a autori primenjuju neke novije trendove iz oblasti *lightweight hash* funkcija zasnovanih na blok-šifri. LED algoritam prikazan na slici 4.41. koristi postupak neraspoređivanja ključeva, što je i jedna od glavnih prednosti ovog algoritma. Diferencijalne analize grešaka zasnovane na tehnikama super-S-kutija dobijaju značajna poboljšanja za napade grešaka [77].

$$tekst \rightarrow \overset{SK^0}{\oplus} \rightarrow 4runde \rightarrow \overset{SK^1}{\oplus} \rightarrow 4runde \cdots \overset{SK^{s-1}}{\oplus} \rightarrow 4runde \rightarrow \overset{SK^s}{\oplus} \rightarrow šifrovan\ tekst$$

Slika 4.41. LED koraci šifrovanja

Koncept algoritma zanovan je na matrici 4X4, gde svaki zapis predstavlja element iz GF (engl. *Galois field*) (24) koji ima osnovni polinom prikazan u jednačini (4.10) kojim se množe polja [76]:

$$X^4 + X + 1. \tag{4.10}$$

LED šifarski algoritam zasnovan je na konceptima: *S-boxes* i *MixColumnsSerial* [76].

4.9.4.1. S-kutija

U konceptu S-kutije algoritam više puta koristi S-kutiju, koja je osnov za funkcionisanje i drugih šifarskih algoritama koji nisu preterano zahtevni, a prikazana je u tabeli 4 [76].

Tabela 4. Heksadecimalni sadržaj S-kutije LED algoritma

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S[x]	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

4.9.4.2. MixColumnsSerial

Sloj *MixColumnsSerial* posmatramo kao četiri primene hardverski prilagođene matrice *A*, pri čemu je rezultat ekvivalentan upotrebi MDS [78] matrice *M* [76] koja je prikazana jednačinom (4.11):

$$(\mathbf{A})^{\mathbf{4}} = \begin{vmatrix} 0\,1\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \\ 4\,1\,2\,2 \end{vmatrix}^4 = \begin{vmatrix} 4\,1\,2\,2 \\ 8\,6\,5\,6 \\ B\,E\,A\,9 \\ 2\,2\,F\,B \end{vmatrix} = M \tag{4.11}$$

Da bi se šifrovao tekst dužine 64-bita označen sa *m,* potrebno nam je šesnaest četvorobitnih elemenata koje označavamo od m_0 do m_{15} raspoređenih kao u jednačini (4.12) [76]:

$$\begin{bmatrix} m_0 & m_1 & m_2 & m_3 \\ m_4 & m_5 & m_6 & m_7 \\ m_8 & m_9 & m_{10} & m_{11} \\ m_{12} & m_{13} & m_{14} & m_{15} \end{bmatrix} \tag{4.12}$$

Kako je ova početna vrednost šifrovana, treba voditi računa da se vrednosti učitavaju po redovima. Ključ posmatramo kao elemente od k_0 do k_l, dok je potključ SK_i raspoređen u matricu prikazanu u jednačini (4.13) [76]:

$$\begin{bmatrix} sk_0^i & sk_1^i & sk_2^i & sk_3^i \\ sk_4^i & sk_5^i & sk_6^i & sk_7^i \\ sk_8^i & sk_9^i & sk_{10}^i & sk_{11}^i \\ sk_{12}^i & sk_{13}^i & sk_{14}^i & sk_{15}^i \end{bmatrix} \tag{4.13}$$

Potključ SK_i dobija se podešavanjem preko jednačine (4.14) [76]:

$$SK_{i,j} = k(j+i*16 \bmod l). \tag{4.14}$$

Kada koristimo 64-bitni ključ za šifrovanje označen sa *K*, svi njegovi potključevi su jednaki *K*. Međutim, kod ključa dužine 128 bitova označenim sa *K*, potključevi imaju vrednost *K1* i *K2* koji predstavljaju levi i desni deo ključa *K* [76].

Matrice potključevi za 64-bitni slučaj ključa prikazane su u jednačini (4.15) [76]:

$$\begin{bmatrix} k_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ k_8 & k_9 & k_{10} & k_{11} \\ k_{12} & k_{13} & k_{14} & k_{15} \end{bmatrix} \tag{4.15}$$

Matrice potključevi za 80-bitni slučaj ključa sa dva prva potključa prikazane su u jednačini (4.16) [76]:

$$\begin{bmatrix} k_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ k_8 & k_9 & k & k_{11} \\ k_{12} & k_{13} & k_{14} & k_{15} \end{bmatrix} \begin{bmatrix} k_{16} & k_{17} & k_{18} & k_{19} \\ k_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ k_8 & k_9 & k_{10} & k_{11} \end{bmatrix} \tag{4.16}$$

Matrice potključevi za 128-bitni slučaj ključa prikazane su u jednačini (4.17) [76]:

$$\begin{bmatrix} k_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ k_8 & k_9 & k & k_{11} \\ k_{12} & k_{13} & k_{14} & k_{15} \end{bmatrix} \begin{bmatrix} k_{16} & k_{17} & k_{18} & k_{19} \\ k_{20} & k_{21} & k_{22} & k_{23} \\ k_{24} & k_{25} & k_{26} & k_{27} \\ k_{28} & k_{29} & k_{31} & k_{31} \end{bmatrix} \tag{4.17}$$

addRoundKey( ) – operacija ima dva argumenta – *stanje* i SK^i; za kombinovanje ova dva argumenta koristi ekskluzivno ili (*engl. XOR*)na bitovima [76].

step ( ).– ova operacija ima kao argument *stanje,* i broj koraka u procesu primene ovog šifarskog algoritma zavisi od veličine ključa: 64-bitni ključ ima 8 koraka, dok 128-bitni ključ ima 12koraka. Pored broja koraka za ovu operaciju su nam bitna još četiri stanja algoritma koja se koriste u nizu *AddConstants*, *SubCells*, *ShiftRows* i *MixColumnsSerial* [76].

4.9.5. Piccolo

Piccolo je 64-bitni šifarski algoritam koji ima ključeve dužine 80 i 128 bitova. Šifarski algoritam *Piccolo* sastoji se od dva dela, jedan za obradu podataka i drugi za raspoređivanje ključeva. Razlike u primeni veličine ključeva jesu u broju rundi za deo namenjen obradi podataka, kao i za deo namenjen raspoređivanju ključeva. Prvo dajemo oznake korišćene u ovom radu, a zatim definišemo svaki deo [79], [80], [81].

U daljem tekstu se koriste: $a_{(b)}$, gde *b* označava dužinu bita *a, a*|*b ili* (*a*|*b*) koji označava lanac, $a \leftarrow b$ koji označava ažuriranje vrednosti s *a* za vrednost *b;* $^t\boldsymbol{a}$ predstavlja transponovani vektor ili matricu $\boldsymbol{a}$, dok $\{a\}_b$ predstavlja vrednost broja *a* sa osnovom *b*. Broj rundi, *r*, za *Piccolo*-80 je 25 dok za *Piccolo*-128 ima vrednost 31 i označava se G_{25} i G_{31} [79].

4.9.5.1. Deo obrade podataka

Deo za obradu podataka algoritma *Piccolo* koji se sastoji od *r* rundi, G_r, uzima 64-bitne podatke $X \in \{0,1\}^{64}$, četiri 16-bitna ključa za beljenje $wk_i \in \{0,1\}^{16} (0 \leq i < 4)$ i 2*r* ključa runde dužine 16 bitova $rk_i \in \{0,1\}^{16} (0 \leq i < 2r)$ kao ulaze, a kao izlaze su podaci dužine 64 bita $Y \in \{0,1\}^{64}$. G_r se definiše kao u jednačini (4.18) [79]:

$$G_r: \begin{cases} \{0,1\}^{64} \times \{\{0,1\}^{16}\}^4 \times \{\{0,1\}^{16}\}^{2r} \rightarrow \{0,1\}^{64} \\ (X_{(64)}, wk_{0(16)}, \ldots, wk_{3(16)}, rk_{3(16)}, \ldots, rk_{2r-1(16)}) \rightarrow Y_{(64)} \end{cases} \tag{4.18}$$

gde je *F* – funkcija dužine 16 bitova a *RP* predstavlja permutaciju dužine 64 bita. Funkcija dešifrovanja G_r^{-1} dobija se iz G_r jednostavnom promenom redosleda beljenja i ključeva rundi prikazanih u jednačini (4.19) [79]:

$$G_r^{-1}: \begin{cases} \{0,1\}^{64} \times \{\{0,1\}^{16}\}^4 \times \{\{0,1\}^{16}\}^{2r} \rightarrow \{0,1\}^{64} \\ (Y_{(64)}, wk_{0(16)}, \ldots, wk_{3(16)}, rk_{3(16)}, \ldots, rk_{2r-1(16)}) \rightarrow X_{(64)} \end{cases} \tag{4.19}$$

4.9.5.2. F-funkcija

F-funkcija predstavlja se kao $F: \{0,1\}^{16} \rightarrow \{0,1\}^{16}$; sastoji se od dva sloja S-kutije koji su odvojeni pomoću difuzione matrice. Sloj S-kutije sastoji se od četiri S-kutije koje su dužine 4 bita i zadovoljavaju uslove da budu bijekcije, koja je definisana u tabeli 5, na osnovu koje se podaci dužine 16 bitova $X_{(16)}$ ažuriraju kao u jednačinama (4.20) i (4.21) [79]:

$$(x_{0(4)}, x_{1(4)}, x_{2(4),} x_{3(4)}) \leftarrow (S(x_{0(4)}), S(x_{1(4)}), S(x_{2(4)}), S(x_{3(4)})) \tag{4.20}$$

gde je

$$X_{(16)} = x_{0(4)} | x_{1(4)} | x_{2(4)} | x_{3(4)}. \tag{4.21}$$

Tabela 5. S-kutija u heksadecimalnoj formi algoritma Piccolo

x	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
S[x]	e	4	b	2	3	8	0	9	1	a	7	f	6	c	5	d

Difuzija matrice ***M*** definisana je jednačinom (4.22) [79]

$$M = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \tag{4.22}$$

Jednačina (4.23) predstavlja funkciju difuzije na osnovu koje se ažurira 16-bitni podatak $X_{(16)}$ [79]:

$${}^{t}(x_{0(4)}, x_{1(4)}, x_{2(4),} x_{3(4)}) \leftarrow \boldsymbol{M} \cdot {}^{t}(x_{0(4)}, x_{1(4)}, x_{2(4),} x_{3(4)}) \tag{4.23}$$

gde se množenje između matrica i vektora vrši preko GF(2^4) definisanog nesvodivog polinoma: $x^4 + x + 1$ [79].

4.9.5.3. Permutacija runde

Permutacija runde: $\{0, 1\}^{64} \rightarrow \{0, 1\}^{64}$ deli 64-bitni ulaz $X_{(64)}$ na osam podataka dužine 8 bitova $X_{(64)} = x_{0(8)} | x_{1(8)} | x_{2(8)} | x_{3(8)} | x_{4(8)} | x_{5(8)} | x_{6(8)} | x_{7(8)}$, a zatim ih permutuje kao u jednačini (4.24) i formira $Y_{(64)}$ [79]:

$$\begin{aligned} \mathrm{RP}: & \left(x_{0(8),} x_{1(8)}, x_{2(8)}, x_{3(8)}, x_{4(8)}, x_{5(8)}, x_{6(8)}, x_{7(8)}\right) \\ & \leftarrow \left(x_{2(8),} x_{7(8)}, x_{4(8)}, x_{1(8)}, x_{6(8)}, x_{3(8)}, x_{0(8)}, x_{57(8)}\right). \end{aligned} \tag{4.24}$$

4.9.5.4. Deo raspoređivanja ključeva

Kako algoritam može da radi sa ključevima dužine od 80 i 128 bitova, postoje dva načina za raspoređivanje ključeva. Ključ za beljenje definisan je kao $wk_{i(16)}(0 \leq i < 4))$ i ključevi runde $rk_{j(16)}(0 \leq j < 2r)$. Funkcije za raspoređivanje ključeva u zavisnosti od dužine ključa označene su KS_r^{80} i KS_r^{128} [79]

Vrednosti konstanti con_i^{80} i con_i^{128} u zavisnosti od dužine ključa definisane su jednačinama koje slede, u kojima je c_i predstavljen sa pet bitova kao u jednačini (4.25) [79].

$$(con_{2i}^{80}|con_{2i+1}^{80}) \leftarrow (c_{i+1}|c_0|c_{i+1}|\{00\}_2|c_{i+1}|c_0|c_{i+1}) \oplus \{0f1e2d3c\}_{16}$$
$$(con_{2i}^{128}|con_{2i+1}^{128}) \leftarrow (c_{i+1}|c_0|c_{i+1}|\{00\}_2|c_{i+1}|c_0|c_{i+1}) \oplus \{6547a98b\}_{16}. \quad (4.25)$$

Raspoređivanje ključeva za KS_r^{80} deli ključ dužine 80 bitova na pet ključeva dužine 16 bitova, definisanih kao $k_{i(16)}$ ($0 \leq i < 5$) i ključ za beljenje $wk_{i(16)}$($0 \leq i < 4$) i ključ runde $rk_{j(16)}$ ($0 \leq j < 2r$) prikazanih u jednačini (4.26), gde treba znati da su k_i^L i k_i^R leva (L) i desna (R)v polovina k_i [79]:

$$wk_0 \leftarrow k_0^L|k_1^R,\ wk_1 \leftarrow k_1^L|k_0^R,\ wk_2 \leftarrow k_4^L|k_3^R,\ wk_3 \leftarrow k_3^L|k_4^R\ . \quad (4.26)$$

Dok svako *r* moramo da definišemo kao u jednačini (4.27):

$$(rk_{2i}, rk_{2i+1}) \leftarrow (con_{2i}^{80}, \quad con_{2i+1}^{80}) \oplus \begin{cases} (k_2, k_3)\ ako\ je\ \boldsymbol{i}\ mod\ \mathbf{5} = \mathbf{0}\ ili\ \mathbf{2} \\ (k_0, k_1)\ ako\ je\ \boldsymbol{i}\ mod\ \mathbf{5} = \mathbf{1}\ ili\ \mathbf{4} \\ (k_4, k_4)\ ako\ je\ \boldsymbol{i}\ mod\ \mathbf{5} = \mathbf{3} \end{cases} \quad (4.27)$$

Raspoređivanje ključeva za KS_r^{128} deli ključ dužine 128 bitova na osam ključeva dužine 16 bitova, definisanih kao $k_{i(16)}$ ($0 \leq i < 8$) i ključ za beljenje $wk_{i(16)}$($0 \leq i < 4$) i ključ runde rk_j (16) ($0 \leq j < 2r$) kao u jednačini (4.28), gde treba znati da su k_i^L i k_i^R leva (L) i desna (R) polovina k_i [79]

$$wk_0 \leftarrow k_0^L|k_1^R, \quad wk_1 \leftarrow k_1^L|k_0^R, \quad wk_2 \leftarrow k_4^L|k_7^R, \quad wk_3 \leftarrow k_7^L|k_4^R \quad (4.28)$$

Dok za svako *r* vrednosti od *2r-1* moramo da proverimo da li je ostatak pri deljenju sa 0 jednak nuli, kao u jednačini (4.29) [79]:

$$(k_0, k_1, k_2, k_3, k_4, k_5, k_6, k_7) \leftarrow (k_2, k_1, k_6, k_7, k_0, k_3, k_4, k_5). \quad (4.29)$$

U suprotnom važi jednačina (4.30):

$$rk_i \leftarrow k_{(i+2)}\ mod\ 8\ \oplus con_i^{128}\ . \tag{4.30}$$

4.9.6. TWINE

TWINE je 64-bitni šifarski algoritam koji postoji u varijanti *TWINE*-80 sa dužinom ključa od 80 bita i *TWINE*-128 sa dužinom ključa 128 bitova. Nastao je sa zadatkom da ispuni sledeće karakteristike: da ima mali otisak u implementaciji hardvera ispod 2.000 GE [23, 33], malu potrošnju memorije za implementaciju softvera, i, što je vrlo važno, isti proces šifrovanja i dešifrovanja [82].

Otvoren tekst koji treba da se šifruje dužine 64 bita označavamo sa *P*, ključ svake runde je označen sa *RK,* dok je šifrovan tekst označen sa *C*. Ključ runde $RK_{(32 \times 36)}$ nastaje raspoređivanjem ključeva od tajnog ključa, $K_{(n)}$. Jednu rundu algoritma čine nelinearni i difuzni sloj. Nelinearni sloj koristi 4-bitne S-kutije, dok difuzni sloj pravi permutacije 16 blokova i prema rezultatima iz literature[41] dizajniran je da obezbedi bolju difuziju od kružnog pomeraja. Ova funkcija se ponavlja 36 puta kroz runde, gde broj ponavljana ne zavisi od dužine ključa. Treba napomenuti da se u poslednjoj rundi izostavlja difuzni sloj [82].

S-kutija predstavlja 4-bitnu permutaciju definisanu prema tebeli 6. Permutacija indeksa blokova, π: {0, … , 15} → {0, …, 15}, gde je *j*-ti podblok (za *j* = 0, ..., 15) preslikan u π[*j*]-ti podblok, prikazana je u tabeli 7 [82].

Tabela 6. S-kutija sa vrednostima permutacija TWINE algoritma

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	C	0	F	A	2	B	9	5	8	3	D	7	1	E	6	4

Tabela 7. π i π-1 vrednosti permutacija TWINE algoritma

j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\pi[j]$	5	0	1	4	7	12	3	8	13	6	9	2	15	10	11	14
$\pi^{-1}[j]$	1	2	11	6	3	0	9	4	7	10	13	14	5	8	15	12

Za proces dešifrovanja koristi se raspored kutija i ključeva koji je isti kao i raspored koji se koristi za šifrovanje. Jedina razlika između šifrovanja i dešifrovanja jeste obrnuta permutacija indeks bloka [82].

