SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINARSKI RAD:

SIGURNOST BEŽIČNIH RAČUNALNIH MREŽA

Ivica Marić
Mentor: Prof. dr. sc. Leo Budin

ZAGREB, 2004

SADRŽAJ

1. UVOD

Eksplozivni rast bežičnih mreža u posljednje vrijeme podsjeća na rapidni rast Interneta u 90-im godinama prošlog stoljeća. Tome pogoduje i jednostavnost implementacije, fleksibilnost u radu te velik izbor uređaja koji se koriste pri implementaciji mreže(mrežne kartice, pristupne točke). Implementacijom bežične mreže također se umnogome smanjuju troškovi u usporedbi sa klasičnim rješenjima lokalne mreže. Zbog svih prednosti koje donose bežične mreže one su danas u širokoj uporabi u raznim javnim i privatnim organizacijama, a u zadnje vrijeme vidljiv je trend postavljanja tzv. vrućih točaka(*hot spot*) u kojima je dozvoljen besplatan pristup Internetu sa bilo kojim uređajem koji podržava bežične mreže.

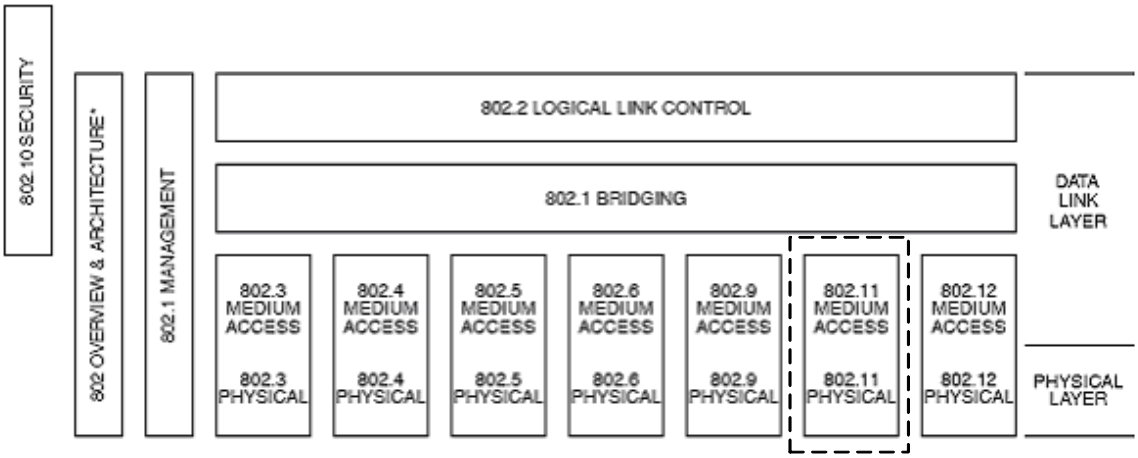
Iako su u standardima koji definiraju bežične računalne mreže navedeni razni elementi sigurnosti pokazuje se da ti elementi u većini slučajevi ostaju neiskorišteni što je, dakako, velik sigurnosni problem. No i kada se aktiviraju svi sigurnosni elementi to ne znači nužno da je postignuta odgovarajuća razina sigurnosti. Razlog tomu su mnogi nedostaci samog standarda koji su naknadno uočeni i koji omogućavaju zlonamjernoj osobi da bez većih poteškoća pristupi i koristi mrežne resurse bez dozvole i znanja vlasnika ili administratora mreže. Sami propusti u standardu obuhvaćaju propuste pri autentifikaciji korisnika mreže kao i propuste u enkripciji podataka između pristupne točke i korisnika. Valjano rješenje, barem u sadašnjem trenutku, se pronalazi u implementaciji VPN tehnologije zajedno sa troškom koje to donosi.

Cilj ovoga seminara je prikazati trenutno stanje u području sigurnosti bežičnih računalnih mreža kao i buduće smjernice u razvoju ovoga aktualnog područja.

2. OPĆENITO O BEŽIČNIM MREŽAMA

2.1 Standardi

Bežične mreže su definirane standardom 802.11 koji je donio IEEE (Institute of Electrical and Electronics Engineers) godine 1999. Standard definira najniža dva sloja OSI modela- fizički(PHY) i podatkovni(MAC) sloj. On je samo dio veće obitelji standarda koji definiraju lokalne(LAN) i gradske mreže(MAN). Obitelj 802 standarda se nalazi na slici 2.1.



Slika 2.1: Obitelj 802 standarda

Standardi 802.11a, 802.11b i 802.11g se razlikuju po fizičkom sloju(frekvencijama rada). Podatkovni sloj je jednak kod sva tri standarda i sastoji se od MAC (*Medium Access Control*) podsloja i LLC (*Logical Link Control*) podsloja. Sloj kontrole pristupa mediju(MAC) se malo razlikuje od takvog sloja u 802.3 standardu koji definira "žične" lokalne mreže, gdje se koristi CSMA/CD (*Carrier Sense Multiple Access/ Collision Detection*) protokol, po tome što se koristi CSMA/CA (*Carrier Sense Multiple Access/Collision Avoidance*) protokol. Razlog tome leži u prirodi medija kojim se vrši komunikacija koji ne omogućava da i pošiljatelj i primatelj istovremeno odašilju i primaju podatke. CSMA/CA je dio obitelji ALOHA protokola. Stanica koja želi poslati podatke prvo osluškuje medij i ukoliko je zauzet tj. netko već šalje podatke stanica poštuje to i povlači se. No ukoliko je medij slobodan određeno vrijeme(prema standardu naziva se DIFS - *Distributed Inter Frame Space*) tada stanica smije započeti odašiljati svoje podatke. Prijemna stanica će za svaki primljeni podatak, nakon što provjeri integritet primljenog paketa, poslati paket(ACK paket) kojim potvrđuje primitak valjanog paketa podataka. Kada odašiljač primi ACK paket znači da nije došlo do kolizije. Ukoliko odašiljač ne primi ACK paket znači da je došlo do kolizije, ili je paket oštećen stigao na odredište, pa je potrebno ponovno poslati paket.

2.1.1 802.11a standard

Fizički sloj ovog standarda definira rad na frekvenciji 5 GHz (frekvencija koja je po međunarodnim standardima dopuštena za korištenje bez posebnih dozvola i naknada) sa OFDM (*Orthogonal Frequency Division Multiplexing*) multipleksiranjem kanala. Standard

omogućava brzine od 6, 9, 12, 18, 24, 36, 48, i 54 Mbit/s. Iako mreže rađene po ovome standardu omogućavaju najveće brzine imaju jednu ogromnu manu - domet je ograničen na cca. 15m što je neprikladno za većinu korisnika i zbog toga nisu toliko popularne.

2.1.2 802.11b standard

Ovaj standard je danas dominirajući na tržištu ponajviše poradi relativno niske cijene implementacije i zadovoljavajućih performansi. Fizički sloj radi na frekvenciji od 2.4 GHz (također frekvencija slobodna za uporabu), koristi DSSS (*Direct Sequence Spread Spectrum*) tehnologiju za odašiljanje signala i omogućava maksimalnu propusnost od 11 Mbit/s. Razlog korištenja DSSS (*Direct Sequence Spread Spectrum*) tehnologije je velika pouzdanost i propusnost jer se koristi širi frekvencijski opseg. Svaka binarna "1" ili "0" se kodira u niz jedinica ili nula te se takvi nizovi šalju kroz sve frekvencijske pojase u frekvencijskom opsegu što značajno pridonosi pouzdanosti u slučajevima kada se pojavljuje interferencija sa drugim uređajima (na istoj frekvenciji rade i mikrovalne pećnice, bežični telefoni te Bluetooth). Ako se i izgubi dio poslanog niza još uvijek se može na prijemnoj strani odrediti koju vrijednost ima bit podatka.

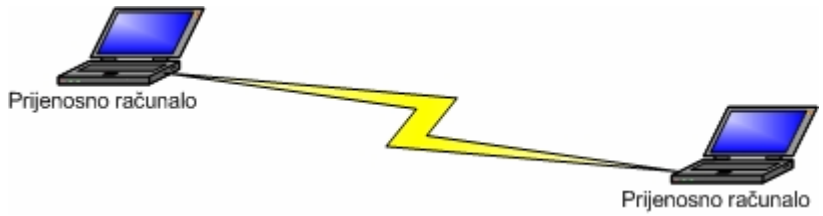
2.1.3 802.11g standard

Ovaj standard je omogućava maksimalnu propusnost od 54 Mbit/s (kao 802.11a) na frekvenciji od 2.4 Ghz (kao 802.11b). Bitno je naglasiti da je ovaj standard kompatibilan i sa 802.11a i sa 802.11b standardom. Fizički sloj(PHY) 802.11g standarda se naziva *Extended Rate PHY*(ERP). ERP podržava četiri različite modulacije: DSSS, OFDM, PBCC(*Packet Binary Convolutional Code*), DSSS-OFDM(hibridna modulacija u kojoj se preambula i zaglavlje moduliraju pomoću DSSS, a teret pomoću OFDM). ERP ima mogućnost detekcije korištene modulacije pri komunikaciji sa određenim klijentom. Podatkovni sloj je isti kao i kod 802.11a i 802.11b standarda.

2.2 Vrste bežičnih mreža

Postoje dva osnovna načina ostvarivanja bežičnih mreža. Odabir neke od njih ovisi o potrebama i mogućnostima korisnika.

Ad hoc mreže
Standard definira ovaj način povezivanja kao *Independent Basic Service Set* (IBSS).
Mreža ovoga tipa uspostavlja se direktno između dva ili više računala(slika 2.2). Ograničavajući faktor je ovdje to što sva umrežena računala moraju biti u relativno malom prostoru poradi male snage njihovih antena. Ovakav tip mreža se uglavnom ne koristi.

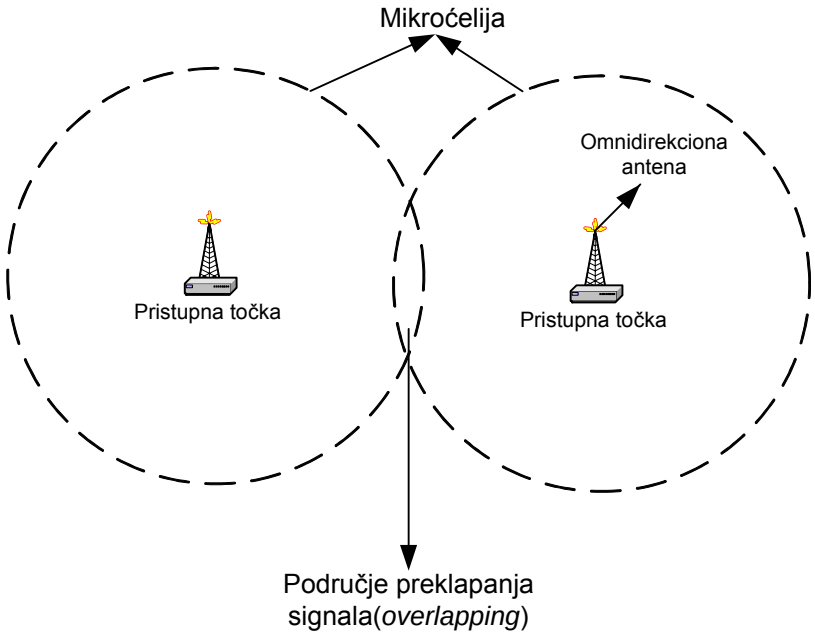


Slika 2.2: Primjer ad hoc mreže

Strukturirane mreže

Standard definira ovaj tip mreže kao *Basic Service Set* (BSS). U ovom načinu rada klijenti komuniciraju preko pristupnih točaka(*acces point*). Pristupne točke su uređaji preko kojih klijenti mogu dobiti pristup mreži(slika 2.4). Prednost ovoga rješenja leži u tome što dopušta veću fleksibilnost u radu kao i veće dosege samog signala te bolju kvalitetu.

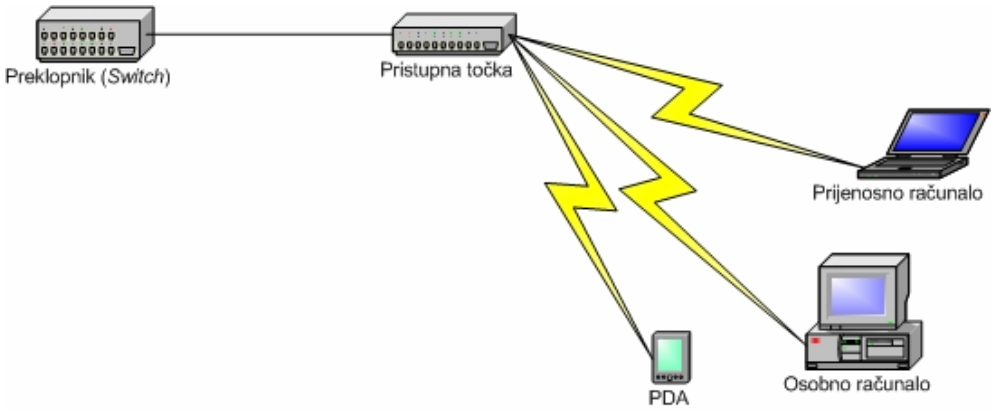
Osnovno područje rada pristupne točke je prostor koji je pokriven signalom, a često se naziva i mikroćelijom(slika 2.3). Taj prostor se može povećati dodavanjem drugih pristupnih točaka. Pristupna točka se pomoću prikladnih uređaja(preklopnik, koncentrator) povezuje na Ethernet i ona komunicira sa svim uređajima unutar svoje ćelije. Pristupna točka upravlja cijelim mrežnim prometom.



Slika 2.3: Prikaz mikroćelije i područja prekrivanja signalom

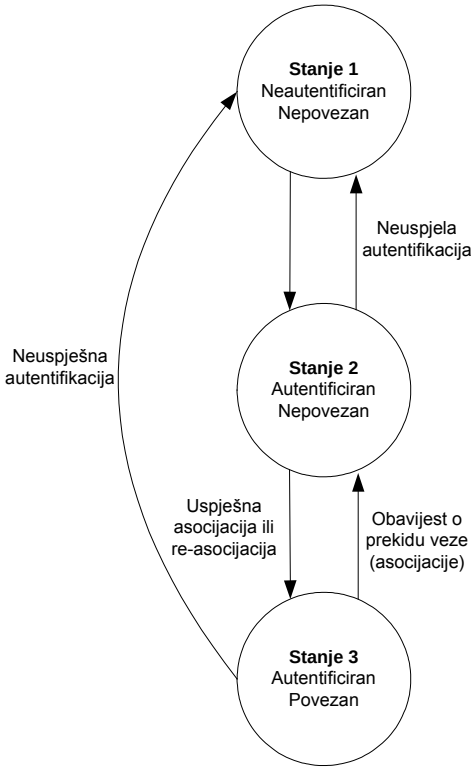
Ukoliko je potrebno proširiti područje pokrivanja može se dodati još pristupnih točaka čime nastaje prošireno područje rada. Preporučuje se da proširena područja uključuju 10-15% prekrivanja da bi korisnici bez gubljenja signala mogli prelaziti iz jedne u drugu ćeliju.

Za dobivanje najboljih performansi potrebno je osigurati da granične pristupne točke rade na drugačijim frekvencijskim pojasevima jer, u suprotnom, može doći do interferencije što degradira performanse u području preklapanja signala.



Slika 2.4: Infrastrukturni način

Klijent mora sa pristupnom točkom uspostaviti vezu da bi mogao biti član mreže. Proces pristupanja mreži može se prikazati konačnim automatom na slici 2.5.



Slika 2.5: Dijagram stanja pri spajanju klijenta na mrežu

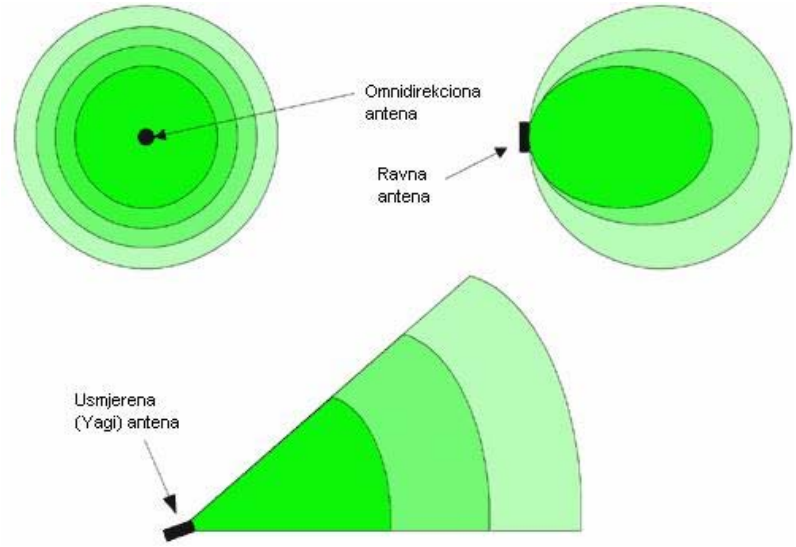
Za prelazak iz stanja u stanje klijent i pristupna točka izmjenjuju poruke koje se zovu upravljački okviri (*managment frames*). Sve pristupne točke u fiksnim vremenskim intervalima odašilju upravljački okvir (*beacon frame*) koji signalizira klijentima postojanje pristupne točke. Kada se klijent želi spojiti na mrežu on osluškuje signal(na svim frekvencijskim pojasevima) i čeka upravljačke okvire koje odašilju sve pristupne točke koje su mu u dometu. Tada klijent odabire kojoj se pristupnoj točci želi pridružiti. Nakon toga klijent i odabrana pristupna točka izmjenjuju nekoliko upravljačkih okvira i ulaze u proces pridruživanja. Postoje dva standardizirana načina autentifikacije korisnika: otvorena autentifikacija i autentifikacija dijeljenim ključem i bit će opisani kasnije. Nakon što klijent prođe autentifikaciju pomiče se u drugo stanje. Odašilje upravljački okvir kojim zahtjeva pridruživanje mreži(tj. ćeliji) i tek kada mu pristupna točka odgovori sa drugim upravljačkim okvirom on prelazi u treće stanje i konačno dobiva pristup mreži.

3. STANDARDIMA DEFINIRANA SIGURNOST BEŽIČNIH MREŽA

Iako standardi definiraju nekoliko sigurnosnih elemenata činjenica je da su bežične mreže najslabija sigurnosna karika unutar neke organizacije. Standardi ne uspijevaju zadovoljiti tri osnovna sigurnosna zahtjeva: pouzdana autentifikacija korisnika, zaštita privatnosti i autorizacija korisnika. Osnovni sigurnosni mehanizam je *Wired Equivalent Privacy* (WEP) i u njemu samome ima značajnih sigurnosnih propusta. Osim toga IEEE je ostavio bitne sigurnosne elemente kao raspodjelu ključeva i robusni način autentifikacije korisnika otvorenim pitanjima. Također većina organizacija koje imaju bežične mreže se oslanjaju na sigurnost definiranu standardima ili čak i ne koriste nikakve sigurnosne mjere.

3.1 Fizičko ograničavanje propagacije signala

Kako se računala u bežičnim mrežama povezuju Postoje tri vrste antena koje se koriste danas u mrežama: omnidirekciona, ravna i usmjerena (Yagi) antena. One pokrivaju različito područje i utječu na veličinu ćelija. Vrste antena i područje propagacije signala svake pojedine antene je prikazano na slici 3.1.



Slika 3.1: Vrste antena i područje propagacije signala

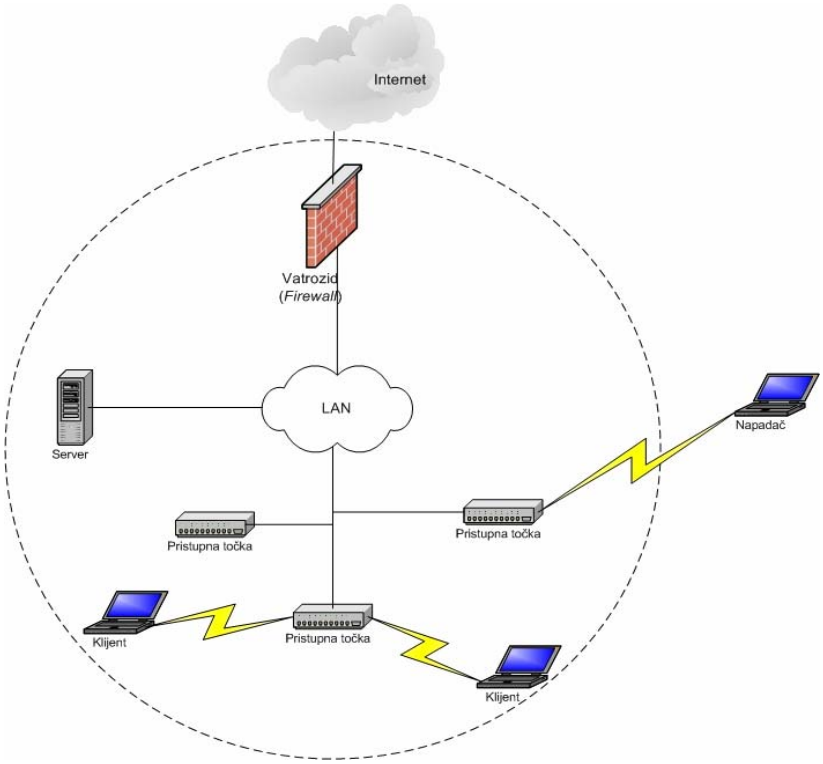
Prema standardima najveća snaga odašiljača ne smije prelaziti 36 dBm, a računa se pomoću izraza [3.1].

$$\textbf{EIRP} = P_S + A - P_G \qquad [3.1]$$

P_S – snaga predajnika
A – pojačanje antene
P_G – gubitak u kabelu koji povezuje pristupnu točku i antenu

gdje je EIRP (*Effective Isotropic Radiated Power)* efektivna snaga koju izrači antena kroz medij jednake gustoće u svim smjerovima. Posljedica ovoga je činjenica da pristupne točke imaju ograničeno područje pokrivanja(tj. ćelija ima ograničenu veličinu). Prilikom dizajniranja same bežične mreže unutar nekoga područja potrebno je provesti korjenit pregled područja i tako utvrditi optimalnu vrstu antena koje će se koristiti i dovoljnu snagu, vodeći računa o svim ograničenjima. Također se mora uzeti u obzir da je frekvencija, koju

koriste mreže po standardima 802.11b i 802.11g, od 2.4 GHz nelicencirana pa se može dogoditi da dođe do interferencije sa bežičnim telefonima(i drugim uređajima koji rade na istoj frekvenciji) što može dovesti do uskrate usluge (*Denial Of Service*). Također je dobro pretpostaviti da potencijalni napadač može imati bolju i osjetljiviju opremu nego što je propisana standardima što proširuje područje dosega mreže, a time i potencijalne opasnosti.
Ukoliko se ne vodi računa o gore navedenim stvarima moguće je da signal dopire i izvan fizičkih granica organizacije kojoj mreža pripada što otvara mogućnosti za tzv. napad s parkirališta (eng. *parking lot attack*). Ovaj napad je prikazan na slici 3.2.



Slika 3.2: Prikaz napada s parkirališta

3.2 Identifikator skupa usluga (SSID)

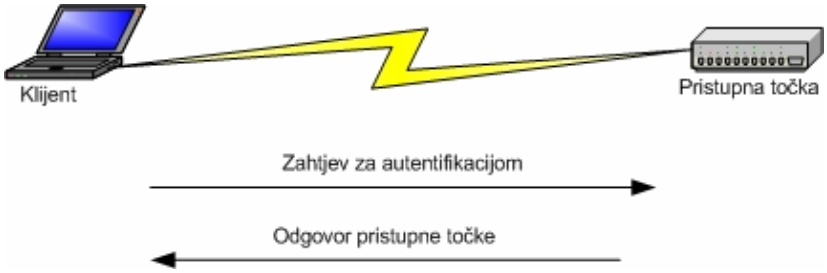
Standard definira i drugi način ograničavanja pristupa a to je identifikator skupa usluga (*Service Set Identifier - SSID*). On je zapravo ime mreže koju pokriva jedna ili više pristupnih točaka. U najčešće korištenom načinu pristupna točka odašilje SSID unutar signalnog upravljačkog okvira (*beacon*) te pomoću toga klijent može odlučiti kojoj će se mreži pridružiti.
U drugom načinu SSID se može iskoristiti kao sigurnosni faktor jer se pristupne točke mogu podesiti da ne odašilju SSID unutar kontrolnog okvira(*beacon frame*). Tada klijent koji se želi spojiti na mrežu mora imati isti SSID kao i mreža kojoj se želi pridružiti. Ukoliko klijent nema ispravan SSID tada pristupna točka odbacuje sve kontrolne okvire koje šalje klijent i tako on ne može proći postupak spajanja.
Iako teoretski izgleda kao dobar način kontrole pristupa u praksi ima značajnih problema. Naime kako se svi kontrolni okviri ne šalju u skrivenom(enkriptiranom) obliku napadač može osluškujući komunikaciju unutar mreže, točnije hvatajući kontrolne okvire koje odašilju sve pristupne točke u komunikaciji sa drugim valjanim korisnicima mreže, saznati SSID mreže i tako se neovlašteno pridružiti mreži.

3.3 Autentifikacija korisnika mreže

Kao što je ranije opisano klijent, da bi dobio pristup mreži, mora prvo proći proces autentifikacije. Standardi definiraju dva načina za provjeru korisnika: autentifikacija otvorenog sustava(*Open System Authentication*) i autentifikacija temeljena na dijeljenoj tajni(*Shared Key Authentication*). Slijedi opis svake od nabrojanih metoda.