TWINE koristi jednu S-kutiju umesto više njih, čime doprinosi da su hardverska i softverska implementacija manje zahtevne. *TWINE* je šifarski algoritam koji ne koristi permutaciju bita u svom rasporedu ključeva. *S-kutija* dužine 4 bita definisana je jednačinom (4.31) [82]:

$$y = S(k) = f((x \oplus b) - 1) \tag{4.31}$$

gde a^{-1} označava inverznu vrednost a sa polinomom $z^4 + z + 1$, a $b = 1$ je konstanta, a $f(\cdot)$ je funkcija koja čuva paralelne odnose definisana jednačinom (4.32) [82]:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{4.32}$$

za $y = f(x)$ sa $y = (y_0 \parallel y_1 \parallel y_2 \parallel y_3)$ i $x = (x_0 \parallel x_1 \parallel x_2 \parallel x_3)$.

Raspored ključeva ovog šifarskog algoritma omogućava nesmetano izvođenje operacija bez prekidanja programa i kreiranje ključa runde bez upotrebe posrednog ključa [82].

4.9.7. KATAN i KTANTAN

4.9.7.1. KATAN

Šifarski algoritam *KATAN* možemo da sretnemo u tri verzije: *KATAN32*, *KATAN48* i *KATAN64*. Sve varijacije šifarskih algoritama koji su zasnovani na *KATAN*-u imaju ključeve dužine 80 bitova i šifrovanje se obavlja u 254 runde. Pored ovog za sve šifarske algoritme porodice *KATAN* zajednička je upotreba istih nelinearnih funkcija [83].

KATAN32 ima originalni i šifrovani tekst koji je dužine 32 bita. Proces šifrovanja se bazira na tome što se originalni tekst učitava u dva registra odgovarajućih dužina. Registar L_1 je dužine od 13 bitova, dok je registar L_2 dužine od 19 bitova. Učitavanje se vrši tako što se bit otvorenog teksta koji ima najmanju težinu učitava na nultu poziciju registra L_2, dok se bit najveće težine otvorenog teksta smešta na poziciju dvanaestog bita registra L_1. Ovde treba voditi računa da pozicije bitova počinju od nula. Šifrovanje počinje tako što se oba registra u prvoj rundi pomeraju za jedno mesto ulevo, pa tako *i*-ti bit dobija poziciju od *i+1*, a upražnjena pozicija L_i popunjava se novim izračunatim bitom. Ovaj postupak važi za oba registra. Šifrovani tekst dobija se posle 254 runde ovog postupka. Nakon procesa

šifrovanja na nultoj poziciji registra L_2 smešten je bit šifrovanog teksta sa najmanjom težinom [83].

U svakoj od rundi šifrovanja *KATAN32* koristi dve nelinearne funkcije $f_a(\cdot)$ i $f_b(\cdot)$. Nelinearne funkcije f_a i f_b definisane su jednačinom (4.33) [83]:

$$\begin{aligned} f_a(L_1) &= L_1[x_1] \oplus L_1[x_2] \oplus (L_1[x_3] \cdot L_1[x4]) \oplus (L_1[x_5] \cdot IR) \oplus k_a \\ f_b(L_2) &= L_2[y_1] \oplus L_2[y_2] \oplus (L_2[y_3] \cdot L_2[y_4]) \oplus (L_2[y_5] \cdot L_2[y_6]) \oplus k_b \end{aligned} \tag{4.33}$$

gde je IR pravilo neregularnog ažuriranja po kome se obavlja XOR nad $L_1[x_5]$ u onim rundama kada se koristi njihovo regularno ažuriranje,dok su k_a i k_b dva bita potključa. Za i-tu rundu k_a definisano je kao k_{2i}, dok k_b dobija vrednost k_{2i+1}. Izbori bitova $\{x_i\}$ i $\{y_j\}$ definisani su za svaku varijantu nezavisno i navedeni su u tabeli 8 [83].

Registri L_1 i L_2 se pomeraju posle postupka izračunavanja nelinearnih funkcija, bit najveće težine se odbacuje, a bit najmanje težine se menja izlazom druge nelinearne funkcije. Bit najmanje težine nelinearne funkcije L_1 je izlaz f_b, dok je bit najmanje težine funkcije L_2 izlaz od f_a [83].

KATAN šifarski algoritam ključ za šifrovanje dužine 80 bitova očitava po principu gde se bit najmanje težine ključa učitava na nultu poziciju LFSR. Tako očitavan ključ kreira potključeve koji se sastoje od pozicija 0 i 1 LFSR. Na taj način svaka runda dobija svoj potključ koji je predstavljen sa k_{2i} i k_{2i+1}, a LFSR se dva puta beleži. Za polinom povratne sprege odabran je primitivni polinom sa minimum 5 elemenata [83].

Kako smo detaljno opisali proces šifrovanja pomoću *KATAN32* šifarskog algoritma, prikazaćemo i ostale šifarske algoritme koji su zasnovani na *KATAN* principu. Varijante *KATAN48* i *KATAN64*, pored očigledne razlike u dužini originalnog i šifrovanog teksta, razlikuju se i u dužini registara koji se koriste za šifrovanje, po ulazima nelinearnih funkcija, kao i po broju nelinearnih fuknkcija koje se koriste za šifrovanje [83].

Tabela 8. prikazuje nam uporedne vredosti parametara koji predstavljaju dužinu matrice i položaja bitova koji su ulazi za nelinearne funkcije što se koriste za šifrovanje. Bitovi X_i i Y_j razlikuju se u zavisnosti od dužine teksta koji se šifruje [83].

Tabela 8. Uporedne vrednosti parametara za šifrovanje KATAN algoritma

	KATAN32/ KTANTAN32	*KATAN48/ KTANTAN48*	*KATAN64/ KTANTAN64*
$\|L_1\|$	13	19	25
$\|L_2\|$	19	29	39
x_1	12	18	24
x_2	7	12	15
x_3	8	15	20
x_4	5	7	11
x_5	3	6	9
y_1	18	28	38
y_2	7	19	25
y_3	12	21	33
y_4	10	13	21
y_5	8	15	14
y_6	3	6	9

Kod modifikacije *KATAN32* u šifarski algoritam sa dužinom teksta od 48 bitova, tj. *KATAN*48 primenjuje funkcije f_a i f_b dva puta u svakoj od rundi šifrovanja. Par f_a i f_b se primenjuje jednom, pa još jednom sa istim ključevima nad registrom koji je ažuriran posle prvog izvršavanja funkcija. *KATAN64* u svakoj runda primenjuje f_a i f_b sa istim ključevima čak tri puta [83].

Ceo proces šifrovanja završava se posle ukupno 254 ciklusa. Na početku ovog procesa *Linear Feedback Shift Register* (LFSR) se inicijalizuje na sve jedinice. Ciklusi šifrovanja broje se preko LFSR koji beleži upotrebu polinoma povratne sprege koji je predstavljen jednačinom (4.34) [83]:

$$x_8 + x_7 + x_5 + x_3 + 1. \tag{4.34}$$

U literaturi je za implementaciju korišćen *Synopsis Design Compiler* i prikazana je implementacija za svaku od verzija *KATAN* šifarskih algoritama koju smo predstavili u tebeli 9 [83].

Tabela 9. Uporedni zahtevi za implementaciju KATAN algoritma

Šifarski algoritam	*KATAN32*	*KATAN48*	*KATAN64*
Ukupna zahtevnost	802 GE	927 GE	1054 GE
Zahtevnost za sekvencijalnu logiku	742 GE	842 GE	935 GE
Zahtevnost za kombinacionu logiku	60 GE	85 GE	119 GE
Potrošnja energije na 100 KHz, i propusna moć od 12,5 Kbps	381 nW	439 nW	555 nW

Uz relativno malo ulaganja u hardver, protok *KATAN* šifarskih algoritama može da se poveća dva ili tri puta u zavisnosti od konkretnog algoritma. Za povećanje brzine šifrovanja potrebno je udvostručiti ili utrostručiti logiku za nelinearne funkcije *fa* i *fb*, kao i logiku za povratne koeficijente brojača i registra ključa [83].

4.9.7.2. KTANTAN

Šifarski algoritmi *KTANTAN* zasnovani su na *KATAN* algoritmima i jedina razlika je u rasporedu ključeva. *KATANTAN* šifarski algoritmi koriste ključ koji je fiksiran, dok se varijacije prave u odnosu na odabir potključeva [83].

KTANTAN tretira ključ kao 5 reči od po 16 bitova, iz koje bira isti bit pomoću MUX16to1, na osnovu četiri najznačajnija bita runde koji kontrolišu LFSR. A posle od pet bitova biramo jedan na osnovu četiri bita najmanje težine iz LFSR [83].

Ako uzmemo da je $K = w_4||w_3||w_2||w_1||w_0$, w_0 je bit najmanje težine od *K,* dok je *w4* bit najveće težine od *K*. T_7 predstavlja najmanje značajan bit brojača runde LFSR. a_i = MUX16to1(w_i, $T_7T_6T_5T_4$), gde MUX16to1(*x, y*) vraća *y*-ti bit od *x* [83]. Kao ključni koriste se bitovi prikazani u jednačini (4.35):

$$k_a = T_3 \cdot T_2 \cdot (a_0) \oplus (T_3 \vee T_2) \cdot \text{MUX4to1}(a_4a_3a_2a_1, T_1T_0)$$
$$k_b = T_3 \cdot T_2 \cdot (a_4) \oplus (T_3 \vee T_2) \cdot \text{MUX4to1}(a3a2a1a_0, T_1T_0). \quad (4.35)$$

Kada se uzima u obzir *ka* ili *kb*, koji imaju 80-bitni ključ, njihov jedan bit se koristi samo dva puta, 15 bitova se koriste četiri puta, dok se 64 bita koriste 3 puta. Još jedno ograničenje je da se svaki pojedinačni bit ključa koristi najmanje 5 puta [83].

Zahtevnost implementacije *KTANTAN* šifarskog algoritma data je u tabeli 10.

Tabela 10. Uporedni zahtevi za implementaciju algoritma KTANTAN

Šifarski algoritam	*KTANTAN* 32	*KTANTAN* 48	*KTANTAN* 64
Ukupna zahtevnost	462 GE	588 GE	688 GE
Zahtevnost za sekvencijalnu logiku	244 GE	344 GE	444 GE
Zahtevnost za kombinacionu logiku	218 GE	244 GE	244 GE
Potrošnja energije na 100 KHz, i propusna moć od 12,5 Kbps	146 nW	234 nW	292 nW

4.9.8. PRINCE

PRINCE šifarski algoritam omogućava veći izbor S-kutija, što dovodi do nižih troškova implementacije. S-kutija i njen inverzni deo uključeni u funkciju šifrovanja ne utiču na cenu implementacije. Za smanjenje troškova dešifrovanja u odnosu na šifrovanje koristi se samo jedan množilac na samom početku kola [84].

PRINCE šifarski algoritam je 64-bitni algoritam koji ima ključ dužine od 128 bitova. [84] Svaka runda $PRINCE_{core}$ prikazana je šematski na slici 4.42.

Slika 4.42. Prikaz rundi PRINCEcore

4.9.8.1. Sloj dodavanja ključa

Ključ *k* dužine od 128 bitova deli se na dva potključa k_0 i k_1 od po 64 bita, a zatim se produžuje do dužine ključa od 192 bita, koji se predstavlja jednačinom (4.36) [84]:

$$(k_0||k_1) \rightarrow (k_0||k_0'||k_1) := (k_0||k_0 >>> 1) \oplus (k_0 >> 63||k_1). \quad (4.36)$$

Prva dva potključa k_0 i k_0' koriste se kao ključevi za beljenje. Ključ k_1 dužine 64 bita jeste ključ koji se koristi za šifrovanje. Šifrovanje se obavlja u 12 rundi u postupku koji se naziva $PRINCE_{core}$ koji je prikazan na slici 4.43. [84].

$$m \rightarrow \overset{k_0}{\oplus} \rightarrow PRINCE_{core} \rightarrow \overset{k_0'}{\oplus} \rightarrow c$$

Slika 4.43. Proces PRINCE core šifrovanja

4.9.8.2. Sloj S-kutije

Izbor S-kutija, koja minimizira troškove, presudan je za postizanje rezultata koji mogu da se porede sa ostalim algoritmima. Da bi se obezbedio željeni stepen sigurnosti S-kutija *S:* $F^4_2 \rightarrow F^4_2$, mora da je maksimalna verovatnoća diferencijala jednaka ¼, da postoji tačno 15 diferencijala sa verovatnoćom ¼, da je maksimalna apsolutna pristrasnost linearne aproksimacije jednaka ¼, da postoji tačno 30 linearnih aproksimacija sa apsolutnom pristrasnošću ¼ i da svaka od 15 nenultih komponenata funkcija ima algebarski stepen 3. S-kutija koju koristi data je u tabeli 11 [84].

Tabela 11. S-kutija Prince šifarskog algoritma

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S[x]	B	F	3	2	A	C	9	1	6	7	8	0	E	5	D	4

4.9.8.3. Linearni sloj

Linearni sloj sastoji se od dva različita sloja, jednog koji koristi matricu *M* i drugog koji koristi matricu *M'*). U oba slučaja se 64-bitno stanje množi sa 64 x 64 matricom u prvom slučaju *M, a u drugom M'*. Matrica *M'* koristi se samo u srednjoj rundi da bi obezbedila α-svojstvo refleksije. Matrica *M* se koristi u funkcijama runde. Da bismo uspeli da obezbedimo potpunu difuziju, nakon dve runde kombinujemo *M'*-mapiranje sa primenom matrice *SR* (*shift rows)* koja permutuje 16 elemenata prema tebeli 12 [84].

Tabela 12. Tabela permutacija linearnog sloja PRINCE algoritma

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	5	10	15	4	9	14	3	8	13	2	7	12	1	6	11

Na osnovu toga dobijamo jednačinu (4.37):

$$M = SR \circ M'. \tag{4.37}$$

Da bi troškovi implementacije bili minimalni, broj jedinica u matricama *M'* i *M* trebalo bi da bude što manji, dok istovremeno treba postići da je najmanje 16 S-kutija aktivno u 4 uzastopne runde. Svaki izlazni bit S-kutije mora da utiče na 3 S-kutije u sledećoj rundi i za to su potrebne četiri 4 x 4 matrice kao gradivni blokovi za *M'* – sloj koji ima minimalno tri jedinice. Tako možemo koristiti matrice prikazane jednačinom (4.38) [84].

$$M_o = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};\ M_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$M_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};\ M_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.38)$$

Korak koji sledi namenjen je za generisanje M̂ matrice, veličine 4x4, gde su redovi i kolona permutacija matrica M_0, M_1, M_2, M_3. Permutacije redova biraju se tako da se dobije simetrična blok-matrica. Izbor gradivnih blokova i simetrična struktura omogućavaju nam da dobijena matrica 16 x 16 prikazaje u jednačini (4.39) [84]:

$$\widehat{M}^{(0)} = \begin{bmatrix} M_0 & M_1 & M_2 & M_3 \\ M_1 & M_2 & M_3 & M_0 \\ M_2 & M_3 & M_0 & M_1 \\ M_3 & M_0 & M_1 & M_2 \end{bmatrix}\ \widehat{M}^{(1)} = \begin{bmatrix} M_1 & M_2 & M_3 & M_0 \\ M_2 & M_3 & M_0 & M_1 \\ M_3 & M_0 & M_1 & M_2 \\ M_0 & M_1 & M_2 & M_3 \end{bmatrix} \quad (4.39)$$

Da bismo dobili permutaciju za potpuno 64-bitno stanje, konstruišemo 64 x 64 blok-dijagonalnu matricu *M'* sa ($\hat{M}^{(0)}$; $\hat{M}^{(1)}$; $\hat{M}^{(1)}$; $\hat{M}^{(0)}$) kao dijagonalne blokove [84].

4.9.8.4. Sloj dodavanja konstante runde

Konstante rundi dužine 64 bita u heksadecimalnom obliku prikazane su u tabeli 13 [84].

Tabela 13. Konstante rundi PRINCE algoritma

RC_0	0000000000000000
RC_1	13198a2e03707344
RC_2	a4093822299f31d0
RC_3	082efa98ec4e6c89
RC_4	452821e638d01377
RC_5	be5466cf34e90c6c
RC_6	7ef84f78fd955cb1
RC_7	85840851f1ac43aa
RC_8	c882d32f25323c54
RC_9	64a51195e0e3610d
RC_{10}	d3b5a399ca0c2399
RC_{11}	c0ac29b7c97c50dd

Moramo znati da je za sve $0 \le i \le 11$, $RC_i \oplus RC_{11-i}$ konstanta $\alpha = c0ac29b7c97c50dd$, $RC_0 = 0$ i da su RC_1, RC_2 RC_3, RC_4, i α izvedeni iz razlomljenog dela π [84].

Iz činjenice da konstante runde zadovoljavaju $RC_i \oplus RC_{11-i} = \alpha$ i da je *M'* involucija, zaključeno je da je inverzna operacija $PRINCE_{core}$ sa parametrom *k* jednaka operaciji $PRINCE_{core}$ sa parametrima ($k \oplus \alpha$). Ovo svojstvo $PRINCE_{core}$ nazivamo *α-svojstvom refleksije* (engl. α-*reflection property*) [84].

4.9.9. SIMON

SIMON šifarski algoritam podržava pet različitih veličina blokova, kao i tri različita ključa, tako da ima ukupno deset različitih kombinacija. Algoritam je napravljen da bude malo hardverski zahtevan ali i jednostavan za implementaciju, kao i da zadrži performanse na željenom nivou. U tabeli 14. navedene su različite veličine blokova i ključeva *SIMON* šifarskog algoritma [85].