3.3.1 Autentifikacija otvorenog sustava (Open System Authentication)

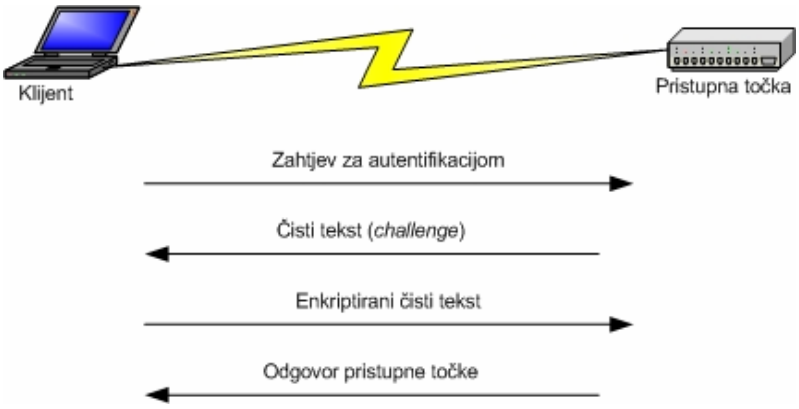
Ovaj način autentifikacije je podrazumijevani u standardu 802.11. Kako samo ime sugerira on dopušta pridruživanje mreži svakome tko to zatraži. Dakle on ne predstavlja nikakvu metodu autentifikacije. Prikazan je na slici 3.3.



Slika 3.3: Autentifikacija otvorenog sustava

3.3.2 Autentifikacija temeljena na dijeljenoj tajni (Shared Key Authentication)

Temelji se na činjenici da obje strane u procesu autentifikacije imaju jednak dijeljeni ključ (*Shared Key*). Pretpostavlja da je taj ključ prenesen klijentu i pristupnoj točci sigurnim kanalom. Pristupna točka šalje klijentu izazov(*challenge*) koji klijent enkriptira svojim tajnim ključem i šalje natrag pristupnoj točci. Pristupna točka dekriptira primljenu poruku sa svojim tajnim ključem, koji je isti kao i kod klijenta, te ukoliko se radi o istom tekstu koji je i poslala tada je klijent prošao proces autentifikacije te se može pridružiti mreži. Proces autentifikacije na ovaj način je prikazan na slici 3.4. Ukoliko klijent želi provjeriti pristupnu točku tada on čini isto samo u obrnutom smjeru.
Ovaj način autentifikacije se nikako ne preporučuje i smatra se da je bolje koristiti otvorenu kontrolu pristupa. Razlog tome je ponovno slanje upravljačkih okvira u nekriptiranom obliku preko nesigurnog medija. Naime napadač može uhvatiti upravljačke okvire sa čistim tekstom kao i sa enkriptiranim istim tekstom i na taj način doći do ključa koji se koristio. Detaljniji opis samoga napada se nalazi u slijedećem poglavlju.



Slika 3.4: Prikaz autentifikacije temeljene na dijeljenoj tajni

3.4 Wired Equivalent Privacy (WEP)

WEP je definiran u 802.11 standardu i nastoji ispuniti tri osnovne pretpostavke:

- Povjerljivost: temeljna svrha WEP-a je spriječiti prisluškivanje(*eavesdropping*) mrežnog prometa.
- Kontrola pristupa: također ima ulogu kontrole pristupa jer pristupne točke imaju mogućnost zabrane prometa klijentima koji se ne prođu uspješno proces autentifikacije.
- Integritet: dodatno polje u svakom okviru služi za provjeru integriteta samog okvira.

U sva tri slučaja snaga WEP-a se temelji na težini otkrivanja tajnog ključa pomoću napada čistom silom(*brute force attack*), no kako ćemo vidjeti kasnije ima puno bržih i učinkovitijih napada na WEP.

WEP se koristi na podatkovnom sloju OSI modela kako bi zaštitio podatke tijekom prijenosa. WEP se oslanja na tajnosti ključa koji se koristi između pristupne točke i klijenta i pomoću njega enkriptira tijela okvira poruke. Enkripcija se vrši u slijedećim koracima:

1. Zaštitno kodiranje (*checksumming*)

   Kako bi zaštitili integritet poruke nad njom se vrši operacija zaštitnog kodiranja sa CRC32 polinomom te se zaštita zapisuje na kraj podatka koji se želi zaštititi. Dakle čisti tekst dobivamo kao $P=\{M,c(M)\}$ gdje je M originalni podatak. Valja primijetiti da $c(M)$ pa tako i P ne ovisi o dijeljenom ključu k. Čisti tekst P je ulaz za drugi korak.

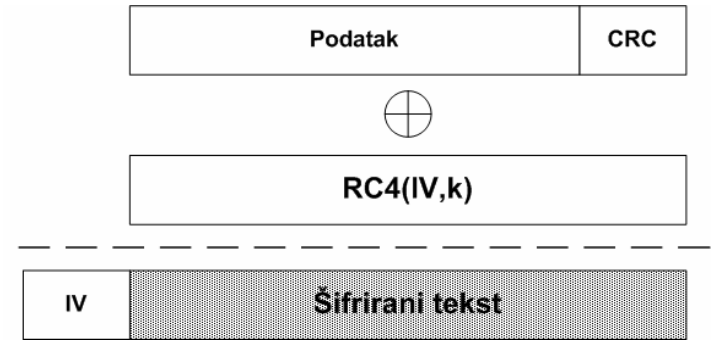
2. Enkripcija

   U drugom koraku enkriptiramo čisti tekst iz prethodnog koraka pomoću algoritma RC4. Na neki način(npr. slučajnim odabirom) biramo inicijalizacijski vektor IV koji uz ključ k služi kao ulaz u RC4 algoritam. Algoritam generira veliki broj pseudo-slučajnih bitova kao funkciju ključa k i inicijalizacijskog vektora IV. Ovaj niz bitova označava se sa RC(IV,k). Nakon toga se vrši operacija ekskluzivno-ili nad bitovima čistog teksta i dobivenim nizom pseudo-slučajnih bitova da bi se dobio šifrirani tekst(*ciphertext*). Dakle:

$$C = P \oplus RC(IV, k) \qquad [3.2]$$

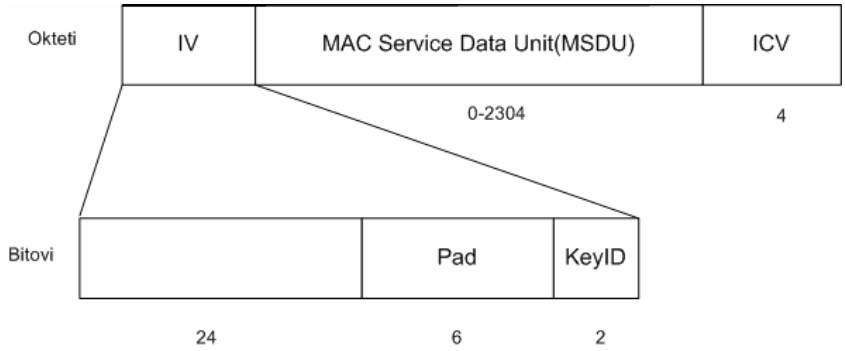
   Konačno, odašiljemo paket koji se sastoji od inicijalizacijskog vektora i šifriranog teksta preko bežične mreže.

Shematski, okvir je prikazan na slici 3.5.



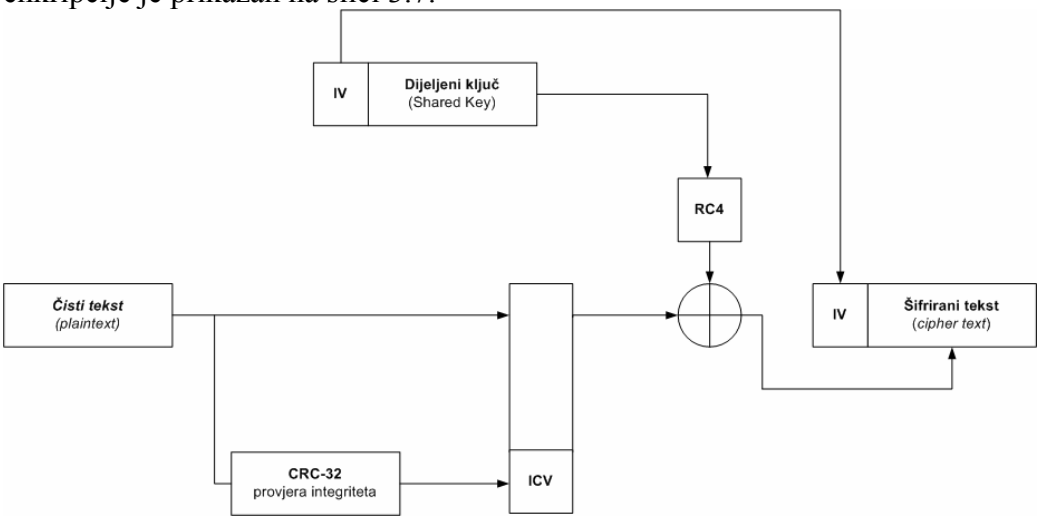
Slika 3.5: Shematski način dobivanja okvira

WEP okvir ima izgled prikazan na slici 3.6.



Slika 3.6: Shematski izgled okvira

Proces enkripcije je prikazan na slici 3.7.



Slika 3.7: Shematski prikaz standardne WEP enkripcije

CRC-32 (*Cyclic Redundancy Check*)
Ovaj algoritam izvorno služi za očuvanje integriteta podataka u komunikacijskom kanalu sa smetnjama i šumom. Njegova osnovica je, kako se u imenu i navodi, 32-bitni polinom koji se u heksadecimalnom obliku zapisuje kao `04C11DB7`. U WEP-u ovaj algoritam ima drugu, kriptografsku ulogu, i kao takav je vrlo loš izbor jer ne štiti u potpunosti integritet poruke(moguće je promijeniti određene bitove tako da se to ne detektira na prijemnoj strani). Puno bolji i prikladniji izbor bi bila jedna od funkcija kosanja(*hash function*) primjerice SHA-1 ili MD-5.

RC4

RC4 je najčešće korišten enkripcijski algoritam u softverskim aplikacijama. Dizajnirao ga je Ron Rivest 1987. godine i bio je poslovna tajna sve dok 1994. godine nije procurio njegov izvorni kod.

Sam algoritam se sastoji od dva dijela: algoritam za raspoređivanje ključeva(*Key Scheduling Algorithm-KSA*) i generator pseudo-slučajnih brojeva. Algoritam za raspoređivanje ključeva pretvara slučajno generirani ključ(obično veličine 40-256 bita) u početnu permutaciju *S* $\{0,\ldots 2^n-1\}$,gdje je *n* duljina riječi u bitovima, koju koristi generator pseudo-slučajnih brojeva kako bi proizveo pseudo-slučajan niz bitova na izlazu.

Generator pseudo-slučajnih brojeva inicijalizira dvije varijable *i* i *j* na 0, te tada u petlji izvršava četiri jednostavne operacije u kojima je *i* brojač dok se *j* povećava pseudo-slučajno, nakon toga u polju *S* zamjenjuje dvije vrijednosti na koje pokazuju *i* i *j* te kao izlaz daje vrijednost *S* na koju pokazuje *S[i]+S[j]*. Valja primijetiti da svaki član niza *S* biva najmanje jednom zamijenjen(moguća je zamjena sa samim sobom) i zbog toga se permutacija *S* dosta brzo se mijenja.

Algoritam za raspoređivanje ključeva se može prikazati slijedećim odsječkom u pseudo-kodu:

```
KSA(K){
      za(i=0;i<=N-1;i++)
         S[i]=i;
      j=0;
      za(i=0;i<=N-1;i++){
         j= j + S[i] + K[i mod length];
         Zamijeni(S[i], S[j]);
      }
}
```

Generator pseudo-slučajnih brojeva se također može prikazati slijedećim odsječkom u pseudo-kodu:

```
PRGA(K){
   za(i=0;i<=N-1;i++){
      j=0;
      i= i + 1;
      j= j + S[i];
      Zamijeni(S[i], S[j]);
      Izlaz=S[S[i] + S[j]];
   }
}
```

U slijedećem poglavlju će biti opisani sigurnosni propusti u RC4 algoritmu.

3.5 Upravljanje ključevima (*Key Managment*)

Ovaj nadasve bitan detalj nije definiran u standardu nego je njegovo rješavanje prepušteno na volju proizvođačima mrežne opreme. Rezultat toga je da je samo nekolicina najvećih proizvođača mrežne opreme ugradilo u svoje uređaje bilo kakav način upravljanja ključevima. Nažalost i ti proizvođači ne iznose dovoljno informacija o nivou sigurnosti koju su ugradili u svoje proizvode. Da stvari budu gore neki proizvođači u opisu svojih rješenja iznose da koriste protokole i metode sa dobro poznatim sigurnosnim propustima, primjerice Diffie-Hellman protokol koji je ranjiv na napad čovjek u sredini.

Standard definira dvije metode za korištenje WEP ključeva. Prva metoda dozvoljava prozor sa četiri ključa. Klijent ili pristupna točka mogu dekriptirati podatke koji su enkriptirani sa bilo kojim od ta četiri ključa. No sam prijenos podataka je ograničen na samo jedan od ta četiri ključa –standardni(*default*) ključ. Druga metoda je mapiranje

ključeva(*Key mapping method*). U ovoj metodi svaka jedinstvena MAC adresa može imati svoj ključ. Ključevi su pohranjeni u pristupnoj točki i broj različitih ključeva ovisi o kapacitetu pristupne točke. Odvojeni ključ za svaku MAC adresu nameće pitanje koliko često će se mijenjati ključevi jer sama promjena ključeva se mora vršiti ručnim unošenjem(jer je to jedini siguran način) kod svakog korisnika mreže što donosi nove probleme kako korisnicima tako i administratoru bežične mreže.

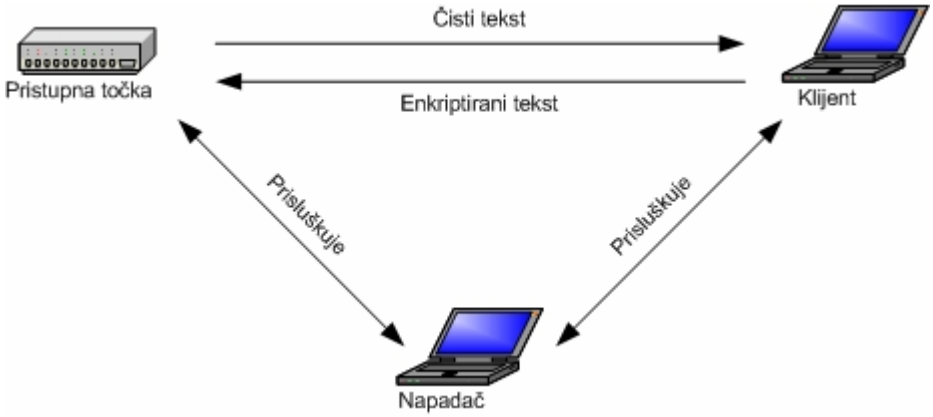
Ovdje je opisan standardni WEP sa 40-bitnim ključem. Veća duljina ključa je prema američkim zakonima zabranjena no neki proizvođači uvode svoje nadogradnje i proširuju ključ na 128-bita što smanjuje vjerojatnost uspješnog napada čistom silom(*brute force attack*) no ne pridonosi ukupnoj sigurnosti mreže jer se i dalje koristi 24 bita za inicijalizacijski vektor pa veličina ključa nije bitna ukoliko se upotrijebi dva puta isti inicijalizacijski vektor.

4. SIGURNOSNI PROPUSTI U STANDARDIMA

Prije nego se upustimo u raspravu o sigurnosnim propustima u standardu bitno je promotriti koliko je napad na bežičnu računalnu mrežu izvediv u praksi. Početni problem svakoga napada je doći do signala same mreže i tako izvesti aktivan ili pasivan napad. Da bi napadač bio u mogućnosti izvesti pasivan napad mora imati opremu koja je u mogućnosti osluškivati i presretati promet između pristupne točke i klijenta te je potrebno temeljito znanje fizičkog sloja definiranog 802.11 standardom. Za aktivni napad potrebno je imati i opremu koja je sposobna odašiljati podatke na mrežu. Oprema koja bi pouzdano obavljala navedene zadaće iziskivala bi znatna materijalna sredstva. Također postoji trend, posebno kod proizvođača bežične opreme, zanemariti napade na podatkovnom sloju smatrajući ih nepraktičnima i neizvedivima. Ovaj pristup je pogrešan iz dva razloga. Prvi je mogućnost postojanja napadača koji nije ograničen materijalnim resursima i vremenom tj. koji je u mogućnosti uložiti velika sredstva i svoje vrijeme da bi dobio pristup podacima. Kao primjer se može uzeti industrijska špijunaža koja je prilično profitabilan posao. Sjetimo se samo kako je procurio dio izvornog koda Windows-a 2000, a nedavno je Cisco objavio kako je ukraden njihov najnoviji operativni sustav za novu generaciju usmjernika(*router*). Drugo, potrebno sklopovlje za praćenje i aktivni napad dostupno je svima u obliku bežičnih kartica za stolna ili prijenosna računala. Postoje praktični pasivni napadi koji su izvedeni sa takvim karticama modificiranjem pogonskih programa(*drivers*). Primjerice PCMCIA kartica Orinoco tvrtke Lucent dopušta izmjenu pogonskih programa(reverznim inženjerstvom) na način da se može u mrežu ubacivati proizvoljan promet i time izvesti aktivan napad. Vrijeme uloženu u takav posao je netrivijalno ali to se samo jednom mora napraviti s obzirom da se tada gotovi upravljački programi mogu objaviti na Internetu i time postaju dostupni svima. Zbog toga razumno je pretpostaviti da dovoljno motiviran napadač može dobiti puni pristup podatkovnom sloju i u mogućnosti je obavljati pasivne ili aktivne napade.

4.1 Propusti u autentifikaciji korisnika

Načini provjere autentifikacije su objašnjeni u prethodnom poglavlju i ovdje se neće o tome raspravljati nego ćemo se usredotočiti na sigurnosne propuste. Autentifikacija otvorenog sustava(*Open System Authentication*) ne pruža nikakvu zaštitu pa o njoj ovdje nema potrebe raspravljati. No autentifikacija temeljena na dijeljenoj tajni bi trebala biti prepreka neovlaštenom pristupu. Kako je već objašnjeno u prethodnom poglavlju, klijent od pristupne točke dobiva u drugom koraku tekst koji treba enkriptirati vlastitim ključem te ga, u trećem koraku, poslati natrag pristupnoj točki. Ovaj način autentifikacije je ranjiv na napad čovjek u sredini(*man-in-the-middle attack*) koji je prikazan na slici 4.1. Naime napadač koji prisluškuje komunikaciju klijenta i pristupne točke može uhvatiti tekst koji pristupna točka šalje klijentu, te nakon toga i enkriptirani tekst koji klijent šalje pristupnoj točki. Došavši u posjed čistog i enkriptiranog teksta te inicijalizacijskog vektora napadač može dobiti pristup mreži.



Slika 4.1: Napad čovjek-u-sredini

4.2 Propusti u WEP-u

Pri komunikaciji klijenta i pristupne točke podaci se šalju u obliku okvira. Sami okviri nisu enkriptirani pa je napadač u mogućnosti doći do inicijalizacijskog vektora koji je korišten u enkripciji. Poznata zamka svih enkripcijskih algoritama koji rade sa tokom podataka(*stream ciphers*) je to da enkripcija dviju različitih poruka istim inicijalizacijskim vektorom daje informacije o samim porukama.
Dakle ako je:

$$C_1 = P_1 \oplus RC4(v,k) \quad [4.1]$$

$$C_2 = P_2 \oplus RC4(v,k) \quad [4.2]$$

tada je:

$$C_1 + C_2 = (P_1 \oplus RC4(v,k)) \oplus (P_2 \oplus RC4(v,k)) = P_1 \oplus P_2 \quad [4.3]$$

Drugim riječima provođenjem ekskluzivnog ILI na dva enkriptirana bloka poništava se efekt enkripcije i dobiva se rezultat istovjetan onome kao kada bi napravili ekskluzivno ILI nad porukama sa čistim tekstom. Zbog ovoga svojstva mogući su mnogi načini napada, a specijalan slučaj je kada je napadaču poznata jedna riječ čistog teksta, tada drugu riječ može automatski dobiti. Općenito stvarni čisti tekst ima dovoljno zalihosti kako bi napadač mogao otkriti P_1 i P_2 poznavajući samo $P_1 \oplus P_2$. Postoje mnoge klasične metode koje su primjenjive na ovaj slučaj. Također što je veći broj poznatih enkriptiranih riječi veća je i vjerojatnost da napadač otkrije podatke. Dakle da bi napad ovoga tipa uspio napadač mora imati podatke koji su enkriptirani istim inicijalizacijskim vektorom i mora barem djelomično poznavati čisti tekst. Kako se inicijalizacijski vektori ne enkriptiraju napadač može primjetiti kada se vektor ponovi i tako doći do podataka.
Rješenje problema se nalazi ili u izmjeni tajnog ključa nakon svakog okvira ili u izmjeni inicijalizacijskog vektora. Izmjena ključa nakon svakog okvira nije prihvatljiva pa WEP standard preporučuje(ali ne zahtjeva izričito) da se inicijalizacijski vektor mijenja sa svakim okvirom. Mnogi proizvođači mrežne opreme su slijedili preporuku i implementirali različite načine izmjene inicijalizacijskog vektora. Neki proizvođači su to učinili na veoma loš način. Primjerice većina PCMCIA bežičnih mrežnih kartica nakon svakoga pokretanja postavlja inicijalizacijski vektor na nultu vrijednost i zatim ga povećavaju za jedan nakon svakog odaslanog okvira. Dakle napadač ne mora doći u posjed svih inicijalizacijskih vektora nego je dovoljno da zna samo dio vektora sa početka i može doći do nekih podataka. Ponovno pokretanje kartice se događa svaki puta kada se ona umetne u prijenosno računalo ili se računalo pokrene što je prilično čest slučaj.

No da stvari bude gore sam WEP standard ima arhitektonski propust koji pogađa sve implementacije protokola, bez obzira koliko one pomno implementirane bile, i time izlaže korisnika ozbiljnoj opasnosti ponovne upotrebe ključa(*keystream reuse*). Naime polje u koje se zapisuje vrijednost inicijalizacijskog vektora je samo 24 bita široko i gotovo da jamči da će se isti inicijalizacijski vektor koristiti za više od jednog okvira. Dakle broj mogućih različitih vrijednosti inicijalizacijskog vektora je 2^{24}=16 777 216. To je prividno velik broj no uzevši u obzir da prosječna stanica koja odašilje okvire veličine 1500 byte-ova pri prosječnoj propusnosti od 5 Mbps (maksimalna propusnost je 11 Mbps) iscrpiti sve vektore za manje od pola dana. Prikažimo to slijedećim računom:

$$N = 2^{24} = 16\ 777\ 216$$
$$\mathrm{T} = 5\ 000\ 000\ \frac{\mathrm{bit}}{\mathrm{s}}$$
$$\mathrm{L} = 1500\ \mathrm{byte} = 12\ 000\ \mathrm{bita}$$

gdje je:

N – broj različitih inicijalizacijskih vektora
T – brzina veze
L – duljina okvira

dakle:

$$\mathrm{n} = \frac{\mathrm{T}}{\mathrm{L}} = \frac{5000000}{12000} \simeq 417\,\frac{okvira}{s} \qquad [4.4]$$

$$t = \frac{N}{n} = \frac{16\ 777\ 216}{417} \simeq 40233\ \mathrm{s} \simeq 11\ \mathrm{h} \qquad [4.5]$$

Dakle nakon već pola dana pristupna točka će morati nove okvire slati sa ponovljenim inicijalizacijskim vektorima što mrežu izlaže opisanim opasnostima.
Na ovaj problem značajan utjecaj ima i način odabira inicijalizacijskog vektora. Kako standard ne propisuje način na koji se treba mijenjati vektor, čak ni ne propisuje da se treba mijenjati, na savjesti proizvođača je hoće li i koju metodu odabrati. Postoje dva kod većine proizvođača prihvaćena načina odabira inicijalizacijskog vektora:

- slučajni odabir
  Vjerojatnost da će dva okvira imati isti inicijalizacijski vektor nakon *n* okvira je:

  $$P_2 = \frac{1}{24} \quad \text{za } n=2 \qquad [4.4]$$

  ili:

  $$\mathrm{P_n} = \frac{\mathrm{P_{n-1}} + (n-1)(1-\mathrm{P_{n-1}})}{24} \quad \text{za } n>2 \qquad [4.5]$$

  Dakle po ovim formulama se vidi da postoji 50% šanse da dva okvira imaju isti inicijalizacijski vektor već nakon 4823 odaslana okvira! Također postoji 99% šanse da dva okvira imaju isti inicijalizacijski vektor nakon 12 430 odaslanih okvira. Ovaj broj okvira izmjene jedna pristupna točka i jedan klijent za otprilike 11 minuta. Ovo je poznato kao i rođendanski paradoks – kaže da ako skupimo 23 ljudi u jednoj sobi tada je vjerojatnost da dvoje ili više ljudi imaju rođendan na isti dan 50%.

- inkrementiranje za 1 nakon svakog odaslanog okvira
  Kako je već spomenuto ova metoda se koristi kod većine bežičnih kartica za prijenosna računala. Neki proizvođači kreću od 0 dok neki uzimaju konstantnu vrijednost. Vjerojatnost kolizije je 100% nakon što dva uređaja koja koriste ovu metodu započnu odašiljati okvire.

Jednom kada su otkrivena dva okvira sa istim inicijalizacijskim vektorom postoji mnogo metoda koje mogu poslužiti za otkrivanje podataka. Najjednostavniji slučaj je kada nam je tekst jedne poruke poznat i automatski možemo dobiti drugu. Postoji mnogo načina otkrivanja prikladnih kandidata za poznati, čisti, tekst. Primjerice, mnogi protokoli kojima se koristimo na Internetu(npr. TCP, IP) imaju dobro definirana i predvidljiva polja. Kao primjer možemo uzeti i način prijave korisnika na sustav koji je uglavnom jednak: nakon pozdravne poruke od korisnika se traži login: (identifikator korisnika) i password: (lozinku) i to su uvjerljivi kandidati za poznati tekst.