Tabela 14. Veličina bloka i ključeva SIMON algoritma

Veličina bloka	Veličina ključa
32	64
48	72, 96
64	96, 128
96	96, 144
128	128, 192, 256

SIMON šifarski algoritam koji ima *n*-bitnu reč ima 2*n*-bitni blok i označavamo ga sa $SIMON_{2n}$, gde *n* može da ima vrednosti 16, 24, 32, 48 ili 64. $SIMON_{2n}$ sa *m*-rečju (*mn*-bitnim) ključem će se nazivati $SIMON_{2n/mn}$. *SIMON*64/128 je vezija sa 64-bitnim blokovima otvorenog teksta i ključem od 128 bitova [85].

*SIMON*2*n* funkcija runde koja zavisi od ključa je dvostepena *Feistel* mapa R_k prikazana jednačinom (4.40) [85]:

$$R_k(x, y) = (y \oplus f(x) \oplus k, x) \tag{4.40}$$

gde je $f(x) = (S_x \,\&\, S^8{}_x)$ i *k* je ključ runde.

Inverzna vrednost funkcije runde prikazana je jednačinom (4.41) [85]:

$$R_k{}^{-1}(x, y) = (y, x \oplus f(y) \oplus k) \tag{4.41}$$

SIMON šifarski algoritam na osnovu rasporeda ključeva generiše sekvencu od ukupno *T* ključnih reči koje idu od k_0 do k_{T-1}, gde *T* predstavlja broj rundi. Mapa šifrovanja pročitana zdesna nalevo ima oblik od Rk_{T-1} do Rk_0. Efekat funkcije runde Rk_i nad dve reči

šifrom (x_{i+1}, x_i) u *i*-tom koraku ovog procesa prikazan je u tabeli 15. Kod *SIMON* šifarskog algoritma prva i poslednja runda sa stanovišta šifrovanja ne donosi nikakve promene. Njihova uloga je da definišu ključeve [85].

Tabela 15. Parametri šifrovanja SIMON algoritma

Veličina bloka *(2n)*	Veličina ključa *(mn)*	Veličina reči *(n)*	Ključ reči *(m)*	Konstanta sekvence	Broj rundi *(T)*
32	64	16	4	z_0	32
48	72	24	3	z_0	36
	96		4	z_1	36
64	96	32	3	z_2	42
	128		4	z_3	44
96	96	48	2	z_2	52
	144		3	z_3	54
128	128	64	2	z_2	68
	192		3	z_3	69
	265		4	z_4	72

Sve runde šifrovanja pomoću algoritma *SIMON* su iste, što bi značilo i da su operacije simetrične u odnosu na kružna pomeranja *n*-bitnih reči. Ključevi koriste sekvencu 1-bitnih konstanti runde. Razlika između verzija algoritma *SIMON* sa istom veličinom bloka definisana je na osnovu pet sekvenci koje su označene od Z_0 do Z_4. Svaka od ovih sekvenci definisana je pomoću jedne od sledeće 31 sekvence koje su prikazane jednačinom (4.42) [85]:

$$\begin{aligned} U = U_0\, U_1\, U_2 \ldots &= 11111010001001010110000111001 10 \\ V = V_0\, V_1\, V_2 \ldots &= 10001110111110010011000010110 10 \\ W = W_0\, W_1\, W_2 \ldots &= 10000100101100111110001101110 10 \end{aligned} \quad (4.42)$$

Prve dve sekvence su jednostavno $Z_0 = U$ i $Z_1 = V$, dok Z_2, Z_3 i Z_4 i imaju periodu od 62 i formirane su izračunavanjem XOR operacijom nad bitovima 2 sekvence prikazane jednačinom (4.43):

$$T = T_0\, T_1\, T_2 \ldots = 01010101\ldots \quad (4.43)$$

pomoću *U*, *V,* i *W* tako da se ne poremeti redosled, a gde je $(Z_i)_j$ *j*-ti bit od Z_i koji su prikazani jednačinom (4.44) [85].

$$Z_2 = (Z_2)_0(Z_2)_1(Z_2)_2 \ldots = 10101111011100000011010010011 00 \quad (4.44)$$

$Z_3 = (Z_3)_0(Z_3)_1(Z_3)_2... =$ 0101000010001111110010110110011...,
1101101110101100011001011110000
0010010001010011100110100001111...,

$Z_4 = (Z_4)_0(Z_4)_1(Z_4)_2 \ldots =$ 1101000111100110101101100010000
0010111000011001010010011101111...

Sekvence *U*, *V* i *W* mogu se definisati matricama 5x5 koje su prikazane jednačinom (5.45) [85].

$$U = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} V = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} U = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.45)$$

4.10. Poređenje šifarskih algoritama

Posle opisa pomenutih algoritama dajemo prikaz njihovih karakteristika u tabeli 16 [86], koja će nam u daljem radu biti jedan od ulaznih parametara simulacionog modela, a od posebnog nam je značaja kolona pod nazivom energija za šifrovanje bajta koja se u daljem tekstu javlja kao promenljiva *Ec* u jednačini (6.1).

Tabela 16. Uporedne vrednosti lakih šifarskih algoritama

Algoritam	**Veličina bloka /Veličina ključa**	**Vrsta runde**	**Odmotane runde (Unrolled) runde**	**Broj ciklusa**	**Kašnjenje po rundi (ns)**	**Energija za šifrovanje bajta(pJ)**
AES	128/128	SPN	1	11	3.32	21,92
NOEKEON	128/128	SPN	1	18	3.41	21,20
LED 128	64/128	SPN	1	50	5.25	82,08
Present	64/80	SPN	2	17	2.09	19,44
Prince	64/128	SPN	1	13	4.06	18,64
Piccolo	64/80	Feistel	1	26	3.28	22,24
TWINE	64/80	Feistel	2	19	3.10	26,80
Simon64/96	64/96	Feistel	2	22	2.18	26,56
KATAN64	64/80	Shift register	16	17	2.04	17,52

5. Procena optimalne kombinacije protokola i šifarskog algoritma

Procena optimalne kombinacije protokola topologije i šifarskog algoritma je od suštinske važnosti u momentu kada treba da donesemo odluku o tome koji protokol topologije ćemo koristiti i kada treba da odaberemo način implementacije koji ćemo koristiti. Naime, pored odabira protokola topologije od podjednake je važnosti da se razmotri i način na koji se šifarski algoritmi implementiraju u sistem. Da bismo pojednostavili način opredeljivanja za najbolja rešenja, napravili smo matematički model za donošenje odluke, kao i njegovu algoritamsku implementaciju.

5.1. Matematički model

Matematički model [5] za izbor optimalne kombinacije protokola topologije i kriptografskog algoritma je glavni naučni doprinos ove teze. Njegova svrha je da na osnovu parametara pokrivenosti posmatranog područja, količine saobraćaja u BSM i energetske zahtevnosti šifarskog algoritma odredi koeficijente podobnosti i prikaže koje kombinacije protokola topologija i laganih šifarskih algoritama su prihvatljive za implementaciju u datim okolnostima. Još jedna prednost ovog modela je to što je krajnje jednostavan za implementaciju.

Posmatramo slučajnu promenljivu *W* prema jednačini (5.1).

$$W = \frac{X * Y}{Z} \tag{5.1}$$

gde za *X* uzimamo vrednosti za ukupan broj poruka poslatih u celom sistemu, za *Y* uzimamo vrednosti koje su predstavljene u tabeli 16 iz kolone *energija za šifrovanje bajta* za svaki od šifarskih algoritama, a *Z* je koeficijent pokrivenosti posmatranog područja za svaki od protokola. Sve vrednosti iz skupa vrednosti za *X, Y, Z* moraju da imaju normalnu raspodelu. Vrednosti za *W* su vrednosti koje predstavljaju uporedo posmatrane protokole i šifarske algoritme u obliku matrice.

Na osnovu dobijenih vrednosti za *W* (W_1, W_2, W_3, ..., W_n,) izračunavamo ocenu očekivanja, koju označavamo sa $\hat{\mu}$, gde važi jednačina (5.2).

$$\hat{\mu} = \frac{1}{n} * (w_1 + w_2 + \cdots + w_n). \tag{5.2}$$

Interval poverenja za nepoznato očekivanje predstavljamo Studentovom raspodelom koja se na uzorku većem od 30, što je kod nas slučaj, aproksimira nomalnom raspodelom gde nam je S ocena disperzije prikazana u jednačini (5.3).

$$\frac{\hat{\mu}-\mu}{\frac{s}{\sqrt{n}}} \sim \mu(0,1). \tag{5.3}$$

U jednačini (5.4) δ predstavlja standardnu devijaciju

$$\delta = \sqrt{\frac{1}{n-1} * \sum_{i=1}^{n} (w_i - \hat{\mu})^2}. \tag{5.4}$$

Na osnovu pravila tri sigma *(*engl. *three-sigma rule of thumb)* [86] uzimamo interval od 68% i dobijamo rezultate prikazane u jednačinama (5.5) i (5.6).

$$P = \left\{ \left| \frac{\hat{\mu}-\mu}{\frac{s}{\sqrt{n}}} \right| < \varepsilon \right\} = 68\% \tag{5.5}$$

$$\emptyset(\varepsilon) = \frac{0{,}68}{2} + 0{,}5 = 0.84 \Rightarrow \varepsilon = 0{,}995, \tag{5.6}$$

koje važe za interval prikazan jednačinom (5.7)

$$\mu \in \left[\hat{\mu} - \varepsilon * \frac{S}{\sqrt{n}}, \hat{\mu} + \varepsilon * \frac{S}{\sqrt{n}}, \right]. \tag{5.7}$$

Odbacujemo deo intervala poverenja zajedno sa krajem sa desne strane i prihvatljivi su nam rezultati zasnovani na intervalu koji je predstavljen jednačinom (5.8).

$$\left[min, \hat{\mu} - \varepsilon * \frac{S}{\sqrt{n}}, \right]. \tag{5.8}$$

5.2. Algoritamska implementacija matematičkog modela

Algoritam koji sledi predstavlja pseudokôd za implementaciju predloženog matematičkog modela. Ovaj algoritam je implementiran kao nadogradnja simulatora *Atarraya* za BSM koji je korišćen u ovom radu. Algoritam je jednostavan za implementaciju i samim tim može da se koristi kao nadogradnja bilo kog simulacionog okruženja koje ima potrebu da analizira podobnost protokola topologija i šifarskih algoritama u BSM. Algoritam je zbog njegove jednostavnosti moguće implementirati u bilo kom programskom jeziku, čak ga je moguće i koristiti samostalno gde se kao ulazni parametri posmatraju izlazi iz nekog drugog simulacionog okruženja.

Algorithm 1: Energy estimate model alghoritham

Input: Array: X, Y, Z.
Output: The largest element in the set

```
normality-check(Array)
foreach element-of-array do
    if ks-normality-test=true then
        calculate p-value;
        calculate ks-statistc-value;
        return true;
    return false
sumW ← 0
for i ← 1 to n do
    for j ← 1 to m do
        if (normality-check(X[i]) and normality-check(Y[j]) and
         normality-check(Z[j])) = true then
            W[i, j] ← X[i] ∗ Z[i]/Y [j];
            sumW ← sumW + W[i, j];
        return Wrong sample.
n ← i ∗ j;
nikapa ← sumW/n;
sumPW ← 0;
for i ← 1 to n do
    for j ← 1 to m do
        sumPW ← sumPW + (W[i, j] − nikapa)^2;
if (epsilono> |(nikapa − ni)/(S/√n)| then
    P ← 0.68;
    Fiofepsilon ← (P/2) + 0.5;
    epsilono ← 0, 995;
cutoff ← nikapa − (epsilono ∗ S/√n);
for i ← 1 to n do
    for j ← 1 to m do
        if W[i, j]<cutoff then
            return Acceptable.
        return Not acceptable.
```

6. Simulacija i studija slučaja

U ovom poglavlju predstavljen je simulacioni scenario na osnovu koga je u *Atarraya* simulatoru potvrđena ispravnost predstavljenog matematičkog modela. Simulirana je količina saobraćaja različitih protokola topologije u BSM. Pored toga izračunata je pokrivenost posmatranog područja svakog protokola. Svaki od protokola analiziran je prema istim parametrima simulacije da bismo na adekvatan način mogli da poredimo rezultate simulacije. Simulacija i studija slučaja potvrdili su ispravnost predstavljenog matematičkog modela.

6.1. Simulaciono okruženje

Za postavku simulacije koristili smo *Atarraya* simulator [88] kao pogodan alat za potrebe postavke ovog simulacionog okruženja. Ovaj simulator pokazao se kao najbolja polazna osnova za generisanje topologije posmatranog područja, broja poruka i veličine samih poruka, koja zavisi od broja okolnih čvorova koje posmatrani čvor koriste kao posrednika za prosleđivanje poruka.

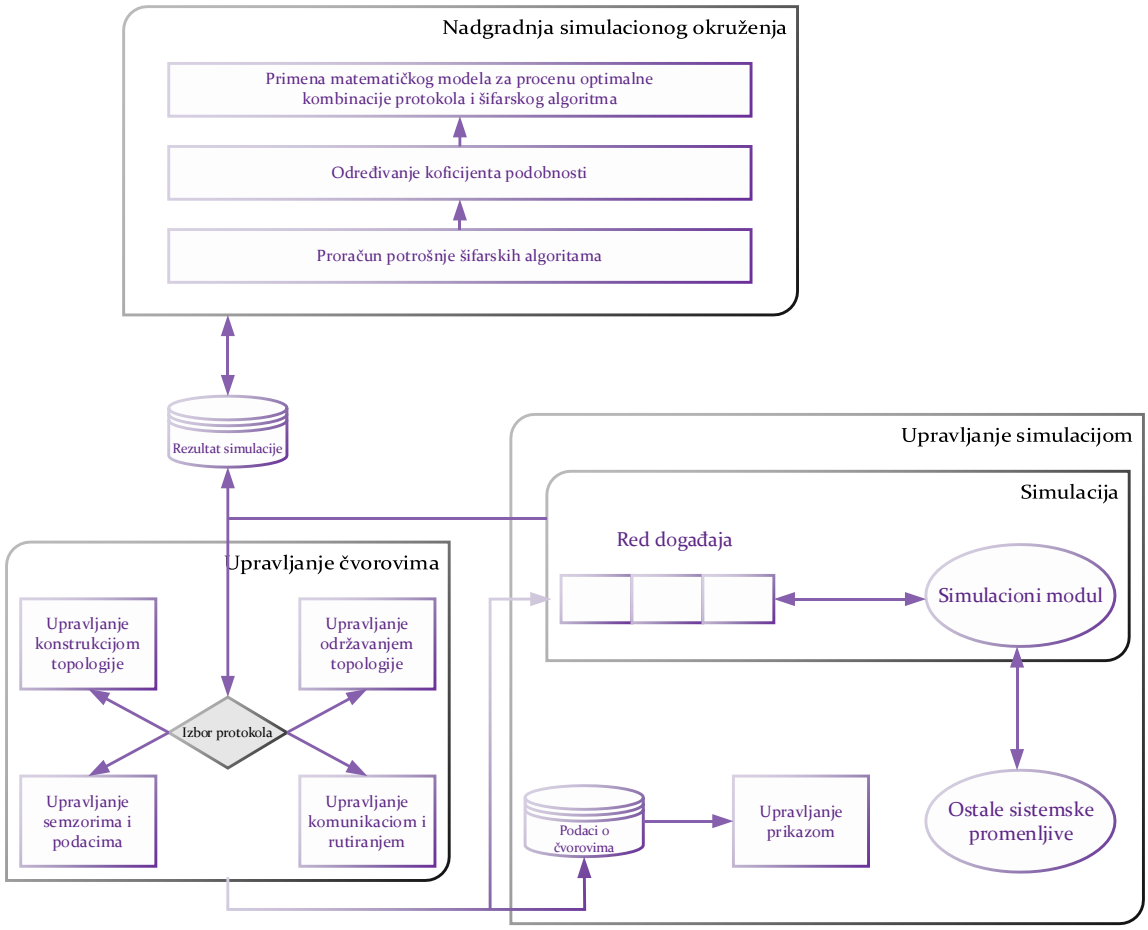
Atarraya je simulator otvorenog kôda napisan kao Java projekat i iz tog razloga bio je pogodan za nadogradnju i personalizaciju, jer se u svojoj osnovi ne bavi algoritmima šifrovanja kao ni potrošnjom energije za šifrovanje [89]. U tabeli 16 prikazana je potrošnja različitih šifarskih algoritama, na osnovu koje možemo da unapredimo simulaciono okruženje i načinimo komparativnu analizu različitih protokola za generisanje topologije i odabir glave klastera sa aspekta energetske efikasnosti implementacije šifarskih algoritama.

Još jedan bitan faktor koji je razmatran u ovom radu jeste komunikaciona pokrivenost posmatranog područja. Ona zavisi od izbora protokola topologije, a do sada nije bila razmatrana u dostupnoj literaturi.

6.1.1. Arhitektura simulatora

U ovom poglavlju prikazana je osnovna struktura simulatora, kao i njegova nadgradnja za potrebe našeg istraživanja. Na slici 6.1. prikazane su glavne komponente *Atarraya* simulatora koje su podeljenje u dva osnovna dela: deo za upravljanje čvorovima i deo za upravljanje simulacijom. Pored funkcionalnosti koju obezbeđuje sam simulator (upravljanje simulacijom, upravljanje čvorovima), na rezultatima simulacije napravljena

je nadgradnja, čija se struktura i algoritamska implementacija zasnivaju na matematičkom modelu koji je prikazan u poglavlju 5 ove disertacije.



Slika 6.1. Šematski prikaz komponenata simulatora i njegove nadgradnje

6.1.1.1. Upravljanje simulacijom

Klasa za upravljanje simulacijom ima zadatak raspoređivanje događaja iz reda događaja, njihovo prosleđivanje na obradu u model za upravljanje čvorovima. Klasa se kreira pomoću metode *StartSimulation ().* Ova klasa sadrži red događaja, simulacioni časovnik, bazu podataka sa podacima o čvorovima i simulacioni modul. Ova klasa se izvršava dok se ne ispuni jedan od tri uslova: da u redu nema više događaja, da svi čvorovi dostignu konačno stanje ili da je protokol za održavanje topologije utvrdio da glava klastera nema više aktivnih susednih čvorova, pa je mreža ugašena). Ako se prvi uslov ispuni, a simulator nije obavešten da su protokoli izvršili događaje, to znači da je došlo do greške tokom simulacije i biće obavešten o tome u izveštaju o simulaciji [89].