4.2.1 Napadi na WEP

Postoji više vrsta napada na WEP. Postoje dvije osnovne vrste napada:

- pasivni napadi
  U ovoj vrsti napada napadač samo prisluškuje komunikaciju korisnika sa mrežom i ni na koji način ne utječe na podatke koje razmjenjuju pristupna točka i klijenti.
- aktivni napadi
  Napadač aktivno utječe na promet na mreži. On to može činiti na više načina primjerice može ubacivati svoje podatke, lažirati komunikaciju klijenta i pristupne točke, zagušivati promet na mreži, neovlašteno koristiti mrežne resurse. Aktivni napadi su općenito zahtjevniji za napadača jer mora uložiti veći trud, više vremena i materijalnih sredstava nego što bi trebao za pasivni napad.

PASIVNI NAPADI

Analiza prometa
Ovo je najjednostavniji pasivni napad i sastoji se od prisluškivanja mreže s ciljem praćenja broja i veličine paketa u mreže. Za ovu vrstu napada napadaču je potrebna zadovoljavajuća antena, mrežna kartica koja radi u modu za slušanje (dakle ne odašilje nikakve pakete) i programska podrška koja će vršiti analizu veličine i broja paketa. Ovim napadom napadač može saznati tri osnovne informacije: količinu prometa u mreži, fizičku lokaciju pristupnih točaka te vrste protokola koji se koriste na mreži. Pojava naglog povećanja prometa na mreži može poslužiti kao indikator nekog bitnog događaja. Uz usmjerenu antenu(Yagi antena) i u kombinaciji sa GPS(*Global Positioning System*) sustavom napadač metodom triangulacije može doći do fizičke lokacije pristupne točke ili centra bežične mreže. Informaciju o vrsti protokola napadač može dobiti brojeći pakete u vremenskom intervalu. Najbolji primjer je TCP(*Transmission Control Protocol*). Ovaj protokol sinkronizira komunikaciju između krajnjih točaka odašiljući tri paketa. Prvo onaj koji šalje odašilje SYN paket onome s kime želi komunicirati. Tada prijemna strana odašilje paket SYNACK. Nakon toga prva strana šalje ACK paket i time komunikacija može otpočeti. Napadač može primijetiti takav uzorak te dobiti za daljnje napade bitnu informaciju o protokolima koji se koriste.

Pasivno prisluškivanje
U ovom napadu napadač također samo osluškuje mrežu. Jedini uvjet za uspješan napad ovoga tipa je pristup signalu mreže. Ovdje dolazi do izražaja koliko je mreža fizički zaštićena tj. koliko se vodilo računa o rasprostiranju signala pristupnih točki u prostoru. No čak ako je i mreža, fizički, dobro dizajnirana moguće je da napadač ima bolju opremu nego što standard nalaže i tako uspije dobiti pristup mreži. Standardni scenarij ide tako da napadač osluškuje mrežu i čeka da se ponovi isti inicijalizacijski vektor te tako, na ranije

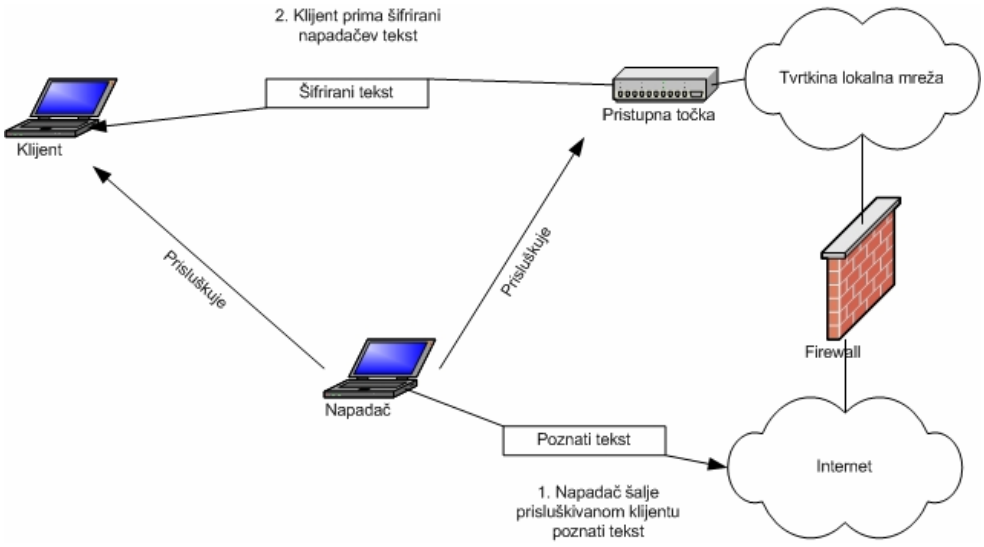
opisan način, dolazi do $P_1 \oplus P_2$. Nakon toga napadač, ukoliko mu je poznata jedna riječ iz para P_1, P_2 može odmah doći do druge poruke. Ukoliko napadač ne zna niti jednu poruku tada može, koristeći ranije dobivene informacije o protokolu, pretpostaviti neke konstantne dijelove poruka i tako doći do podataka. Uzmimo primjer da se u mreži koristi TCP/IP protokol. Zaglavlje IP protokola ima na fiksnoj udaljenosti od početka paketa zapisane u fiksnoj duljini IP adresu izvora i odredišta koje napadač može poznavati i tako dobiti dio poruke kojim se može poslužiti u otkrivanju cijele poruke. TCP protokol, također, ima na točno određenom mjestu zapisane izvorišni i odredišni port na koji se spaja i koji je točno poznat za svaku uslugu(npr. web poslužitelj se javlja na portu 80, news na 119, e-mail poslužitelj na portu 25 itd.) te i to iskoristiti u otkrivanju podataka. Isti se princip može primijeniti i na zaglavlja raznih aplikacija koje imaju dobro definiran oblik(npr. HTTP protokol koji se koristi na Internetu).
Napadač također može napraviti bazu podataka koja se sastoji od parova $(IV, C_1 \oplus C_2)$ tako da svaki puta kada se upotrijebi isti ključ može doći automatski do $P_1 \oplus P_2$ i daljnjim postupcima analize na kraju do P_1 ili P_2.

AKTIVNI NAPADI

Napad ponavljanjem inicijalizacijskog vektora (Initialization Vector Replay Attacks)
Napad ponavljanjem inicijalizacijskog vektora je praktično izveden napad. Jedan od mogućih scenarija je slijedeći:

- Napadač preko Interneta pošalje poruku(npr. e-mail) klijentu koga želi napasti.
- Napadač zatim pažljivo prisluškuje mrežu i čeka da pristupna točka pošalje klijentu poruku sa poznatim tekstom.
- Napadač će sada maknuti enkripcijsku zaštitu jer ima poznat inicijalizacijski vektor i poruku koja je enkriptirana.

Sada napadač može dodavati svoje podatke u enkriptirani paket te ga ponovno može enkriptirati.
Osnovna pretpostavka ovoga napada je da se inicijalizacijski vektor i WEP ključ mogu neprestano ponavljati dokle god mreža ne prihvati da je to ispravan paket podataka.
Napad je prikazan na slici 4.2.

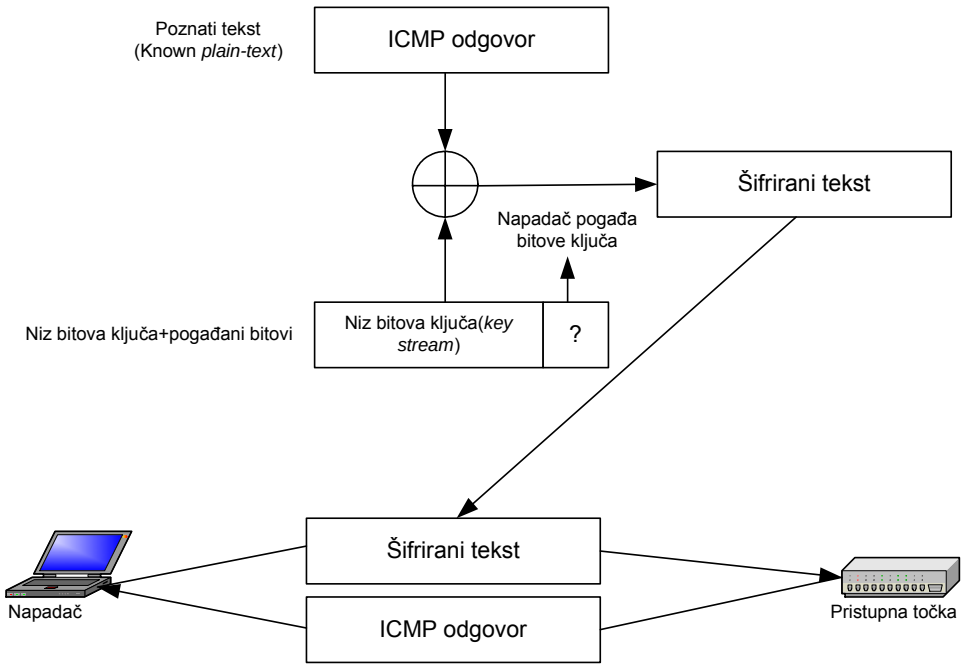


Slika 4.2: Napad ponavljanjem inicijalizacijskog vektora

Jednom kada je napadač dobio niz bitova kojim je enkriptiran paket(*keystream*) on može taj niz primijeniti na druge podatke koje će sam ubaciti u mrežu. Sam proces proširivanja ključa ima nekoliko koraka:

- Napadač može izgraditi paket tako da njegovu dosadašnju veličinu poveća za jedan oktet. Idealni kandidat za to je ICMP odgovor.
- Napadač tada povećava niz bitova ključa za jedan bit.
- Vrijednost bitova dodatnog okteta se pogađaju ali to nije problem jer je samo 256 mogućih vrijednosti.
- Kada napadač pogodi ispravnu vrijednost okteta on dobiva odgovor na ICMP paket koji je poslao.

Napadač nastavlja ovaj postupak dok god ne dobije niz bitova ključa željene veličine.
Prethodno opisan postupak se može shematski prikazati slikom 4.3.



Slika 4.3: Napad proširivanjem ključa

Napad obrtanjem bitova podataka(*Bit-Flipping Attacks*)
Ovaj napad ima isti cilj kao i prethodni samo što se u ostvarivanju cilja služi drugom metodom. Naime ova vrsta aktivnog napada iskorištava slabost vektora integriteta poruke(*ICV*). Iako veličina podatka koji enkriptirani paket nosi može varirati, mnogo elemenata se nalazi na konstantnom mjestu unutar paketa. Napad se može opisati u nekoliko točaka:

- Napadač prisluškuje okvire na mreži
- Pokupi jedan okvir s mreže i slučajnim odabirom zamijeni vrijednosti bitova(proizvoljan broj) unutar polja koje sadrži teret.
- Mijenja sadržaj polja u kojem se nalazi vektor integriteta poruke(*ICV*).
- Napadač šalje izmijenjeni paket na mrežu.
- Prijemna strana (klijent ili pristupna točka) prima paket i računa vektor integriteta poruke na osnovu podataka koji se nalaze u paketu. Prijemna strana tada uspoređuje izračunatu i dobivenu vrijednost vektora integriteta poruke(koja je u polju ICV paketa). Ukoliko su ta dva vektora ista, prihvaća izmijenjeni paket.

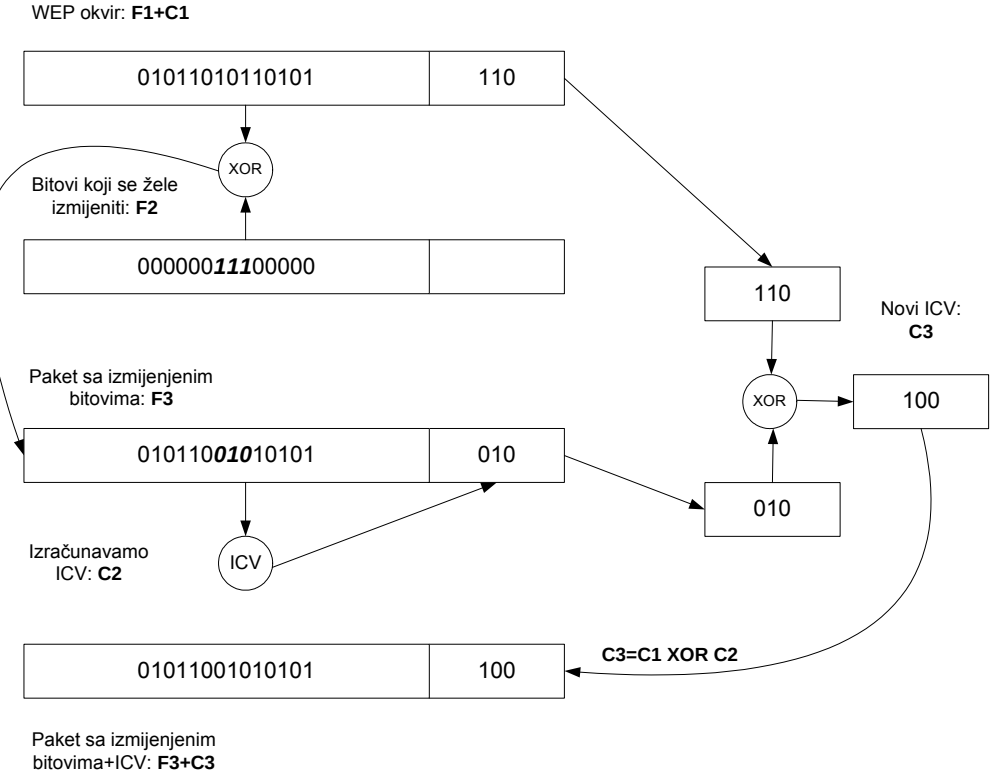
Prijemna strana tada de-enkapsulira paket i predaje ga višem, trećem, sloju OSI modela. Pošto je napadač zamijenio bitove paketa za treći sloj provjera integriteta na tome sloju ne uspijeva. IP stog tada generira predvidljivi izvještaj o greški.

- Napadač prisluškuje promet na mreži čekajući predvidljivi enkriptirani odgovor.
- Nakon što prepozna i primi odgovor napadač dolazi u posjed niza bitova ključa i može ga iskoristiti za prethodno opisan napad.

Uspjeh ovoga napada se temelji na propustu vektora integriteta. Ovaj vektor je u enkriptiranom dijelu paketa pa kako napadač može uspješno izmijeniti vrijednost bitova?

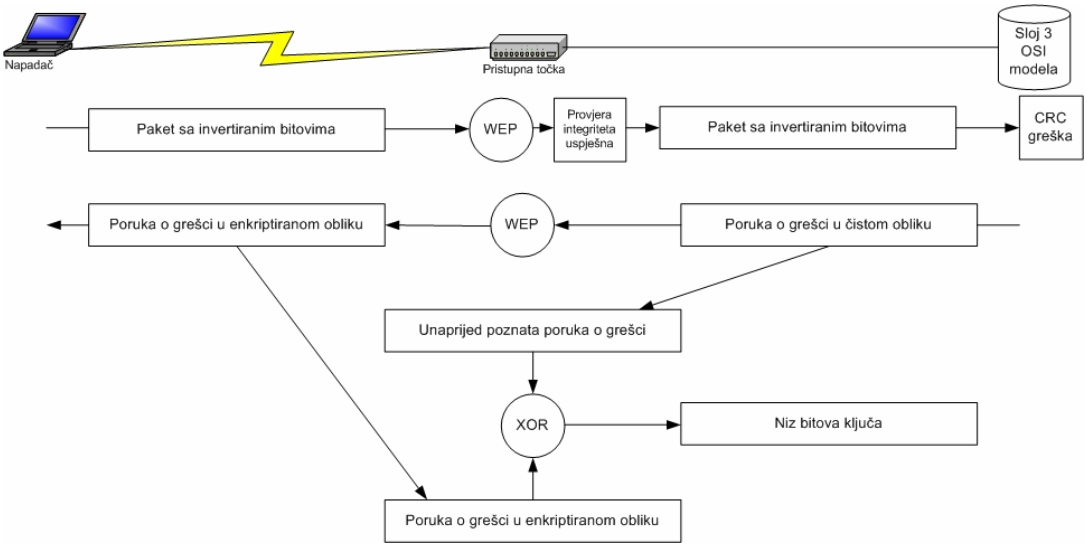
- Napadač uhvati jedan paket i želi mu izmijeniti ICV(C1).
- Generira paket jednake duljine sa postavljenim bitovima(F2)
- Treći paket se dobije ekskluzivnim ili nad prva dva paketa(F3=F1 XOR F2)
- Napadač računa ICV za treći paket(C2)
- Vektor integriteta koji će se umetnuti dobiva se ekskluzivnim ili nad vektorima primljenog i paketa koji se dobio ekskluzivnim ili generiranog i originalnog paketa(C3=C1 XOR C2).

Shematski prikaz prethodno opisanog postupka je prikazan na slici 4.4.



Slika 4.4: Prikaz zamijene bitova

Sam napad je prikazan na slici 4.5.



Slika 4.5: Napad obrtanjem bitova podataka(*Bit-Flipping attacks*)

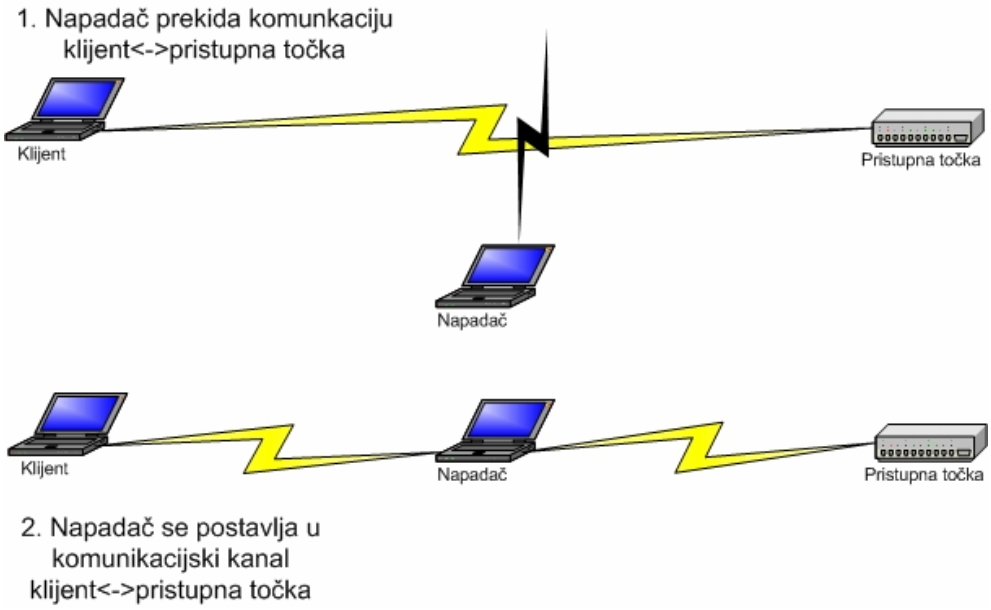
Napad čovjek-u-sredini(*Man-in-the-middle attack*)

Ovaj napad može biti iskorišten kako bi napadač pročitao ili modificirao podatke. Oslanja se na propust u standardu koji ne omogućava obostranu autentifikaciju klijenta i pristupne točke. Glavna zamisao napada je da se napadač postavi u komunikacijski kanal između klijenta i pristupne točke i presreće njihovu komunikaciju. Napad se provodi u nekoliko koraka:

- Napadač prekida komunikaciju klijenta i pristupne točke i ne dopušta klijentu da ponovno uspostavi vezu sa pristupnom točkom.

- Klijent nastoji uspostaviti vezu sa pristupnom točkom, ali nije u mogućnosti to obaviti pa uspostavlja vezu sa napadačevim računalom koje glumi pristupnu točku. Također u ovom koraku se napadač predstavlja pravoj pristupnoj točki kao klijent i uspostavlja vezu s njom. Na ovaj način napadač uspostavlja dva enkriptirana tunela: napadač-klijent i napadač-pristupna točka.

Napad je prikazan na slici 4.6.



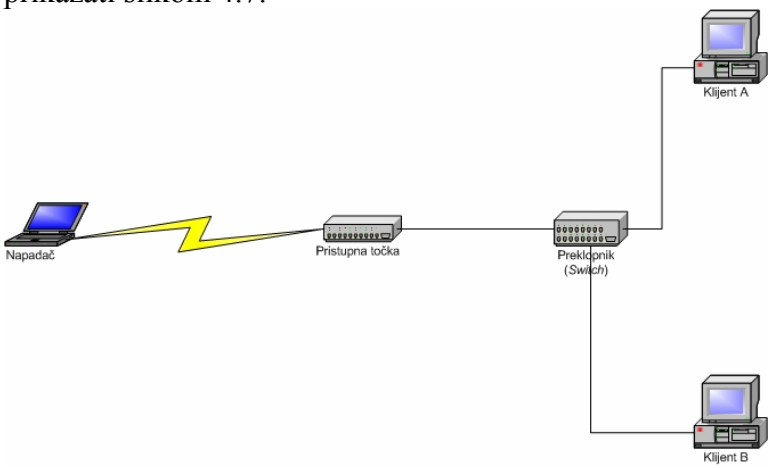
Slika 4.6: Napad čovjek-u-sredini

ARP napadi

Ova vrsta napada je podskup prethodno opisanog napada. Izdvojena je samo zato što se može iskoristiti i protiv računala koja nisu na bežičnoj mreži i jer napadač ne mora uspostaviti vezu sa klijentom nego je dovoljno da se lažno predstavi pristupnoj točki i time dobije pristup mreži. Uloga ARP-a(*Address Resolution Protocol*) je prevođenje fizičke adrese klijenta koja se koristi u drugom sloju OSI modela u IP adresu koja se koristi na trećem sloju OSI modela. Promjena načina prevođenja MAC adrese u IP adresu napadaču dozvoljava da mrežni promet prema nekom računalu usmjeri preko svoga računala. To napadaču dozvoljava da čita tuđe podatke, mijenja ih ili sprema pakete kako bi ih kasnije dekodirao.

Za uspješan napad ovoga tipa napadač mora imati pristup mreži. Napadač šalje krivotvoreni odgovor na APR upit i na taj način mijenja način na koji se do tada povezivala određena MAC sa IP adresom. Dakle napadač nije promijenio MAC adresu nego samo način na koji se ona prevodi u IP adresu. Jednom kada je to napravio napadač se nalazi u sredini komunikacije između dva klijenta i može utjecati na komunikaciju.

Napad se može prikazati slikom 4.7.



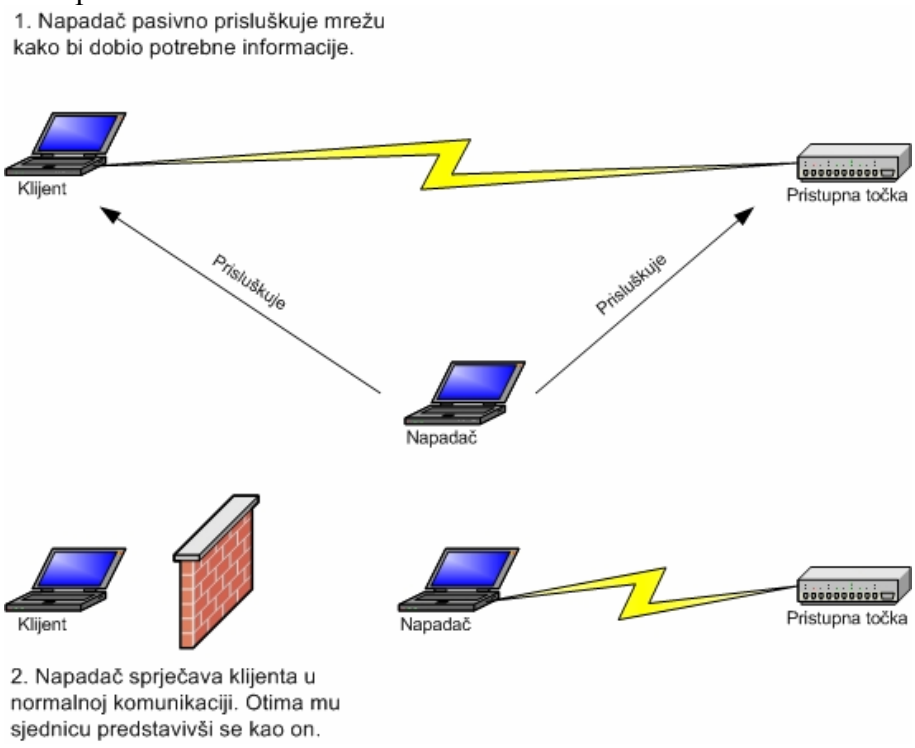
Slika 4.7: ARP napad

Krađa sjednice(Session Hi-jacking)

Krađa sjednice je napad koji se usmjeren protiv integriteta sjednice između korisnika i pristupne točke. Napadač može ukrasti sjednicu autentificiranom i autoriziranom korisniku mreže. Meta zna da je izgubila sjednicu ali ne zna da je njezinu sjednicu preuzeo napadač i meti se to čini kao normalni ispad bežične mreže. Jednom kada je napadač uspio ukrasti klijentovu sjednicu on može nastaviti raditi u mreži proizvoljno dugo. Za uspješan napad ovoga tipa potrebna su dva uvjeta:

- Prvo se mora prikazati mreži kao meta da bi ga mreža uopće prihvatila. To uključuje krivotvorenje paketa višeg sloja, korištenje metode autentifikacije koju mreža koristi te primjenu zaštitne enkripcije ako mreže to zahtijeva. Ovim radnjama najčešće prethodni pasivni napad prisluškivanjem kako bi napadač doznao potrebne informacije.
- Druga potrebna radnja je sprječavanje mete u komunikaciji sa pristupnom točkom. Napadač ovu zadaću obavlja slanjem lažiranih kontrolnih okvira koji meti signaliziraju prekid trenutne sjednice.