Prvo što nit u petlji uradi jeste da verifikuje da li je događaj validan. Ako je to slučaj, događaj će biti registrovan (ako je korisnik odabrao ovu opciju), simulacioni časovnik će

se ažurirati i događaj će biti poslat rukovaocu čvora. Tamo će se događaj dostaviti odgovarajućem rukovaocu događaja (*event handler*) u skladu sa odgovarajućim protokolom kojem događaj pripada. Jednom kada se događaj izvrši, simulator će se vratiti u petlju i ponovo pokrenuti proces. Simulator ažurira sat sa vremenom izvršenja događaja na osnovu činjenice da su svi događaji u redu sortirani prema predviđenom vremenu izvršenja, tako da ne postoji nešto poput putovanja u prošlost [89].

6.1.1.2. Upravljanje protokolima

Upravljanje protokolima definisano je protokolom koji će se koristiti u simulaciji. U ovoj klasi za svaki algoritam za formiranje topologije definisano je više protokola: protokoli za konstrukciju topologije, održavanje topologije, upravljanje podacima senzora i protokoli za usmeravanje komunikacija. Po otpočinjanju simulacije, ova klasa stvara instance izabranih protokola u svakoj od četiri različite kategorije [89].

6.1.1.3. Višestruke operacije

Višestruke operacije prvenstveno se koriste za stvaranje niza topologija, izvođenje velikog broja simulacija i testiranje velikih komponenata. Koncept simulatora je da se ove operacije izvšravaju u nezavisnoj niti, što korisniku omogućava upotrebu korisničkog interfejsa dok simulator izvodi operacije u pozadini. Ova klasa se instancira kad god se započne jedna od pomenutih operacija [89].

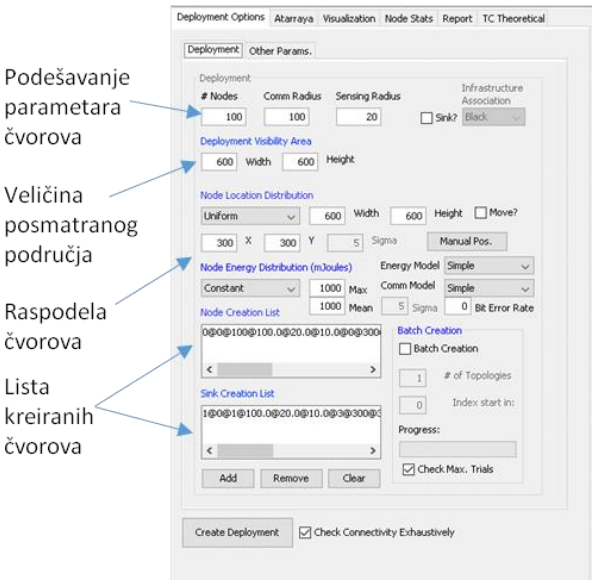
6.1.1.4. Upravljanje prikazom

Upravljanje prikazom je klasa simulatora čiji je osnovni zadatak da grafički prikaže topologije. Sve opcije za prikaz topologija definisane su ovom metodom. U simulatoru postoji više opcija za vizualni prikaz topologije [89].

6.1.2. Grafički interfejs (interface) simulatora

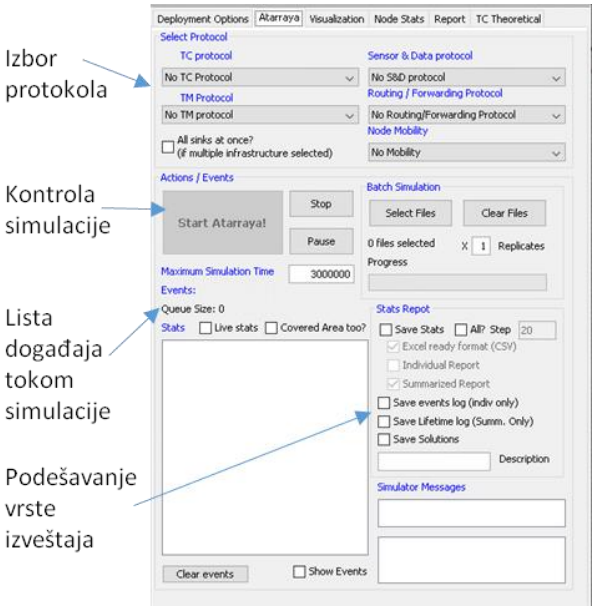
Grafički korisnički interfejs simulatora zadužen je za postavljanje parametara i upravljanje tokom simulacije. Pored toga kroz grafički interfejs moguće je pregledati rezultate simulacije kako za celi tok simulacije, tako i za svaki pojedinačni čvor. U ovom poglavlju detaljno je prikazana svaka od dostupnih opcija.

Na slici 6.2. prikazani su osnovni parametri za postavku simulacije kao što su izbor broja čvorova, komunikacioni radijus čvorova, kao i radijus očitavanja čvorova. Pored navedenih parametara moguće je birati veličinu posmatranog područja, raspodelu čvorova na području.



Slika 6.2. Opcije korisničkog interfejs simulatora za postavku simulacije

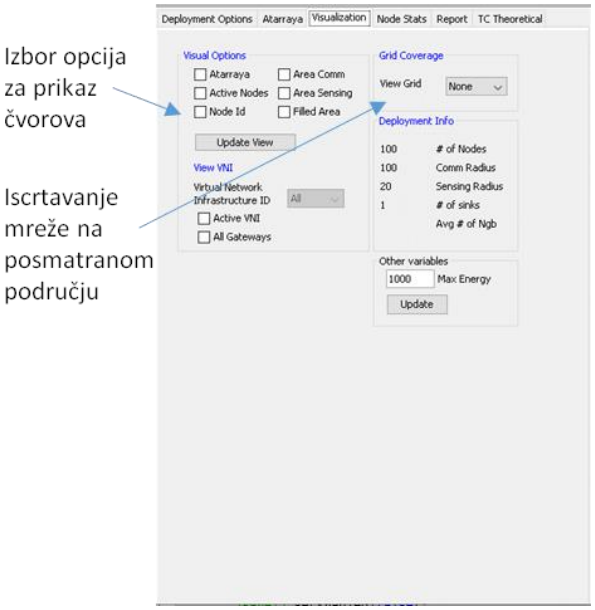
Na slici 6.3. prikazan je odeljak za izbor protokola za koji želimo da simuliramo zadato okruženje. Pored izbora protokola, samo pokretanje i kontrola simulacionog procesa takođe su naznačeni u ovom odeljku. Kao što je prikazano na slici, možemo pratiti listu događaja tokom simulacije i podešavati koje sve podatake želimo da nam budu prikazani u sumarnom izveštaju.



Slika 6.3. Opcije korisničkog interfejs simulatora za pokretanje simulacije

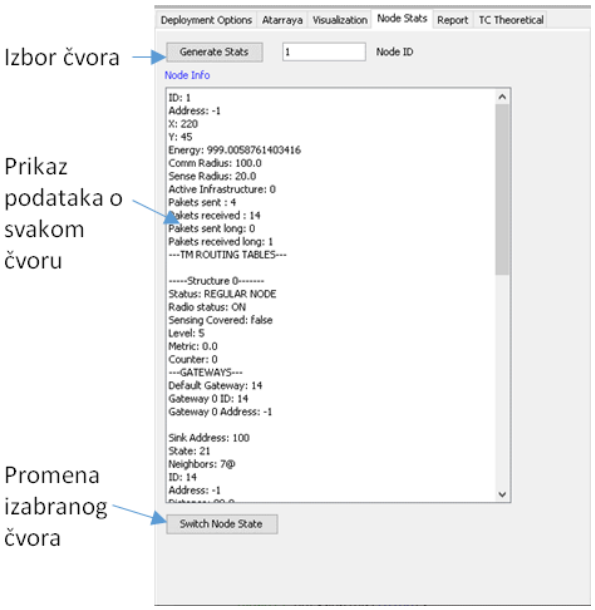
Kartica za vizualizaciju prikaza na slici 6.4. daje nam mogućnost da odaberemo koje informacije o čvorovima želimo da vidimo. Neke od informacija su ID čvora,

komunikacioni radijus, radijus očitavanja, aktivni čvorovi. Sve ove grafičke elemente radi lakše orijentacije moguće je posmatrati i kroz mrežu (engl. *grid*).



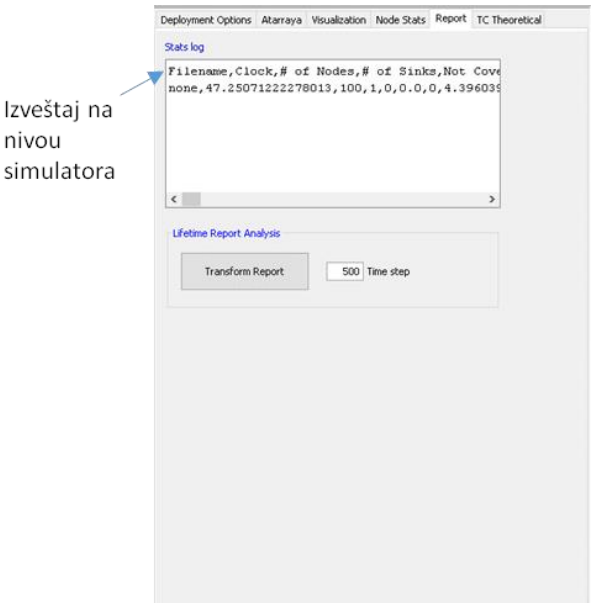
Slika 6.4. Opcije korisničkog interfejs simulatora za vizualizaciju simulacije

Na slici 6.5. prikazan je odeljak gde je posle izvršene simulacije moguće videti informacije o svakom čvoru. Izbor čvorova ide prema ID čvora, a informacije koje možemo da dobijemo za čvor odnose se na broj poruka koji je razmenjen, njegove koordinate u mreži, kom klasteru pripada i koliko ima podređenih čvorova.



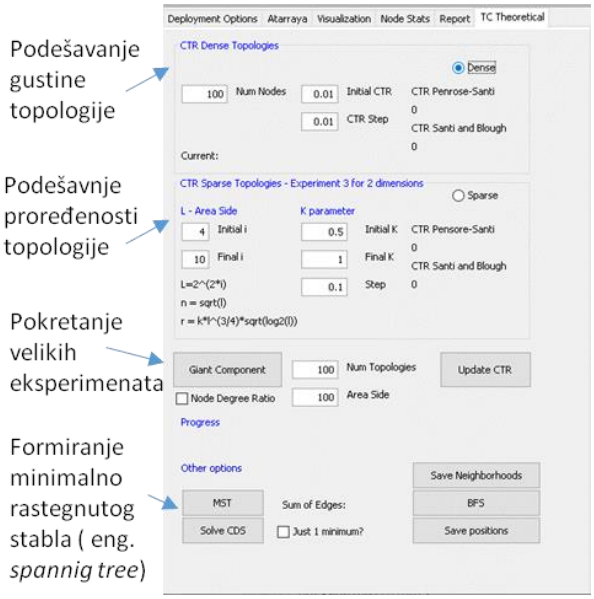
Slika 6.5. Opcije korisničkog interfejs simulatora za prikaz rezultata simulacije po čvorovima

Na slici 6.6. prikazan je odeljak sa izveštajem za celokopni simulacioni ciklus.



Slika 6.6. Opcije korisničkog interfejs simulatora za prikaz sumarnih rezultata simulacije

Na slici 6.7. prikazan je odeljak za kontrolu topologije, podešavanje gustine i proređenosti topologije, pokretanje velikih eksperimenata, kao i formiranje minimalno rastegnutog stabla (engl. *spanning tree*).



Slika 6.7. Opcije korisničkog interfejs simulatora za podešavanje dodatnih parametara

6.2. Simulacioni scenario

Simulacioni scenario zasnovali smo na posmatranju područja od 16 ha, kvadratnog oblika. Ova površina uzeta je za simulacioni model jer je uobičajena za bilo koje od polja primene bežičnih senzorskih mreža, kao i dovoljno velika za nesmetan raspored čvorova i proračun pokrivenosti komunikacionim radijusom senzorskih čvorova. Pored toga, ova

veličina sa gustinom čvorova i ostalim parametrima simulacije, koji su dati u tabeli 17, može linearno da se skalira na veće površine bez značajnijih uticaja na performanse BSM, kao i na promenu rezultata koji su prikazani u ovom radu.

U tabeli 17. prikazani su osnovni simulacioni parametri, koji su isti za sve algoritme koji se koriste za formiranje topologije, kao i za sve šifarske algoritme.

Tabela 17 . Osnovni parametri simulacije

Veličina posmatranog područja	400 x 400 m
Komunikacioni radijus čvora	100m
Radijus očitavanja čvora	20m
Veličina malog paketa	od 10 do 20 bytes
Veličina velikog paketa	od 20 do 40 bytes
Metrika simulatora	W1=0,5; W2=0,5
Raspored čvorova	normalna raspodela

Posmatrani protokoli topologija su: A3, A3 *coverage*, EECDS, CDS sa pravilom K, čije smo načine funkcionisanja opisali ranije u ovom radu, kako bismo se detaljno upoznali sa prednostima i manama svakog od njih. Kao glavna metrika za efikasnost protokola simulacije koristi se broj poslatih i primljenih poruka. Simulator u svom izvornom stanju kao rezultat simulacije daje informacije o broju velikih i malih poruka, kako poslatih tako i primljenih, ali da bismo protokole topologija mogli da posmatramo sa aspekta energetske efikasnosti svake od topologija, moramo nadograditi simulator i količinu saobraćaja svakog čvora predstaviti u bajtovima, a uzeti u obzir koeficijent pokrivenosti posmatranog pordručja na osnovu komunikacionog radijusa *claster head* čvorova.

Prema posmatranim protokolima napravljena je uporedna analiza potrošnje energije po jednom bajtu šifrovanja za šifarske algoritme prikazane u tabeli 16. Da bi postavljeni model bio validan, navedene vrednosti moraju da zadovoljavaju kriterijume normalne raspodele. Navedene vrednosti ne odstupaju značajno od normalne raspodele, što potvrđuju vrednosti koje slede za *K-S* statistički test: 0,42396 i *p-value*: 0,05545.

Potrošnja energije šifarskih algoritama prema veličini prosleđenih poruka zasnovana je na rezultatima dobijenim iz simulatora i na matematičkoj uporednoj obradi koju smo bazirali na jednostavnoj formuli i implementirali u simulator, da bismo izračunali prosek potrošnje svih šifarskih algoritama po svakom protokolu ponaosob i prikazali

odstupanje algoritama od proseka u pozitivnom i negativnom smislu. Ukupna prosečna potrošnja energije šifarskog algoritma E_{avg} po jednom simulacionom ciklusu za svaki od protokola računa se prema jednačini (6.1):

$$E_{avg} = \left(M_s * rand_{num_{btw(,x_s,y_s)}} + M_b * rand_{num_{btw(x_b,y_b)}}\right) * E_c. \quad (6.1)$$

Ovde je M_s – broj malih poruka (između 10 i 20 bajta) , M_b – broj velikih poruka (između 20 i 40 bajta), E_c – potrošnja energije šifarskih algoritama po bajtu čije vrednosti su za svaki od šifarskih algoritama prikazane u tabeli 16 u četvrtom poglavlju, funkcija *rand_num_btw(x_s,y_s)* korišćena je za generisanje slučajnih brojeva iz zadatih opsega za veličinu poruka, a x_s,y_s, x_b,y_b su opsezi veličine poruka, gde x_s predstavlja najmanju malu poruku, y_s predstavlja najveću malu poruku, x_b predstavlja najmanju veliku poruku, y_b predstavlja najveću veliku poruku.

6.3. Rezultati simulacije

Parametri koji su dobijeni kao međukorak simulacije prikazani su u tabelama koje slede i figuriraju u konačnom rezultatu zasnovanom na matematičkom modelu koji je predstavljen u prethodnom odeljku. Pored pomenutih parametara predstavićemo i grafički prikaz svih topologija sa rasporedom čvorova na posmatranom području, ali podeljeni prema *claster head* čvorovima i regularnim čvorovima koji služe samo za prikupljanje podataka.

U tabeli 18. prikazana je kompleksnost simulacije u odnosu na vreme koje je potrebno da se ona izvrši.

Tabela 18. Vreme izvršavanja simulacije po protokolu topologije

Protokol topologije	Vreme simulacije
a3	31,28
a3 *coverage*	111,35
EECDS	134,86
CDS pravilo sa K	94,71

U daljem tekstu dati su precizni rezultati simulacije, pa su ti podaci korišćeni za dalju obradu.

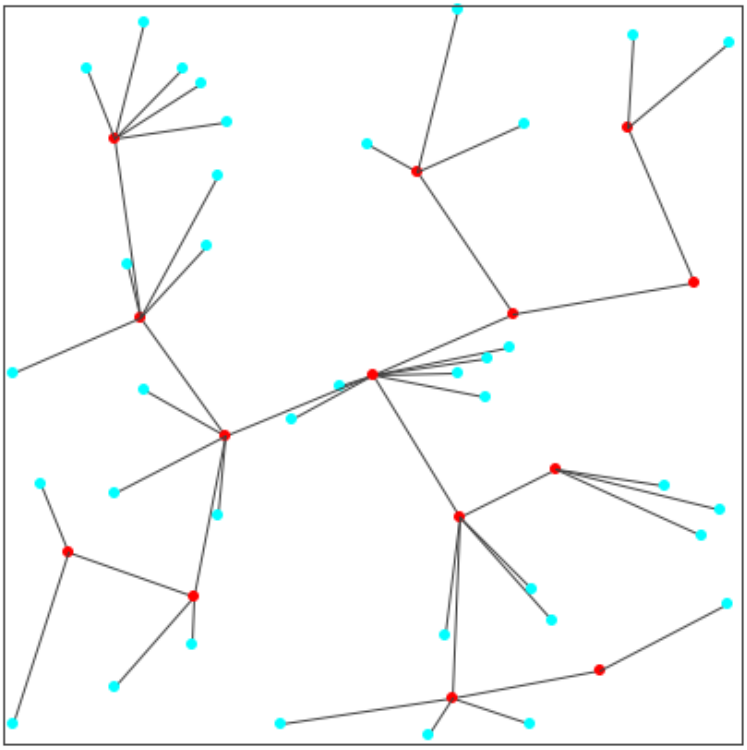
6.3.1. A3 protokol rezultati simulacije

U tabeli 19. prikazani su rezultati simulacije opterećenja sa aspekta količine saobraćaja i prenetih poruka za A3 protokol.

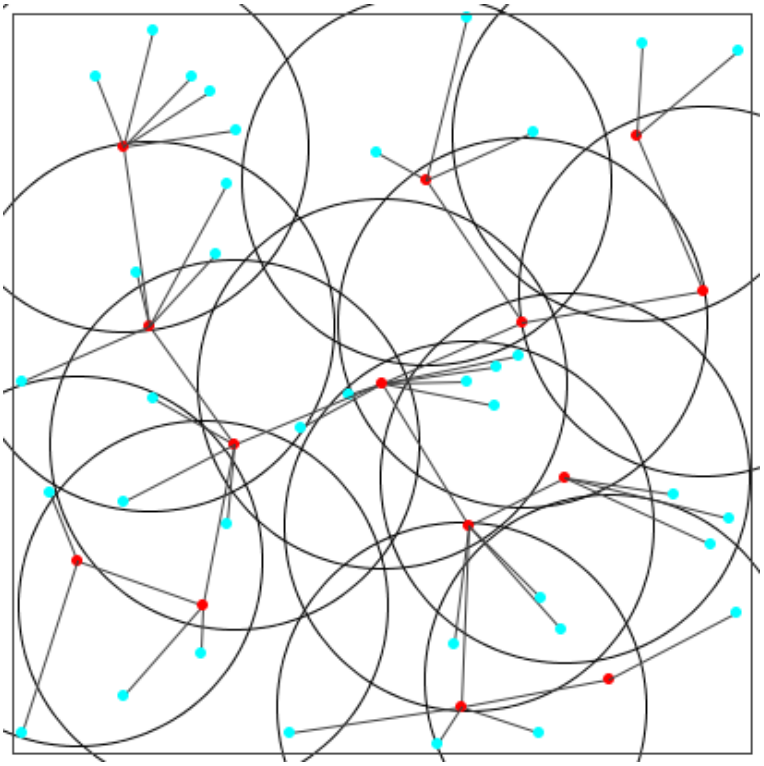
Tabela 19. Rezultati simulacije A3 protokola

Oznaka čvora	*X* koordinata	*Y* koordinata	Komunikacioni radijus	Radijus očitavanja	Broj malih poslatih paketa	Broj malih primljenih paketa	Broj velikih poslatih paketa	Broj velikih primljenih paketa
0	287	2	100,,0	20,,0	4	5	0	1
1	11	100	100,0	20,0	3	14	0	1
2	313	368	100,0	20,0	3	21	0	1
3	79	22	100,0	20,0	3	19	0	2
4	206	197	100,0	20,0	3	38	0	4
5	148	116	100,0	20,0	3	26	1	3
6	5	164	100,0	20,0	4	8	0	1
7	98	329	100,0	20,0	3	17	1	1
8	48	202	100,0	20,0	4	12	0	1
9	114	143	100,0	20,0	3	29	1	3
10	316	191	100,0	20,0	3	26	1	3
11	225	209	100,0	20,0	3	35	0	5
12	328	317	100,0	20,0	3	32	0	2
13	391	167	100,0	20,0	4	5	0	1
14	85	269	100,0	20,0	3	21	0	2
15	246	277	100,0	20,0	3	36	0	5
16	201	277	100,0	20,0	3	23	0	3
17	143	196	100,0	20,0	3	30	0	5
18	116	342	100,0	20,0	3	17	0	1
19	223	188	100,0	20,0	3	38	1	3
20	348	231	100,0	20,0	3	20	0	2
21	161	235	100,0	20,0	3	32	0	4
22	99	231	100,0	20,0	3	22	1	2
23	377	63	100,0	20,0	3	8	0	2
24	47	72	100,0	20,0	4	18	0	3
25	35	90	100,0	20,0	3	33	1	2
26	130	72	100,0	20,0	3	27	0	4
27	313	115	100,0	20,0	2	20	1	2
28	53	26	100,0	20,0	3	20	0	2
29	210	349	100,0	20,0	3	22	0	2
30	84	397	100,0	20,0	4	7	0	1
31	301	242	100,0	20,0	3	38	1	4
32	217	288	100,0	20,0	3	27	0	5
33	98	361	100,0	20,0	4	9	0	1
34	254	112	100,0	20,0	4	9	0	1
35	228	266	100,0	20,0	3	30	0	5
36	113	29	100,0	20,0	3	23	1	2
37	299	334	100,0	20,0	3	41	1	2
38	336	317	100,0	20,0	4	16	0	2
39	333	6	100,0	20,0	2	13	0	1
40	178	147	100,0	20,0	3	25	0	4
41	302	388	100,0	20,0	3	19	0	1
42	341	30	100,0	20,0	3	14	1	1
43	343	36	100,0	20,0	3	12	0	2
44	229	292	100,0	20,0	3	34	1	3
45	332	281	100,0	20,0	4	18	0	3
46	347	356	100,0	20,0	3	20	0	1
47	119	140	100,0	20,0	3	33	0	4
48	279	345	100,0	20,0	3	29	0	2
49	101	43	100,0	20,0	4	12	0	2

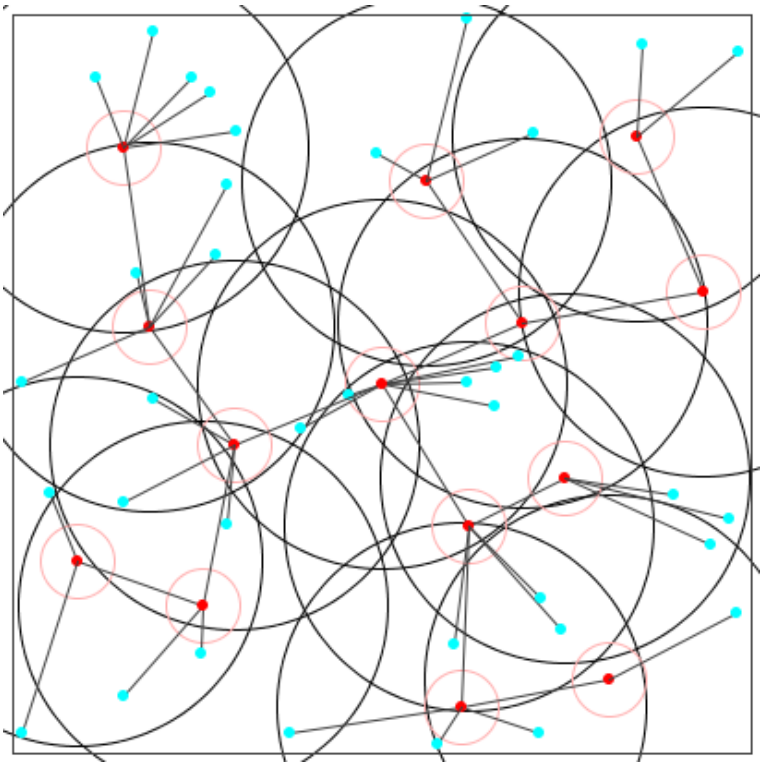
Na slikama od 6.8. do 6.11. prikazani su raspored čvorova, komunikacioni radijus i radijus očitavanja A3 protokola.



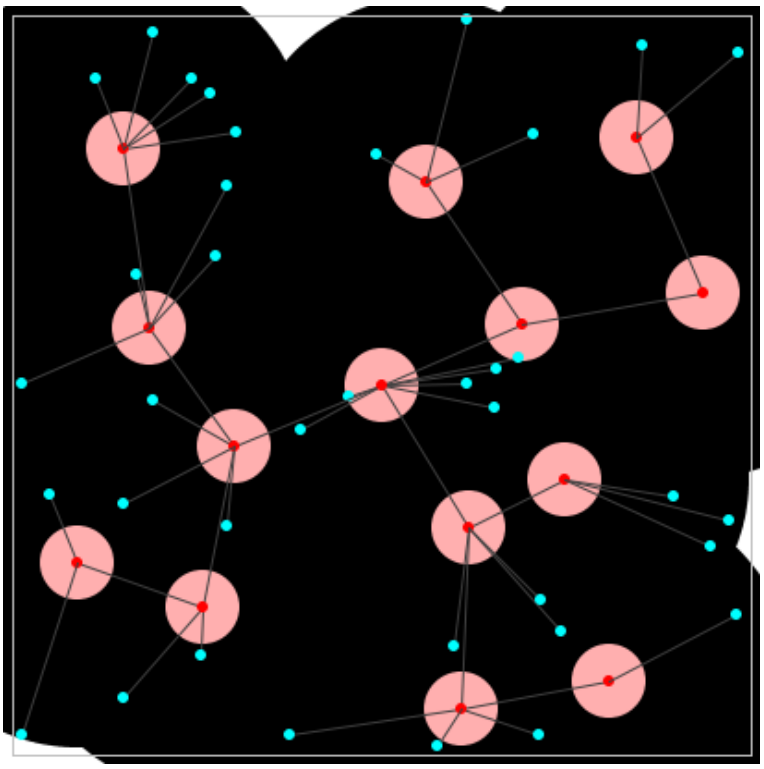
Slika 6.8. A3 protokol – raspored čvorova



Slika 6.9. A3 protokol – komunikacioni radijus čvorova



Slika 6.10. A3 protokol – radijus očitavanja čvorova



Slika 6.11. A3 protokol – pokrivenost posmatranog područja

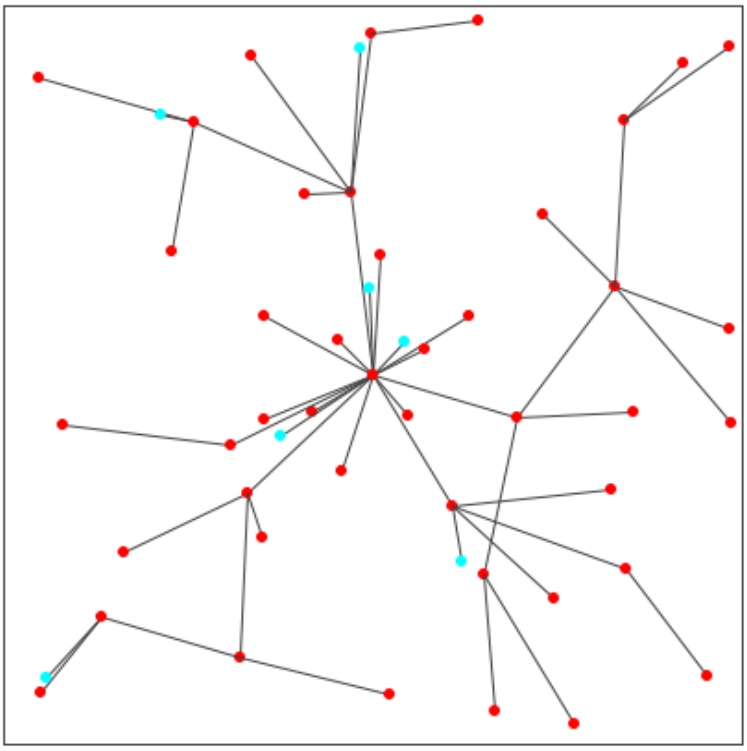
6.3.2. A3 coverage protokol – rezultati simulacije

U tabeli 20. prikazani su rezultati simulacije opterećenja sa aspekta količine saobraćaja i prenetih poruka za A3 *coverage* protokol.

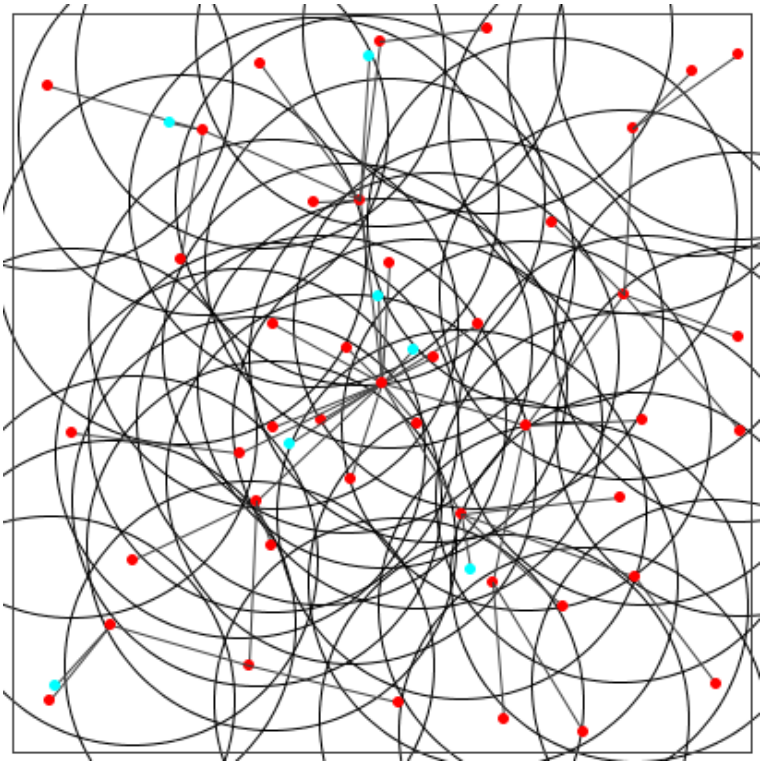
Tabela 20. Rezultati simulacije A3 coverage protokola

Oznaka čvora	X koordinata	Y koordinata	Komunikacioni radijus	Radijus očitavanja	Broj malih poslatih paketa	Broj malih primljenih paketa	Broj velikih poslatih paketa	Broj velikih primljenih paketa
0	357	351	100,0	20,0	5	27	0	1
1	241	382	100,0	20,0	3	35	1	2
2	171	315	100,0	20,0	4	43	0	3
3	246	232	100,0	20,0	4	46	0	4
4	250	122	100,0	20,0	4	33	0	3
5	173	163	100,0	20,0	3	21	1	2
6	271	312	100,0	20,0	4	51	0	3
7	260	68	100,0	20,0	5	27	0	1
8	103	297	100,0	20,0	3	41	0	3
9	50	22	100,0	20,0	5	7	0	1
10	128	94	100,0	20,0	3	17	1	3
11	195	300	100,0	20,0	3	49	1	3
12	125	307	100,0	20,0	4	45	0	2
13	229	160	100,0	20,0	4	41	0	3
14	258	350	100,0	20,0	4	43	0	3
15	234	200	100,0	20,0	2	42	0	3
16	354	153	100,0	20,0	3	21	1	2
17	397	28	100,0	20,0	3	13	1	1
18	71	82	100,0	20,0	4	18	0	3
19	43	303	100,0	20,0	4	34	0	2
20	201	278	100,0	20,0	4	44	0	2
21	44	357	100,0	20,0	5	29	0	2
22	21	110	100,0	20,0	3	12	1	1
23	298	350	100,0	20,0	3	32	0	2
24	272	166	100,0	20,0	3	35	0	4
25	97	280	100,0	20,0	3	46	1	2
26	199	308	100,0	20,0	3	45	0	2
27	69	347	100,0	20,0	4	30	0	2
28	40	271	100,0	20,0	4	34	0	2
29	294	183	100,0	20,0	3	33	1	2
30	187	16	100,0	20,0	3	16	1	1
31	274	208	100,0	20,0	4	36	0	3
32	220	18	100,0	20,0	4	17	0	1
33	217	258	100,0	20,0	4	41	0	2
34	303	36	100,0	20,0	5	17	0	1
35	367	10	100,0	20,0	5	8	0	1
36	19	301	100,0	20,0	3	31	1	1
37	336	382	100,0	20,0	4	31	0	2
38	22	258	100,0	20,0	4	17	0	2
39	277	155	100,0	20,0	4	37	0	3
40	321	192	100,0	20,0	4	34	0	3
41	386	303	100,0	20,0	5	19	0	1
42	315	318	100,0	20,0	3	32	1	1
43	155	31	100,0	20,0	4	14	0	2
44	377	111	100,0	20,0	3	10	1	2
45	38	363	100,0	20,0	4	16	0	1
46	303	351	100,0	20,0	4	31	0	2
47	158	286	100,0	20,0	5	32	0	3
48	54	156	100,0	20,0	3	10	1	2
49	179	315	100,0	20,0	3	43	0	3

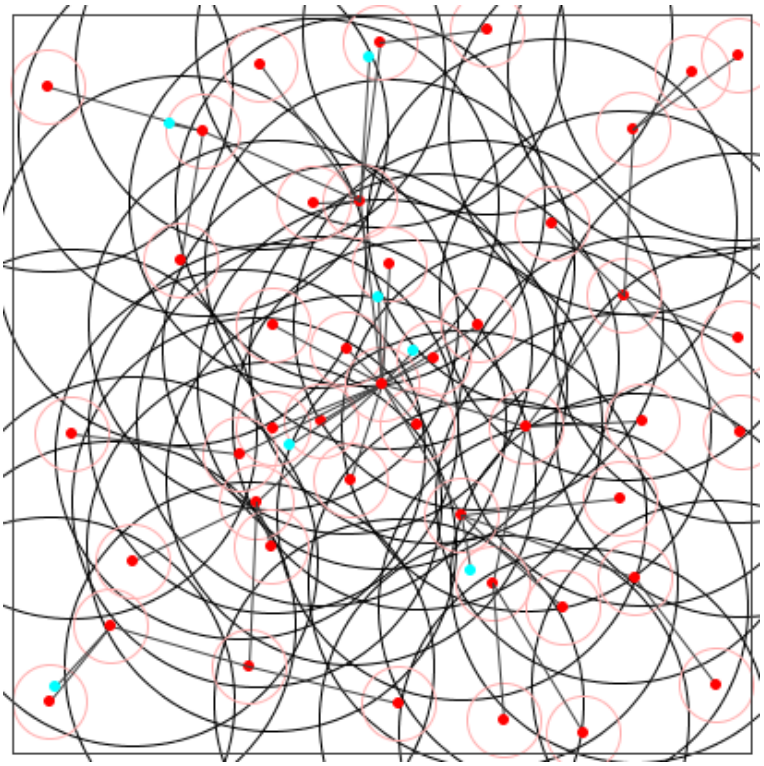
Na slikama od 6.12. do 6.15. prikazani su raspored čvorova, komunikacioni radijus i radijus očitavanja A3 *coverage* protokola.