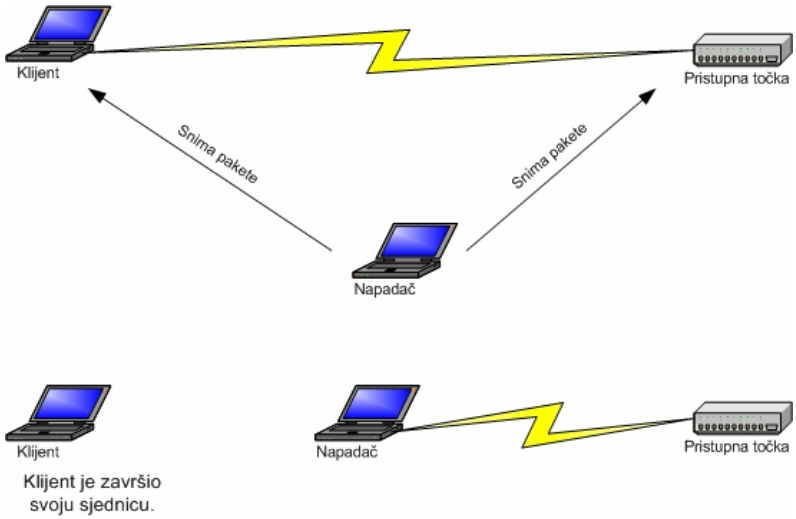
Napad se može prikazati slikom 4.8.



Slika 4.8: Prikaz krađe sjednice

Napad ponavljanjem paketa(*Packet Re-play Attack*)

Napad ponavljanjem paketa je, također, usmjeren na povredu integriteta informacija na mreži. Ovaj napad se koristi kako bi napadač dobio pristup mreži, ali za razliku od prethodnog, ničim se ne utječe na sjednice koje su u tijeku. Napad se ne odvija u realnom vremenu nego se događa nakon što je klijent završio svoju sjednicu. Napadač snima sjednicu između klijenta i pristupne točke ili više takvih sjednica kako bi ih kasnije iskoristio. Kada klijent završi svoju sjednicu napadač ponavlja njegove pakete i tako dobiva pristup mreži. Bez daljnjih sigurnosnih prepreka napadač može koristiti sve ovlasti klijenta čiju je sjednicu snimio. Čak iako napadač ne može zaobići enkripciju koja se koristi na mreži on je u mogućnosti modificirati pakete kako bi oštetio integritet podataka.
Napad se može prikazati slikom 4.9.



Slika 4.9: Napad ponavljanjem paketa

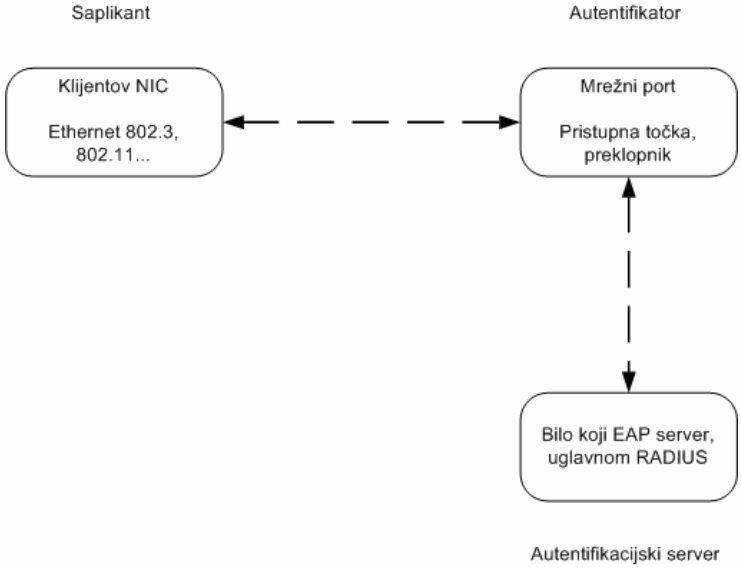
Kako se vidi postoji velik broj napada na WEP(od kojih su neki i u praksi uspješno izvedeni) i to samo govori u prilog činjenici da je WEP, a time i standard koji ga definira, krajnje nesiguran i kao takav bi što prije trebao biti zamijenjen sa nekim sigurnijim i boljim standardom koji bi u potpunosti uklonio navedene propuste.

5. SIGURNOSNE NADOGRADNJE 802.11 STANDARDA

Kako je prikazano u prethodnom poglavlju postojeći standard ne pruža kvalitetnu zaštitu za korisnike bežičnih mreža. Zbog toga je IEEE uočivši propuste u standardu započelo rad na novim prijedlozima i rješenjima koja bi učinili bežične mreže sigurnijima. Plod toga rada je i 802.1X standard koji nastoji poboljšati sigurnost.

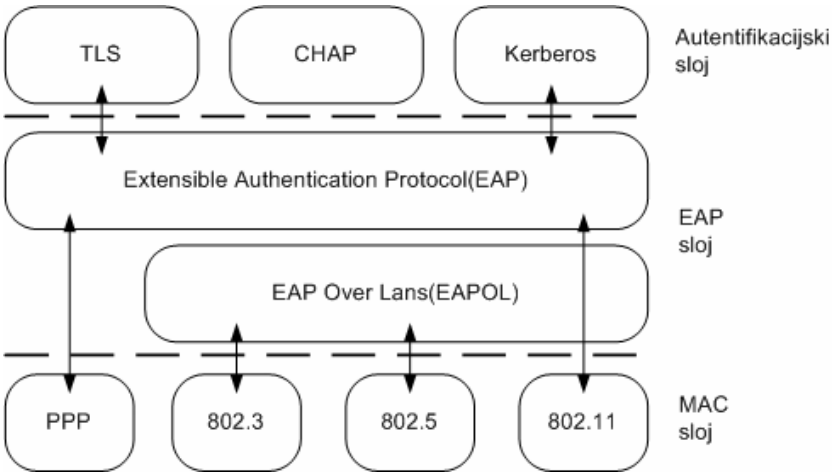
5.1 802.1X standard

Kako je navedeno IEEE je ovime nastojao pružiti korisnicima bežičnih mreža bolju sigurnost prvenstveno kroz bolju autentifikaciju korisnika mreže što rješava dobar dio trenutnih sigurnosnih problema. 802.1X standard nastoji omogućiti pristup mreži samo pravim korisnicima mreže preko boljeg sustava autentifikacije. 802.1X radi na MAC podsloju drugog sloja OSI modela. Pridruživanje mreži izvedeno je preko portova. U standardu port ima značenje združivanja(*association*) klijenta i pristupne točke.
Standard 802.1X pruža arhitektonsku okosnicu(*framework*) nad kojom korisnici mogu koristiti razne metode autentifikacije npr. autentifikacija certifikatima, pametnim karticama, jednokratnim lozinkama. Pruža pristup mreži baziran na portovima(*port-based*) za mrežne tehnologije kao što su Token Ring, FDDI, 802.11, 802.3 LAN. 802.1X pruža sigurnosnu okosnicu apstrahirajući tri osnovna entiteta: *supplicant*, autentifikator ili mrežni port te autentifikacijski poslužitelj. *Supplicant* je entitet koji koristi usluge autentifikatora koje mu on nudi preko portova. Autentifikator može biti preklopnik ili pristupna točka. *Supplicant* se autentificira preko autentifikatora autentifikacijskom poslužitelju koji tada nalaže autentifikatoru da dozvoli pristup *supplicant*u mreži. Pretpostavka je da svi autentifikatori komuniciraju sa istim, centralnim, autentifikacijskim poslužiteljom. U praksi se taj poslužitelj može, radi rasterećenja, nalaziti fizički na više lokacija no u logičkom smislu on je samo jedan.
Tri entiteta u 802.1X standardu se mogu prikazati slikom 5.1.



Slika 5.1: 802.1X entiteti

802.1X standard koristi EAP(*Extensible Autenthication Protocol*) kao podlogu za široku lepezu autentifikacijskih mehanizama. EAP je izgrađen oko izazov-odgovor(*challenge-response*) paradigme. EAP je prvotno namijenjen korištenju u "žičanim" mrežama, no kasnije je implementiran za korištenje u bežičnim mrežama.
EAP stog je shematski prikazan na slici 5.2.



Slika 5.2: EAP stog

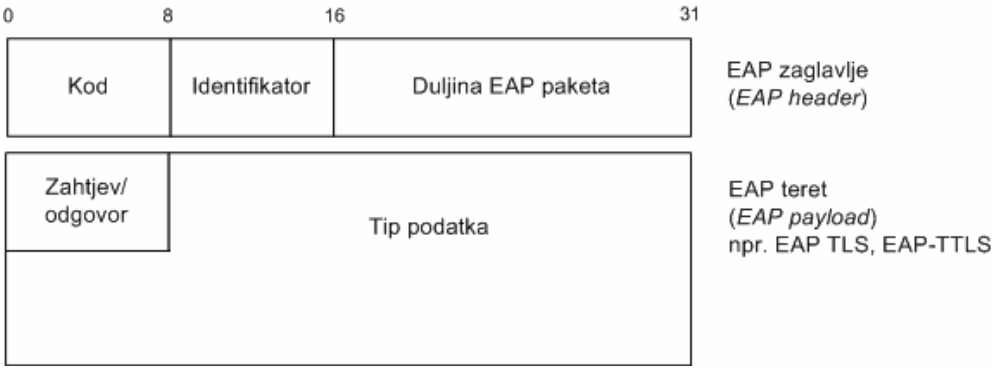
Postoje četiri osnovna tipa poruka u EAP protokolu:

1. EAP zahtjev(*EAP Request*)
2. EAP odgovor(*EAP Response*)
3. EAP uspjeh(*EAP Success*)
4. EAP neuspjeh(*EAP Failure*)

Poruka EAP zahtjev je izazov koji šalje *supplicant*, a EAP odgovor je odgovor autentifikatora *supplicant*u. Druge dvije poruke izvještavaju *supplicant* o ishodu.
Sam EAP paket se enkapsulira(ukoliko se EAP koristi u bežičnoj mreži) unutar EAPoL(EAP over LAN) paketa. EAPoL paketi služe za komunikaciju između *supplicant*a i autentifikatora preko mreže. Postoji tri vrste EAPoL paketa:

1. *EAPoL Start*
   Nalaže autentifikatoru da počne proces autentifikacije.
2. *EAPoL Logoff*
   Obavještava autentifikatora da se korisnik odjavljuje s mreže.
3. *EAPoL Key*
   Nosi informaciju o WEP dijeljenom ključu.

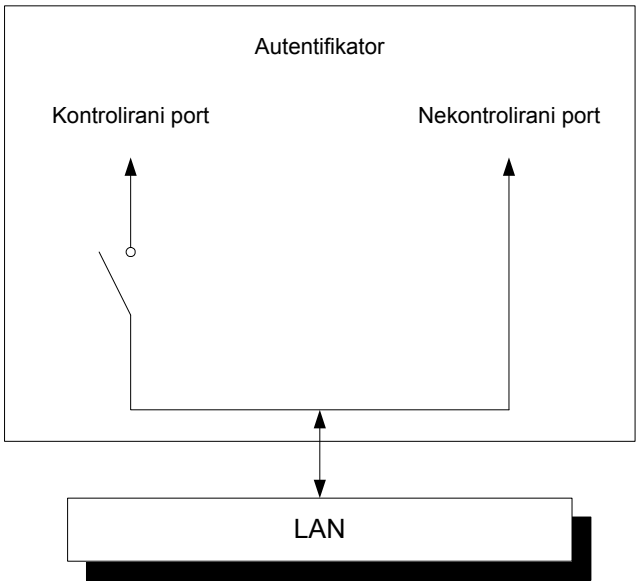
Sam EAP paket se enkapsulira unutar EAPoL paketa.
EAP paket je prikazan na slici 5.3.



Slika 5.3: Izgled EAP paketa

EAP je proširiv protokol u smislu da se unutar EAP zahtjeva/odgovora može enkapsulirati bilo koja metoda autentifikacije. EAP radi na drugom(podatkovnom) sloju OSI modela. Također ima mogućnost da sve zahtjeve za autentifikaciju preusmjeriti ka centralnom

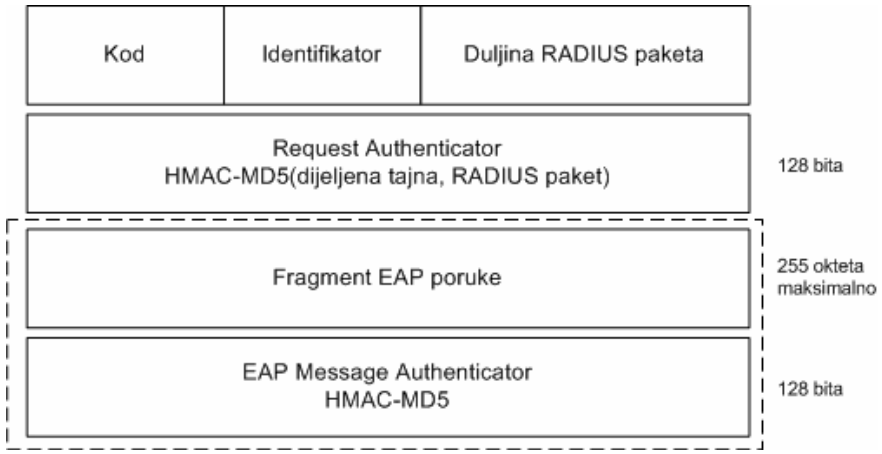
RADIUS poslužitelju što je daleko bolje rješenje od onoga u kojemu bi se svaki port brinuo o autentifikaciji pojedinog korisnika. Kako bi korisnik mogao pristupiti mreži, pristupna točka mora omogućiti EAP paketima da prođu do poslužitelja. Zbog toga autentifikator koristi dualni način rada portova(slika 5.4): nekontrolirani portovi(*uncontrolled ports*) i kontrolirani portovi(*controlled ports*). Nekontrolirani portovi ne dopuštaju nikakav drugi promet osim EAP paketa. Ovaj model je kompatibilan sa klijentima koji ne podržavaju 802.1X standard. Naime administrator može promet sa takovim klijentima preusmjeriti na nekontrolirane portove i time im omogućiti pristup mreži.