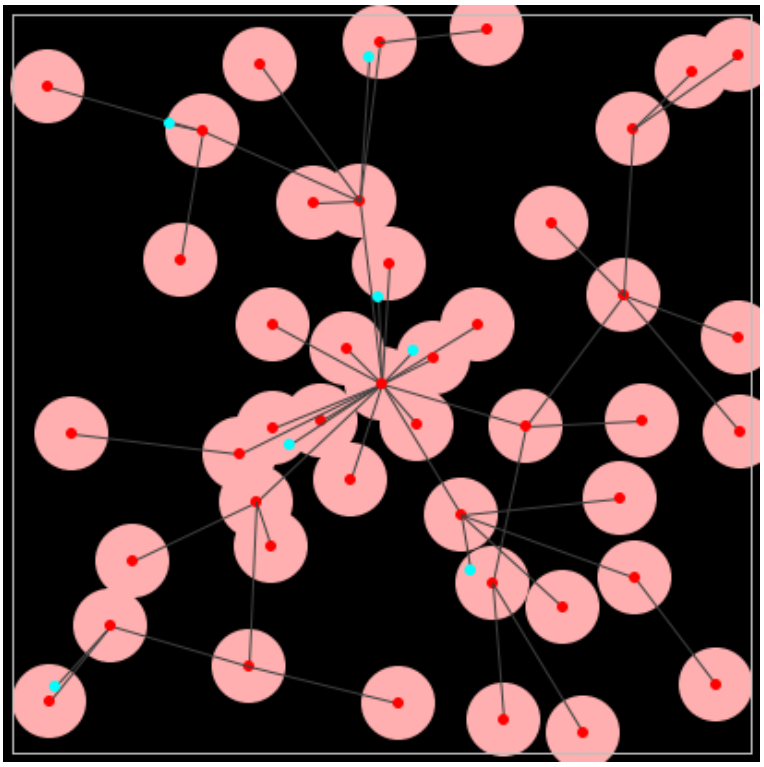
Slika 6.12. A3 coverage protokol – raspored čvorova



Slika 6.13. A3 coverage protokol – komunikacioni radijus čvorova



Slika 6.14. A3 coverage protokol – radijus očitavanja čvorova



Slika 6.15. A3 coverage protokol – pokrivenost posmatranog područja

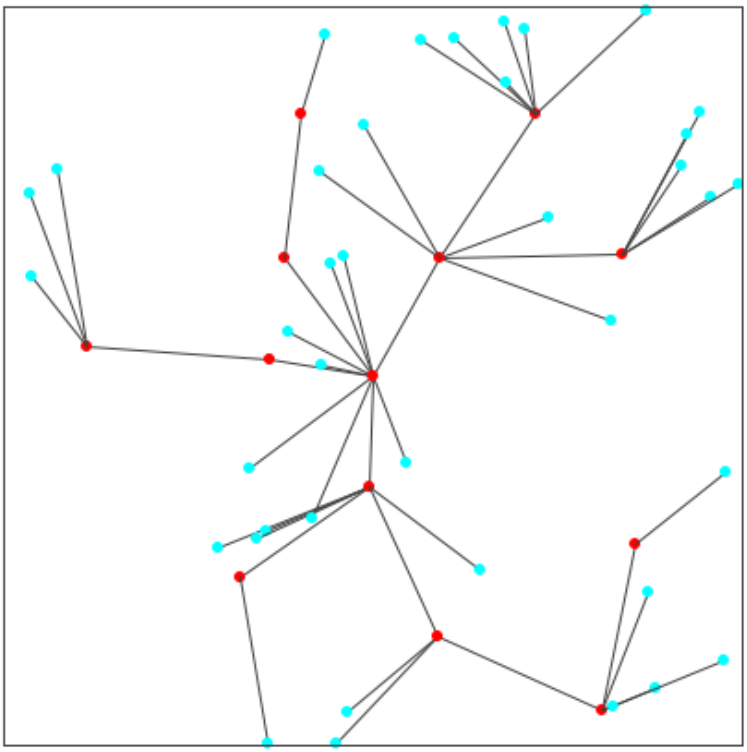
6.3.3. Rezultati simulacije protokola CDS sa pravilom K

U tabeli 21. prikazani su rezultati simulacije opterećenja sa aspekta količine saobraćaja i prenetih poruka za protokol CDS sa pravilom K.

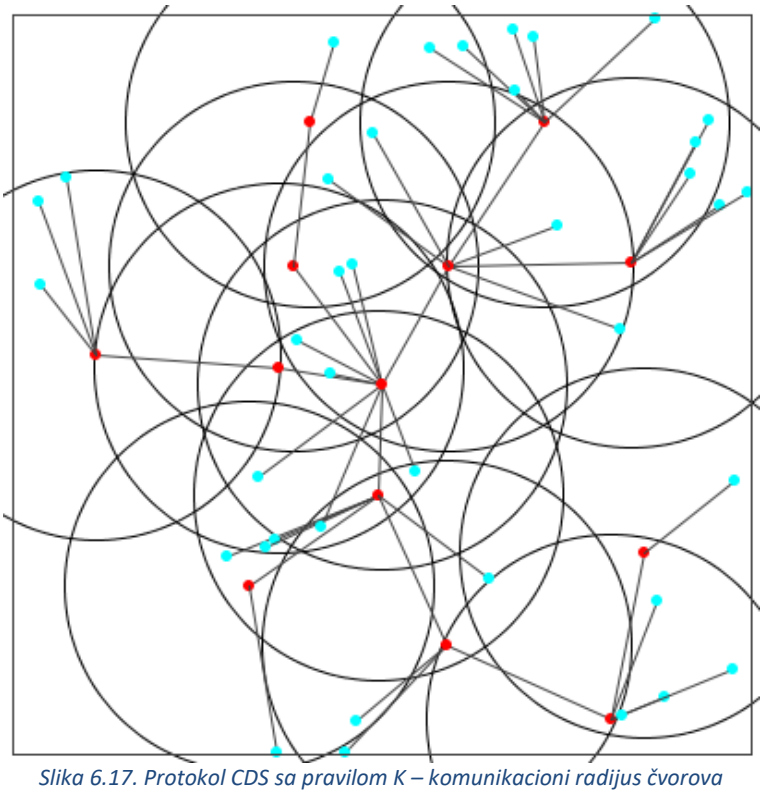
Tabela 21. Rezultati simulacije protokola CDS sa pravilom K

Oznaka čvora	X koordinata	Y koordinata	Komunikacioni radijus	Radijus očitavanja	Broj malih poslatih paketa	Broj malih primljenih paketa	Broj velikih poslatih paketa	Broj velikih primljenih paketa
0	116	293	100,0	20,0	8	50	0	0
1	377	57	100,0	20,0	7	53	0	0
2	282	12	100,0	20,0	6	42	0	0
3	383	103	100,0	20,0	8	46	0	0
4	342	291	100,0	20,0	7	48	0	0
5	186	382	100,0	20,0	6	28	0	0
6	353	369	100,0	20,0	6	34	0	0
7	45	184	100,0	20,0	6	32	0	0
8	142	284	100,0	20,0	12	72	0	0
9	335	134	100,0	20,0	6	60	0	0
10	128	309	100,0	20,0	8	62	0	0
11	295	114	100,0	20,0	6	62	0	0
12	171	89	100,0	20,0	10	67	0	0
13	180	399	100,0	20,0	6	20	0	0
14	288	58	100,0	20,0	6	91	0	0
15	226	18	100,0	20,0	8	70	0	0
16	324	381	100,0	20,0	5	43	0	0
17	244	17	100,0	20,0	7	61	0	0
18	137	288	100,0	20,0	12	72	0	0
19	174	15	100,0	20,0	8	50	0	0
20	144	191	100,0	20,0	14	92	0	0
21	154	176	100,0	20,0	6	80	0	0
22	329	170	100,0	20,0	6	34	0	0
23	218	247	100,0	20,0	6	89	0	0
24	348	2	100,0	20,0	6	49	0	0
25	198	260	100,0	20,0	6	105	0	0
26	258	305	100,0	20,0	8	40	0	0
27	133	250	100,0	20,0	6	89	0	0
28	143	399	100,0	20,0	6	20	0	0
29	236	136	100,0	20,0	6	81	0	0
30	14	101	100,0	20,0	6	18	0	0
31	195	64	100,0	20,0	9	90	0	0
32	349	317	100,0	20,0	7	48	0	0
33	152	136	100,0	20,0	7	79	0	0
34	177	139	100,0	20,0	7	79	0	0
35	272	41	100,0	20,0	8	55	0	0
36	391	252	100,0	20,0	7	14	0	0
37	184	135	100,0	20,0	7	79	0	0
38	367	86	100,0	20,0	8	58	0	0
39	390	354	100,0	20,0	6	34	0	0
40	167	277	100,0	20,0	6	99	0	0
41	330	379	100,0	20,0	9	31	0	0
42	398	96	100,0	20,0	6	36	0	0
43	15	146	100,0	20,0	6	18	0	0
44	271	8	100,0	20,0	7	58	0	0
45	161	58	100,0	20,0	9	63	0	0
46	370	69	100,0	20,0	7	53	0	0
47	235	341	100,0	20,0	8	43	0	0
48	172	194	100,0	20,0	6	100	0	0
49	29	88	100,0	20,0	6	18	0	0

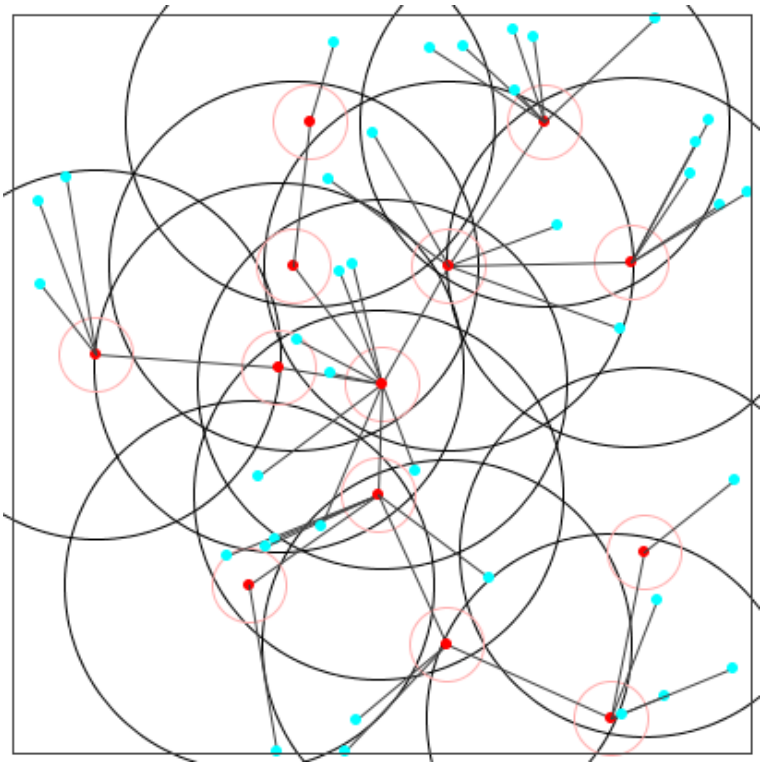
Na slikama od 6.16. do 6.19. prikazani su raspored čvorova, komunikacioni radijus i radijus očitavanja za protokol CDS sa pravilom K.



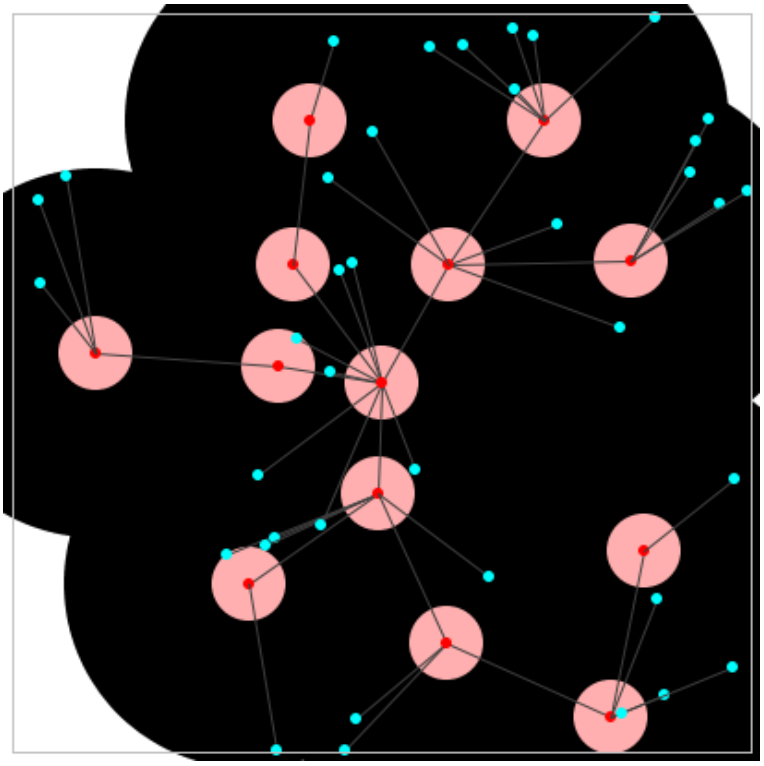
Slika 6.16. Protokol CDS sa pravilom K – raspored čvorova



Slika 6.17. Protokol CDS sa pravilom K – komunikacioni radijus čvorova



Slika 6.18. Protokol CDS sa pravilom K – radijus očitavanja čvorova



Slika 6.19. Protokol CDS sa pravilom K – pokrivenost posmatranog područja

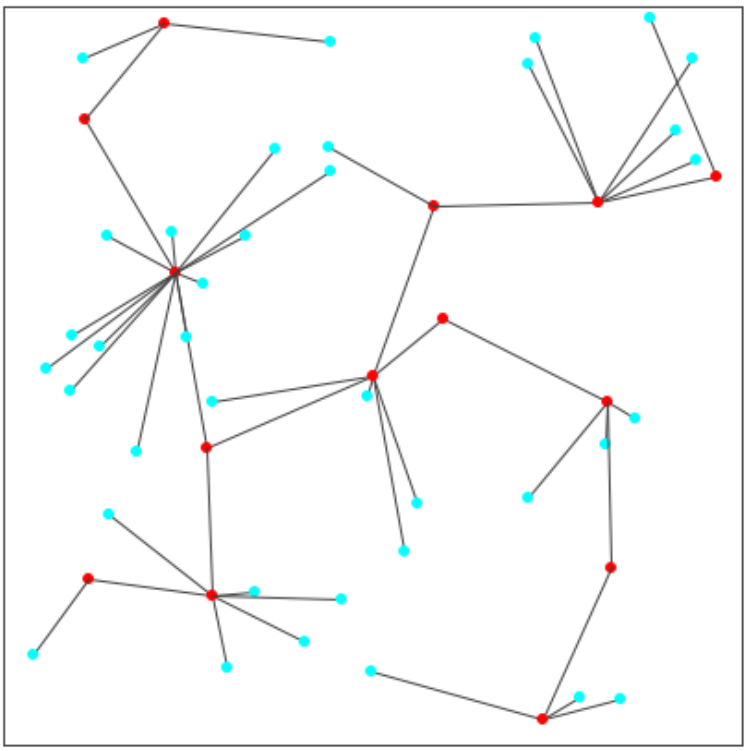
6.3.4. Rezultati simulacije za EECDS protokol

U tabeli 22. prikazani su rezultati simulacije opterećenja sa aspekta količine saobraćaja i prenetih poruka za protokol EECDS.

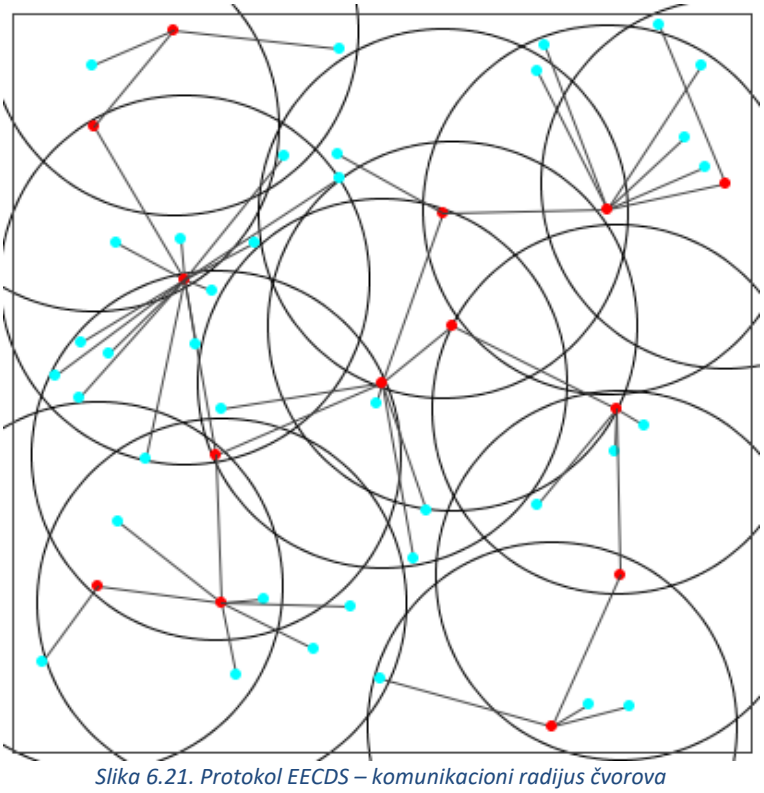
Tabela 22. Rezultati simulacije EECDS protokola

Oznaka čvora	X koordinata	Y koordinata	Komunikacioni radijus	Radijus očitavanja	Broj malih poslatih paketa	Broj malih primljenih paketa	Broj velikih poslatih paketa	Broj velikih primljenih paketa
0	233	108	100,0	20,0	4	78	0	0
1	136	317	100,0	20,0	29	180	0	0
2	284	266	100,0	20,0	7	30	0	0
3	108	150	100,0	20,0	30	299	0	0
4	91	122	100,0	20,0	27	253	0	0
5	37	178	100,0	20,0	23	259	0	0
6	56	124	100,0	20,0	14	234	0	0
7	350	6	100,0	20,0	11	31	0	0
8	46	310	100,0	20,0	22	145	0	0
9	197	211	100,0	20,0	5	19	0	0
10	284	31	100,0	20,0	7	38	0	0
11	373	28	100,0	20,0	5	45	0	0
12	44	61	100,0	20,0	6	89	0	0
13	113	214	100,0	20,0	3	292	0	0
14	342	223	100,0	20,0	5	26	0	0
15	93	144	100,0	20,0	29	285	0	0
16	36	208	100,0	20,0	21	233	0	0
17	23	196	100,0	20,0	23	258	0	0
18	177	19	100,0	20,0	12	69	0	0
19	375	83	100,0	20,0	4	34	0	0
20	72	241	100,0	20,0	29	291	0	0
21	110	239	100,0	20,0	4	322	0	0
22	163	344	100,0	20,0	16	108	0	0
23	52	184	100,0	20,0	25	277	0	0
24	99	179	100,0	20,0	23	248	0	0
25	57	275	100,0	20,0	31	217	0	0
26	217	295	100,0	20,0	3	92	0	0
27	177	89	100,0	20,0	17	161	0	0
28	176	76	100,0	20,0	21	129	0	0
29	334	375	100,0	20,0	9	26	0	0
30	121	358	100,0	20,0	14	125	0	0
31	87	9	100,0	20,0	12	44	0	0
32	326	237	100,0	20,0	5	26	0	0
33	224	269	100,0	20,0	3	92	0	0
34	327	214	100,0	20,0	8	27	0	0
35	386	92	100,0	20,0	5	33	0	0
36	131	124	100,0	20,0	21	207	0	0
37	147	77	100,0	20,0	19	171	0	0
38	199	360	100,0	20,0	16	119	0	0
39	16	351	100,0	20,0	10	53	0	0
40	288	17	100,0	20,0	5	36	0	0
41	312	374	100,0	20,0	9	27	0	0
42	364	67	100,0	20,0	5	45	0	0
43	322	106	100,0	20,0	8	35	0	0
44	43	28	100,0	20,0	7	32	0	0
45	238	169	100,0	20,0	4	19	0	0
46	329	304	100,0	20,0	6	55	0	0
47	292	386	100,0	20,0	12	40	0	0
48	113	319	100,0	20,0	28	175	0	0
49	183	321	100,0	20,0	15	108	0	0

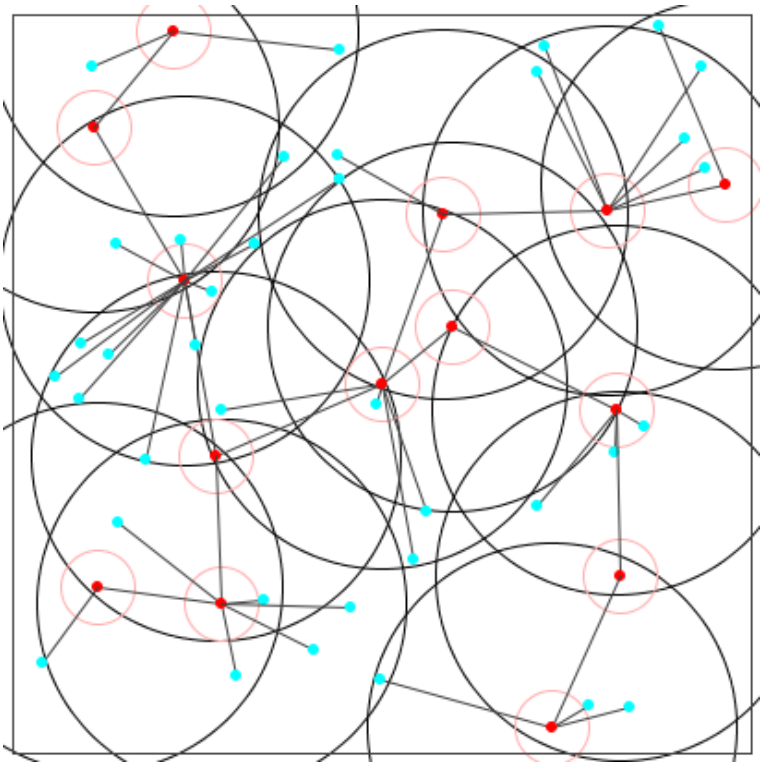
Na slikama od 6.20. do 6.23. prikazani su raspored čvorova, komunikacioni radijus i radijus očitavanja za protokol EECDS.



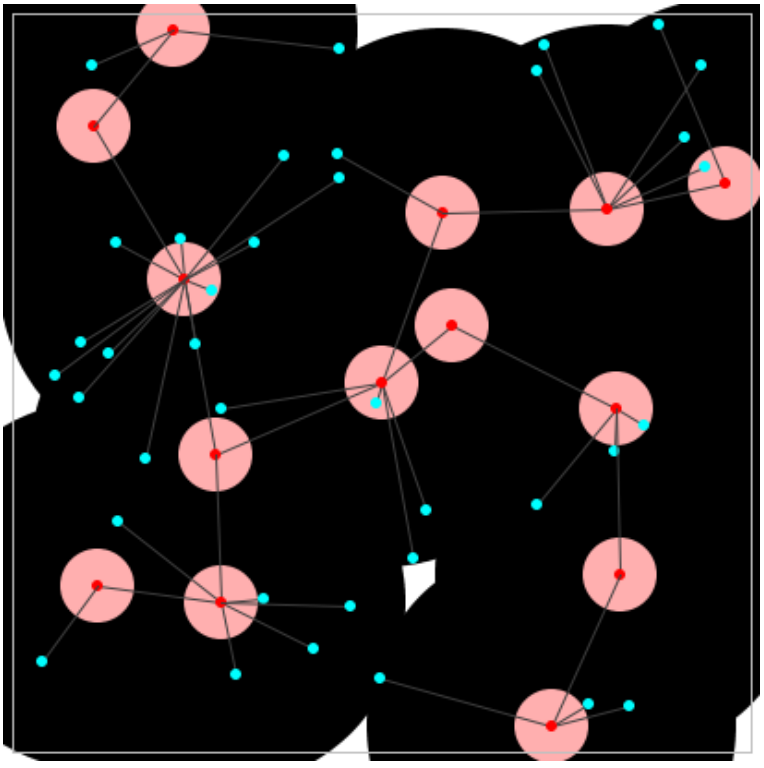
Slika 6.20. Protokol EECDS – raspored čvorova



Slika 6.21. Protokol EECDS – komunikacioni radijus čvorova



Slika 6.22, Protokol EECDS – radijus očitavanja čvorova



Slika 6.23. Protokol EECDS – pokrivenost posmatranog područja

U tabeli 23. dati su rezultati koji prikazuju koeficijent pokrivenosti posmatranog pordučja, sa aspekta komunikacionog radijusa *claster head* čvorova, kao i koliki deo područja je ostao nepokriven i koji procenat je izvan posmatranog područja. Pokrivenost posmatranog područja je od značaja za dobijanje konačnih rezultata koji su predstavljeni u ovom radu. Jedan od ključnih rezultata simulacije jeste količina saobraćaja ustanovljena na osnovu broja primljenih i poslatih poruka u bajtovima kako za glavu klastera tako i za regularne čvorove unutar senzorske mreže i prikazan je u tabeli 24.

Tabela 23. Pokrivenost posmatranog područja komunikacionim radijusom

Protokol topologije	Pokrivenost	Prekriveno izvan posmatranog područja	Ostalo nepokriveno područje
a3	0,9266	0,2255	0,0734
a3 *coverage*	1	0,09	0
EECDS	0,94	0,09	0,06
CDS pravilo sa K	0,92	0,04	0,08

Vrednosti u tabeli 23. za koeficijent pokrivenosti ne odstupaju značajno od normalne raspodele, što pokazuju sledeći parametri za *K-S* statistički test je 0,28748 i *p-value* je 0,71277.

Tabela 24. Ukupna količina saobraćaja prema protokolima topologija i tipovima čvorova

Protokol	Tip čvora	Količina prenetih podataka, bajt (byte)	
		Poslati	Primljeni
A3	Glava klastera	2608	10589
	Regularan čvor	1542	10970
A3 *coverage*	Glava klastera	3906	19341
	Regularan čvor	403	4885
EECDS	Glava klastera	2440	12695
	Regularan čvor	3610	17146
CDS sa pravilom K	Glava klastera	1755	9661
	Regularan čvor	3083	17403

Iz tabele 24. izračunat je ukupan broj bajtova koji je poslat za svaku topologiju i služi kao jedan od ulaznih parametara modela za određivanje prihvatljivih parova topologije i šifarskog algoritma.

Tabela 25. Ukupan broj poslatih poruka prema protokolima

Protokol	A3	A3 *coverage*	EECDS	CDS sa pravilom K
Ukupan broj poslatih poruka	4150	4309	6050	4838

Vrednosti u tabeli 25. koje predstavljaju ukupan broj poslatih poruka u bajtovima ne odstupaju značajno od normalne raspodele, što pokazuju sledeći parametri za *K-S* statistički test je 0,29844 dok je *p-value* 0,67012.

7. Analiza rezultata istraživanja

Matematički model za određivanje koeficijenta podobnosti kombinacije šifarskih algoritama i protokola topologija BSM prikazanog u poglavlju 5.1. implementirali smo u postojeći simulator za BSM. Algoritam implementacije prikazan je u poglavlju 5.2. Koeficijent podobnosti je dobijen na osnovu prikazanog matematičkog modela koji je primenjen i potvrđen simulacijom. Simulaciono okruženje i rezultati simulacije predstavljeni su u poglavlju 6.

Kao potvrdu predloženog modela i rezultat simulacije dobili smo *koeficijente podobnosti,* prikazane u tabeli 26. Koeficijent podobnosti je određen za kombinaciju protokola topologije i šifarskog algoritma. Na osnovu intervala matematičkog modela prikazanog u jednačini 5.8. određujemo koji od koeficijenata podobnosti su prihvatljivi za implementaciju u našoj studiji slučaja iz poglavlja 6.2. U tabeli 26. zelenom bojom prikazane su sve one kombinacije koje su prema našem sistemu odlučivanja i dobijenih rezultata prihvatljive za primenu i implementaciju, dok su crvenom bojom predstavljeni oni algoritmi koji sa stanovišta energetske efikasnosti ne zadovoljavaju postavljene kriterijume.

Ako detaljnije posmatramo vrednosti koje se nalaze u tabeli 26, jasno možemo da uočimo da je kombinacija *Katan64* i A3 protokola dala najbolje rezultate. Pored toga, možemo da primetimo da se protokoli topologija A3 i A3 *coverage* pojavljuju među podobnim algoritmima čak po 8 od 19 puta. Isto tako, šifarski algoritmi kao što su *KATAN64*, *Present* i *Prince* figuriraju kao adekvatan izbor sa 3 od ukupno 4 protokola topologija.

Tabela 26. Prikaz prihvatljivih parova šifarskih algoritama i protokola topologija

	A3	A3 *coverage*	EECDS	CDS sa pravilom K
AES	61.753,12	63.524,16	136.074,04	113.562,02
NOEKEON	59.724,73	61.437,60	131.604,45	109.831,88
LED 128	231.236,13	237.867,84	509.532,70	425.235,89
Present	54.997,08	56.337,12	120.678,80	100.713,76
Prince	52.733,83	54.018,72	115.712,59	96.569,16
Piccolo	62.918,48	64.451,52	138.060,52	115.219,86
TWINE	75.819,03	77.666,40	166.367,89	138.844,08
Simon64/96	75.140,05	76.970,88	164.878,03	137.600,70
KATAN64	49.565,27	50.772,96	108.759,90	90.766,72

Radi bolje preglednosti i lakšeg tumačenja rezultata iz tabele 26, prikazana je i rang-lista koeficijenata podobnosti uređenih parova šifarskih algoritama i protokola topologija u rastućem redosledu u tabeli 27. Koeficijent podobnosti računa se na osnovu formule (5.1) koja je prikazana u petom poglavlju. Što je manji koeficijent podobnosti, to znači da je ta kombinacija šifarskog algoritma i protokola topologije energetski efikasnija.

Tabela 27. Rang-lista kombinacija šifarskih algoritama i protokola topologija

Rang	Šifarski algoritam	Protokol topologije	Koeficijent podobnosti
1	*KATAN64*	A3	49.565,27
2	*KATAN64*	A3 *coverage*	50.772,96
3	*Prince*	A3	52.733,83
4	*Prince*	A3 *coverage*	54.018,72
5	*Present*	A3	54.997,08
6	*Present*	A3 *coverage*	56.337,12
7	*Noekeon*	A3	59.724,73
8	*NOEKEON*	A3 *coverage*	61.437,60
9	*AES*	A3	61.753,12
10	*Piccolo*	A3	62.918,48
11	*AES*	A3 *coverage*	63.524,16
12	*Piccolo*	A3 *coverage*	64.451,52
13	*Simon64/96*	A3	75.140,05
14	*TWINE*	A3	75.819,03
15	*Simon64/96*	A3 *coverage*	76.970,88
16	*TWINE*	A3 *coverage*	77.666,40
17	*KATAN64*	CDS pravilo sa K	90.766,72
18	*Prince*	CDS pravilo sa K	96.569,16
19	*Present*	CD pravilo sa K	100.713,76
20	*KATAN64*	EECDS	108.759,90
21	*NOEKEON*	CDS pravilo sa K	109.831,88
22	*AES*	CDS pravilo sa K	113.562,02
23	*Piccolo*	CDS pravilo sa K	115.219,86
24	*Prince*	EECDS	115.712,59
25	Present	EECDS	120.678,80
26	*NOEKEON*	EECDS	131.604,45
27	*AES*	EECDS	136.074,04
28	*Simon64/96*	CDS pravilo sa K	137.600,70
29	*Piccolo*	EECDS	138.060,52
30	*TWINE*	CDS pravilo sa K	138.844,08
31	*Simon64/96*	EECDS	164.878,03
32	*TWINE*	EECDS	166.367,89
33	*LED 128*	A3	231.236,13
34	*LED 128*	A3 *coverage*	237.867,84
35	*LED 128*	CD pravilo sa K	425.235,89
36	*LED 128*	EECDS	509.532,70

8. Zaključak

U ovom radu predstavljen je pregled arhitekture sistema na kome se zasniva upotreba bežičnih senzorskih mreža kao osnov za komunikaciju IoT uređaja. Posebna pažnja usmerena je na obezbeđivanje kriterijuma za bezbednu komunikaciju između uređaja. Da bi se osigurao bezbedan prenos podataka, predstavljene su različite vrste napada, mehanizmi odbrane od pomenutih napada, a posebana pažnja posvećena je lakim šifarskim algoritmima.

Pored bezbednosti prenosa podataka, i energetska efikasnost jedan je od bitnih aspekata razmatranih u ovom radu. Energetska efikasnost posmatrana je kroz protokole topologija senzorskih mreža, koje su uzete kao osnovna komunikaciona tehnologija između IoT uređaja.

U radu je predstavljen matematički model za odlučivanje koji je uređeni par šifarskog algoritma i protokola topologije najpogodniji sa aspekta energetske efikasnosti. Ovaj matematički model je glavni doprinos ove teze. Prednost modela je njegova jednostavnost, dok je njegova pouzdanost evaluirana kroz similaciju, a dobijeni rezultati prezentovani.

Dalji tok istraživanja može da ide u smeru implementacije nekih drugih vrsta algoritama, kao i ponašanja protokola topologija na prostorima sa velikom visinskom razlikom na maloj površini. Još jedan pravac istraživanja može da bude unapređenje matematičkog modela za odlučivanje proširivanjem ulaznih parametara i određivanjem njihovog međusobnog odnosa. Pored uvođenja novih parametara, za postojeće parametre moguće je odrediti težinske koeficijente i na taj način utvrditi koliko oni figuriraju u celokupnom modelu.

9. Literatura

[1] H. Jawad, R. Nordin, S. Gharghan, A. Jawad, M. Ismail, “Energy-Efficient Wireless Sensor Networks for Precision Agriculture: A Review,” Sensors, vol. 17, no. 8, p. 1781, Aug. 2017.

[2] E. Shi, A. Perrig, “Designing Secure Sensor Networks,” IEEE Wireless Commun., vol. 11, no. 6, pp. 38–43, Dec. 2004.

[3] H. Yang, F. Ye, Y. Yuan, S. Lu, W. Arbaugh, “Toward resilient security in wireless sensor networks,” presented at the the 6th ACM international symposium, 2005.

[4] S. Jajodia, S. Zhu, S. Setia, "Efficient security mechanisms for large-scale distributed sensor networks," In Proceedings of ACM conference on computer and communications security, pp. 62–72, 2003.

[5] N. Radosavljević, Dj. Babić, "Power Consumption Analysis Model in Wireless Sensor Network for Different Topology Protocols and Lightweight Cryptographic Algorithms," *prihvaćen za objavu u Journal of internet technology, 2020, ISSN: 1607-9264.*

[6] D. Miorandi, S. Sicari, F. De Pellegrini, I. Chlamtac, “Internet of things: Vision, applications and research challenges,” Ad Hoc Networks, vol. 10, no. 7, pp. 1497–1516, Sep. 2012.

[7] Zaheeruddin, H. Gupta, “Foundation of IoT: An Overview,” in Internet of Things (IoT), Springer International Publishing, 2020, pp. 3–24.

[8] D. Drajić, "Uvod u IoT," Univerzitet u Beogradu – Elektrotehnički fakultet, Akademska misao, Beograd, 2017.

[9] B. M. H. Alhafidh, W. Allen, "Design and simulation of a smart home managed by an intelligent self-design and simulation of a smart home managed by an intelligent self-adaptive system," *International Journal of Engineering Research and Applications, vol. 6, no. 8, pp. 64–90, 2016.*

[10] D. Drajić, "Biznis model za IoT rešenja," Univerzitet u Beogradu – Elektrotehnički fakultet, Akademska misao, Beograd, 2018.

[11] T. J. Devadas, S. Thayammal,A. Ramprakash, “IoT Data Management, Data Aggregation and Dissemination,” in Intelligent Systems Reference Library, Springer International Publishing, pp. 385–411, 2019.

[12] M. A. Ahad, G. Tripathi, S. Zafar, F. Doja, “IoT Data Management—Security Aspects of Information Linkage in IoT Systems,” in Intelligent Systems Reference Library, Springer International Publishing, pp. 439–464, 2019.

[13] W. Kassab, K. A. Darabkh, “A–Z survey of Internet of Things: Architectures, protocols, applications, recent advances, future directions and recommendations,” Journal of Network and Computer Applications, vol. 163, p. 102663, Aug. 2020.

[14] G. Filios, I. Katsidimas, S. Nikoletseas, A. Souroulagkas, P. Spirakis, I. Tsenempis, “A smart energy management power supply unit for low-power IoT systems,” *presented at the 2020 16th International Conference on Distributed Computing in Sensor Systems (DCOSS), May 2020.*

[15] Z. Fatima, L. Bhargava, and A. Kumar, “Smart Infrastructures,” in Internet of Things (IoT), Springer International Publishing, pp. 301–314, 2020.

[16] D. Drajić, "Pametni gradovi," Univerzitet u Beogradu – Elektrotehnički fakultet, Akademska misao, Beograd, 2018.

[17] M. S. Patil, V. N. Bhonge, "Wireless sensor network and RFID for smart parking system," International Journal of Emerging Technology and Advanced Engineering, vol. 3, pp. 188–192, 2013.

[18] S. K. Janahan, V. Murugappan, S. Arun, N. Kumar, R. Anandan, J. Shaik, "IoT based smart traffic signal monitoring system using vehicles counts," International Journal of Engineering & Technology, vol. 7, 2018.

[19] P. Tambare, P. Venkatachalam, D. Rajendra, "Internet of things based intelligent street lighting system for smart city," International Research Journal of Engineering and Technology, vol. 5, no. 8, 2016.

[20] M. Alam, K. A. Shakil, "Cloud Database Management System Architecture," UACEE International Journal of Computer Science and its Applications, vol. 3, no. 1, pp. 27–31, 2013.

[21] S. Sonawdekar, G. Katkar, M. Gaikwad, M. Farhan, "Smart Hospitals using IOT," International Journal of Scientific & Engineering Research, vol. 9, no. 3, pp. 120, 2018.

[22] M. Alam, B. Alam, "Cloud query language for cloud database," In Proceeding of the international conference on Recent Trends in Computing and Communication Engineering –RTCCE 2013, Hamirpur, HP, pp. 108–112, 2013.

[23] V. Rajs et al., “Realization of Instrument for Environmental Parameters Measuring,” ElAEE, vol. 20, no. 6, Jun. 2014.

[24] I. Jawhar, N. Mohamed, J. Al-Jaroodi, S. Zhang, “A Framework for Using Unmanned Aerial Vehicles for Data Collection in Linear Wireless Sensor Networks,” J Intell Robot Syst, vol. 74, no. 1–2, pp. 437–453, Oct. 2013.

[25] Y. D. Kim, Y. M. Yang, W. S. Kang, D. K. Kim, “On the design of beacon based wireless sensor network for agricultural emergency monitoring systems,” Computer Standards & Interfaces, vol. 36, no. 2, pp. 288–299, Feb. 2014.

[26] T. Ojha, S. Misra, N. S. Raghuwanshi, “Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges,” Computers and Electronics in Agriculture, vol. 118, pp. 66–84, Oct. 2015.

[27] https://standards.ieee.org/standard/802_15_4v-2017.html (pristupano 05.03.2020)

[28] B. Bhushan, G. Sahoo, “Recent Advances in Attacks, Technical Challenges, Vulnerabilities and Their Countermeasures in Wireless Sensor Networks,” Wireless Pers Commun, vol. 98, no. 2, pp. 2037–2077, Sep. 2017.

[29] N. Radosavljević, Dj. Babić, "Pregled sigurnosnih pretnji i mehanizama prevencije i zaštite u bežičnim senzorskim mrežama s primenom u preciznoj poljoprivredi," 19th International Symposium INFOTEH-JAHORINA, March 2020.

[30] N. Radosavljević, Dj. Babić, "Overview of security threats, prevention and protection mechanisms in wireless sensor networks," Journal of Mechatronics, Automation and Identification Technology, vol. 5, no. 2, pp. 1 – 6, 2020.

[31] C. Wang, T. Feng, J. Kim, G. Wang, W. Zhang, “Catching Packet Droppers and Modifiers in Wireless Sensor Networks,” presented at the 2009 6th Annual IEEE Communications Society Conference on Sensor, *Mesh and Ad Hoc Communications and Networks, Jun. 2009.*

[32] A. D. Wood, J. A. Stankovic, “Denial of service in sensor networks,” Computer, vol. 35, no. 10, pp. 54–62, Oct. 2002.

[33] C. Karlof, D. Wagner, "Secure routing in wireless sensor networks : Attacks and countermeasures," In Proceedings of the first international workshop on sensor network and protocols and applications, 2003.

[34] Y. Yu, R. Govindan, D. Estrin, "Geographical and Energy Aware Routing: a recursive data dissemination protocol for wireless sensor networks," UCLA Computer Science Department Technical Report 463, 2001.

[35] C. Tumrongwittayapak, R. Varakulsiripunth, "Detecting sinkhole attacks in wireless sensor networks," In ICROS-SICE international joint conference, pp. 1966–1971, 2009.

[36] N. Tanabe, E. Kohno, Y. Kakuda, "A Path Authenticating Method Using Bloom Filters against Impersonation Attacks on Relaying Nodes for Wireless Sensor Networks," presented at the 2013 IEEE 33rd International Conference on Distributed Computing Systems, *Workshops (ICDCSW), Jul. 2013.*

[37] J. Newsome, E. Shi, D. Song, A. Perrig, "The sybil attack in sensor networks," presented at the the third international symposium, 2004.

[38] L. Yao, L. Kang, F. Deng, J. Deng, G. Wu, "Protecting source-location privacy based on multirings in wireless sensor networks," Concurrency Computat.: Pract. Exper., vol. 27, no. 15, pp. 3863–3876, Jun. 2013.

[39] J. R. Ward, M. Younis, "A cross-layer traffic analysis countermeasure against adaptive attackers of Wireless Sensor Networks," presented at the MILCOM 2016 - 2016 IEEE Military Communications Conference (MILCOM), Nov. 2016.

[40] S. Alsemairi, M. Younis, "Adaptive packet combining to counter traffic analysis in wireless sensor networks," In IWCMC, pp. 337–342, 2015.

[41] S. Pavaimalar, G. ShenbagaMoorthy, "Detection of node capture attacks in wireless sensor networks," International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 2, no. 1, Jan., 2013.

[42] S. Naika, N. Shekokarb, "Conservation of energy in wireless sensor network by preventing denial of sleep attack," International Conference on Advanced Computing Technologies and Applications (ICACTA-2015), 2015.

[43] R. Singh, J. Singh, R. Singh, "Hello flood attack countermeasures in wireless sensor networks," International Journal of Computer Science and Mobile Applications, vol. 4, no. 5, pp. 1-9, May 2016.

[44] D. Adamy, "EW 102: A Second Course in Electronic Warfare," Artech, 2003.

[45] V. C. Manju, M. Sasikumar, "Mitigation Of Replay Attack In Wireless Sensor Network," Int. J. on Information Technology, vol. 5, 2014.

[46] N. Tanabe, E. Kohno, Y. Kakuda, "An Impersonation Attack Detection Method Using Bloom Filters and Dispersed Data Transmission for Wireless Sensor Networks," 2012 IEEE International Conference on Green Computing, 2012.

[47] D. Liu, P. Ning, "Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks," In Proceedings of ACM conference on computer and communications security, 2003, pp. 263–276.

[48] J. Deng, R. Han, S. Mishra, "Defending against path-based DoS attacks in wireless sensor networks," In Proceedings of 3rd ACM workshop on security of ad hoc and sensor networks (SASN'05), Alexandria, 2005.

[49] Y. Shen, S. Liu, Z. Zhang, "Detection of Hello Flood Attack Caused by Malicious Cluster Heads on LEACH Protocol," International Journal of Advancements in Computing Technology (IJACT), vol. 7, no. 2, March 2015.

[50] R. Pickholtz, D. Schilling, L. Milstein, "Theory of Spread-Spectrum Communications - A Tutorial," IEEE Trans. Commun., vol. 30, no. 5, pp. 855–884, May 1982.

[51] http://en.wikipedia.org/wiki/Ultrawideband., UWB-wikipedia. (pristupano 28.03.2020)

[52] I. Oppermann, L. Stoica, A. Rabbachin, Z. Shelby, J. Haapola, "UWB wireless sensor networks: UWEN - a practical example," IEEE Commun. Mag., vol. 42, no. 12, pp. S27–S32, Dec. 2004.

[53] B. Yu, B, Xiao, "Detecting selective forwarding attacks in wireless sensor networks," In Parallel and distributed processing symposium-20th international IPDPS, pp. 8, 2006.

[54] I. Khalil, S, Bagchi, N. B. Shroff, "Liteworp: A light-weight countermeasure for the wormhole attack in multi-hop wireless networks," In Proceedings of DSN, pp. 612–621, 2005.

[55] C. Chen, M. Song, G. Hsieh, "Intrusion detection of sinkhole attacks in large-scale wireless sensor networks," In IEEE international conference on wireless communications, networking and information security (WCNIS), 2010, pp. 711–716.

[56] T. Okazaki, E. Kohno, T. Ohta, Y. Kakuda, "A Multipath Routing Method with Dynamic ID for Reduction of Routing Load in Ad Hoc Networks," in Ad Hoc Networks, Springer Berlin Heidelberg, 2010, pp. 114–129.

[57] T. Okazaki, E. Kohno, Y. Kakuda, “Improvement of Assurance for Wireless Sensor Networks Using Packet Detouring and Dispersed Data Transmission,” presented at the 4th IEEE Int’l Conference on Cyber, Physical and Social Computing (CPSCom), Oct. 2011.

[58] H. Chan, A. Perrig, D. Song, "Random key predistribution schemes for sensor networks," In Proceedings of IEEE symposium on security and privacy, 2003, pp. 197–213.

[59] W. Du, J. Deng, Y. Han, S. Chen, P. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge," In Proceedings of IEEE INFOCOM’04, Hongkong, China, pp. 586–597, 2004.

[60] B. D. Ying, D. Makrakis, H. T. Mouftah, D. Ying et al., "Anti-traffic analysis attack for location privacy in WSNs," EURASIP Journal on Wireless Communications and Networking 2014, 2014.

[61] M. A. Matin, M. M. Islam, “Overview of Wireless Sensor Network,” in Wireless Sensor Networks - Technology and Protocols, InTech, 2012.

[62] J. Wilson, "Sensor Technology Handbook," Elsevier/Newnes: Burlington, MA, USA, 2005.

[63] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam,E. Cayirci, “Wireless sensor networks: a survey,” Computer Networks, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[64] G. Pachnanda, K. Singh, L. Gangwar, "Comparative analysis of A3, eecds and kneigh tree protocols in Wireless sensor networks," International Journal of Electronics and Computer Science Engineering, vol. 2, no. 3, pp. 987-991, 2013.

[65] P. M. Wightman, M. A. Labrador, “A3: A Topology Construction Algorithm for Wireless Sensor Networks,” presented at the IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference, 2008.

[66] Y. Liu, P. Geng, J. Yang, R. Chen, “Analysis and improvement of backbone-based topology control for wireless sensor networks,” JCM, vol. 19, no. 1, pp. 179–195, Jan. 2019.

[67] P. M. Wightman, M. A. Labrador, “A3Cov: A new topology construction protocol for connected area coverage in WSN,” presented at the 2011 IEEE Wireless Communications and Networking Conference (WCNC), Mar. 2011.

[68] M. Cardei, J. Wu, "Energy-efficient coverage problems in wireless ad-hoc sensor networks," Computer Communications, vol. 29, no. 4, pp. 413–420, 2006.

[69] R. Iyengar, K. Kar, S. Banerjee, "Low-coordination Topologies for Redundancy in Sensor Networks," In Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing, pp. 332–342, 2005.

[70] Z. Yuanyuan, X. Jia, H. Yanxiang, "Energy efficient distributed connected dominating sets construction in wireless sensor networks," presented at the Proceeding of the 2006 international conference, 2006.

[71] F. Dai, J. Wu, "An extended localized algorithm for connected dominating set formation in ad hoc wireless networks," IEEE Trans. Parallel Distrib. Syst., vol. 15, no. 10, pp. 908–920, Oct. 2004.

[72] G. Hatzivasilis, K. Fysarakis, I. Papaefstathiou, C. Manifavas, "A review of lightweight block ciphers," J Cryptogr Eng, vol. 8, no. 2, pp. 141–184, Apr. 2017.

[73] S. Heron, "Advanced Encryption Standard (AES)," Network Security, vol. 2009, no. 12, pp. 8–12, Dec. 2009.

[74] S. B. Sinaga, "MESSAGE SECURITY USING CRIPTOGRAPHY NOEKEON ALGORITHM." Unpublished, 2018, doi: 10.13140/RG.2.2.27168.28165.

[75] A. Bogdanov et al., "PRESENT: An Ultra-Lightweight Block Cipher," in Cryptographic Hardware and Embedded Systems - CHES 2007, Springer Berlin Heidelberg, pp. 450–466.

[76] J. Guo, T. Peyrin, A. Poschmann, M. J. B. Robshaw, "The LED Block Cipher, Cryptographic Hardware and Embedded Systems," CHES 2011, Springer, LNCS, 6917, pp. 326-341, 2011.

[77] G. Zhao, R. Li, L. Cheng, C. Li, B. Sun, "Differential fault analysis on LED using Super-Sbox," IET Information Security, IET, 2014, pp. 10.

[78] J. Guo, T. Peyrin, A. Poschmann, "The PHOTON Family of Lightweight Hash Functions," In P. Rogaway, editor, Crypto 2011, LNCS, vol. 6841, pp. 222-239, Springer, 2011.

[79] K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita, T. Shirai, "Piccolo: An Ultra-Lightweight Blockcipher," in Cryptographic Hardware and Embedded Systems – CHES 2011, Springer Berlin Heidelberg, 2011, pp. 342–357.

[80] G. Ramu, Z. Mishra, P. Singh, B. Acharya, "Performance optimised architectures of Piccolo block cipher for low resource IoT applications," IJHPSA, vol. 9, no. 1, p. 49, 2020.

[81] A. Mhaouch, W. Elhamzi, M. Atri, “Lightweight Hardware Architectures for the Piccolo Block Cipher in FPGA,” presented at the 2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), *Sep. 2020.*

[82] T. Suzaki, K. Minematsu, S. Morioka, E. Kobayashi, "Twine: A lightweight, versatile block cipher," ECRYPT Workshop on Lightweight Cryptography (LC11), pp. 146-169, 2011.

[83] C. D. Canniere, O. Dunkelman, M. Knezevic, "KATAN & KTANTAN - A Family of Small and Effcient Hardware-Oriented Block Ciphers," Cryptographic Hardware and Embedded Systems, CHES 2009, Springer, LNCS, 5747, 2009, pp. 272-288.

[84] J. Borghoff et al., “PRINCE – A Low-Latency Block Cipher for Pervasive Computing Applications,” in Advances in Cryptology – ASIACRYPT 2012, Springer Berlin Heidelberg, 2012, pp. 208–225.

[85] R. Beaulieu, S. Douglas, J. Smith, S. Treatman-Clark, B. Weeks, L. Wingers, "The SIMON and SPECK Families of Lightweight Block Ciphers," IACR Cryptology ePrint Archive, pp. 404, 2013.

[86] S. Banik, A. Bogdanov, & F. Regazzoni, "Exploring Energy Efficiency of Lightweight Block Ciphers," Selected Areas in Cryptography – SAC 2015, pp. 178–194, 2016.

[87] G. Upton, I. Cook, "A Dictionary of Statistics (2 rev. ed.)," Oxford University Press, 2014.

[88] http://www.csee.usf.edu/~mlabrador/Atarraya/ (pristupano 25.02.2020)

[89] M. Labrador, P. Wightman, "Topology Control in Wireless Sensor Networks," Springer Netherlands, 2009.