Slika 5.4: 802.1X portovi

Sigurnosni ciljevi 802.1X standarda
Postoji nekoliko sigurnosnih ciljeva koje nastoji ispuniti 802.1X standard. Ovdje će biti navedeni:

1. **Kontrola pristupa i mogućnost međusobne autentifikacije**
   Zbog same naravi bežičnih mreža dizajneri mreže nisu uvijek u mogućnosti ograničiti propagaciju radio signala unutar granica organizacije. Zbog toga mreže mogu biti izložene napadu sa parkirališta. Da bi se to spriječilo sigurnosna okosnica mora imati način za strogu kontrolu pristupa mreži kao i za obostranu autentifikaciju klijenta i pristupne točke na razini svakog pojedinog paketa. Dakle svaki paket se mora moći autentificirati.
   U samom protokolu je to ostvareno na način da autentifikator i autentifikacijski poslužitelj komuniciraju preko RADIUS protokola. Svaki autentifikator ima dijeljeni tajni ključ sa severom. Sve RADIUS poruke sadrže *Request Authenticator* polje koje je sažetak napravljen HMAC-MD5 funkcijom kosanja s dijeljenom tajnom kao ključem. Ovo polje postavlja RADIUS poslužitelj, a provjerava pristupna točka(autentifikator). Pristupna točka postavlja polje *EAP Authenticator* na sličan način. Ova dva polja pružaju autentifikaciju paketa kao i zaštitu integriteta prometa između pristupne točke i RADIUS poslužitelja u pozadini.
   RADIUS paket sa spomenutim poljima je prikazan na slici 5.5.



Slika 5.5: Izgled RADIUS paketa

2. **Fleksibilnost i skalabilnost**
   Bežične mreže imaju široko područje primjene – od mreža unutar velikih korporacija koje imaju visoke sigurnosne zahtjeve pa sve do javnih bežičnih mreža koje pružaju pretplatnicima uslugu pristupa Interneta gdje se sigurnosni zahtjevi svode na posjedovanje korisničkog imena i lozinke bez enkripcije podataka. Standard mora biti dovoljno fleksibilan da zadovolji potrebe svih korisnika bežičnih mreža.
   Odvojivši autentifikatora od samog procesa autentifikacije(klijenta autentificira RADIUS server) 802.1X dopušta iznimnu skalabilnost. Fleksibilnost je ostvarena preko EAPOL poruka u koje se mogu enkapsulirati sve vrste EAP paketa .

3. **Sveprisutna sigurnost**
   Najistaknutije svojstvo bežičnih mreža je mobilnost korisnika. Zbog toga je dizajnom okosnice korisnicima osigurati mogućnost autentifikacije bez obzira na to jesu li u svojoj domaćoj mreži ili u tuđoj. To je omogućeno razdvajanjem autentifikatora i autentifikacijskog poslužitelja na dva različita entiteta.

4. **Stroga povjerljivost podataka**
   Bežični medij zbog svojih svojstava ne osigurava dovoljnu povjerljivost podataka jer svatko sa prikladnom opremom može prisluškivati komunikaciju klijenta i pristupnih točke. Zbog toga standard mora pružiti prikladnu potporu za zaštitu povjerljivosti podataka kroz dinamičku izmjenu ključeva za enkripciju podataka između klijenta i pristupne točke.

Vrste EAP-a

EAP je veoma fleksibilan standard koji se može implementirati na više različitih načina. To omogućava standardu 802.1X da ispuni sigurnosne zahtjeve koje se pred njega postavljaju. 802.1X standard sadrži široku lepezu EAP metoda koje se mogu koristiti. Svaka od metoda ima svoje prednosti i mane. Njihov opis sada slijedi.

MD-5

MD-5 je EAP ekvivalent PPP CHAP protokolu u kojemu se koristi jednosmjerna funkcija kosanja(*hash*) u kombinaciji sa dijeljenom tajnom i izazovom kako bi se provjerilo da li *supplicant* poznaje dijeljenu tajnu. MD-5 se smatra osnovnim sigurnosnim mehanizmom i

kako takav nije preporučljiv za sustave koji zahtijevaju visoku razinu sigurnosti. Naime, kao i svi mehanizmi koji koriste slučajni izazov u kombinaciji sa lozinkom i jednosmjernom funkcijom kosanja, osjetljiv je na napad rječnikom(*dictionary attack*). Naime ako napadač uspije presresti i snimiti izazov te odgovor koji je prošao kroz funkciju kosanja on može poznavajući tu funkciju mijenjati riječ dok ne dobije istu poruku kao odgovor. Zbog toga je bitno da korisnici za lozinku ne odabiru riječi koje se nalaze u rječniku. MD-5 pruža jednosmjernu autentifikaciju (npr. klijent se autentificira mreži).

TLS(*Transport Layer Security*)
TLS nudi veoma siguran način autentifikacije koji zamjenjuje jednostavnu lozinku sa klijentskim i poslužiteljskim certifikatima kroz upotrebu PKI-ja(*Public Key Infrastructure*) kao osnove. TLS podržava uzajamnu autentifikaciju kao i dinamičke WEP ključeve. TLS je izvrstan izbor kada je potrebno implementirati visoke sigurnosne zahtjeve, a već postoji razvijena PKI infrastruktura. No PKI donosi i velike troškove u odnosu na jednostavne lozinke za svakog klijenta. Osim toga potrebno je imati i valjanu programsku podršku kao i prikladno obučene korisnike kako bi se najbolje iskoristilo PKI.

TTLS(*Tunneled Transport Layer Security*)
TTLS je ekstenzija TLS-u u kojoj je uklonjena potreba za klijentskim certifikatima. Ovo je jedan od dva protokola koji podržavaju sigurni tunel preko mreže. Sastoji se od dva koraka:

1. Asimetrični algoritam baziran na poslužiteljskom ključu služi za autentifikaciju poslužitelja i za uspostavu simetrično enkriptiranog tunela između poslužitelja i klijenta.
2. Neka druga autentifikacijska metoda se koristi kako bi poslužitelj autentificirao klijenta i to preko prethodno uspostavljenog sigurnog tunela. Druga metoda može biti EAP tipa(MD-5) ili neka druga starija metoda(CHAP, PAP, MS CHAP, MS CHAP v2).

Simetrični tunel postoji samo da bi se zaštitio proces autentifikacije klijenta i nakon toga je on nepotreban pa se urušava. Dalje je na klijentu da pomoću WEP ključa sa pristupnom točkom stvori sigurni tunel.

PEAP(Protected Extensible Authentication Protocol)
PEAP je drugi protokol koji podržava sigurni tunel preko mreže. On, kao i TTLS, stvara sigurni tunel između klijenta i pristupne točke kroz koji se autentificira klijent. No, za razliku od TTLS-a, PEAP ne podržava starije metode autentifikacije nego samo dozvoljava EAP autentifikacijske metode.

LEAP(Light Extensible Authentication Protocol)
LEAP je razvio Cisco za svoje proizvode za 802.11 standard. On pruža obostranu autentifikaciju, derivaciju sigurnog sjedničkog ključa te dinamičku raspodijelu WEP ključeva ovisno o korisniku i sjednici. LEAP je vlasništvo Cisca i može se ugrađivati samo u Ciscove uređaje. LEAP nije podržan 802.1X standardom jer donosi neke specifičnosti u odnosu na standard. LEAP je ranjiv na napade rječnikom jer se izazov i odgovor šalju u čistom obliku pa napadač može izvesti napad istovjetan onome na MD-5. No, usprkos tome, LEAP uz dobar izbor lozinke pruža značajnu sigurnost.

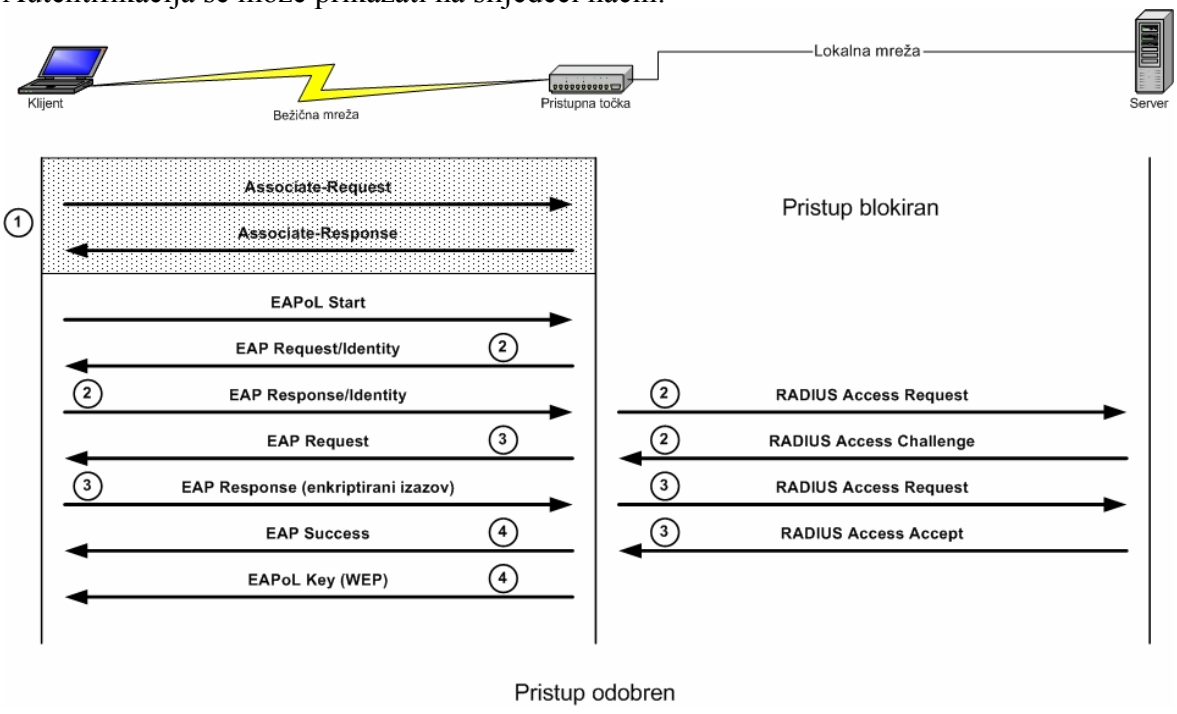
EAP – budući standardi

SIM(*Subscriber Identity Module*)
SIM je trenutno najčešće korištena autentifikacijska metoda kod proizvođača mobilnih telefona. Ima velikih sličnosti sa autentifikacijom pametnim karticama. Proizvođači mobilnih telefona prodaju klijentima SIM kartice kako bi oni mogli dobiti pristup mreži i jedinstveno se autentificirati. Iako EAP SIM još nije standard postoji velika vjerojatnost da će uskoro postati. EAP SIM arhitektura će omogućiti korisnicima da iskoriste svoju GSM opremu u autentifikaciji u bežičnim računalnim mrežama. EAP SIM pruža mogućnost obostrane autentifikacije klijenta i pristupne točke. No neki smatraju da ovaj standard ne pruža dovoljnu sigurnost jer se koriste 128 bitni ključevi koji su na određen način dobiveni iz 64 bitnih ključeva pa su zbog toga ranjivi na podvale.

AKA(Authentication and Key Agreement)
AKA je, kao i SIM, novi standard razvijen od strane pružatelja usluga mobilne telefonije. AKA je sličan SIM-u samo što kao podlogu ne koristi SIM karticu nego USIM(*User Service Identity Module*) kartice sa ugrađenim AKA algoritmima, a ne GSM uređaje sa njihovim autentifikacijskim algoritmima. Valja napomenuti da je USIM definiran unutar UMTS(Universal Mobile Telecommunications System) standarda koji je budućnost mobilnih telekomunikacija. AKA se smatra sigurniji nego SIM jer koristi stalne, a ne izvedene ključeve.

Proces autentifikacije EAP-om

Proces autentifikacije se odvija kroz komunikaciju triju entiteta: *supplicant*a(klijentsko računalo), autentifikatora(pristupna točka) te pozadinskog poslužitelja(RADIUS poslužitelj). *Supplicant* i autentifikator međusobno komuniciraju EAPoL paketima dok se komunikacija između autentifikatora i RADIUS poslužitelja odvija RADIUS paketima. Autentifikacija se može prikazati na slijedeći način:



Slika 5.6: Proces autentifikacije EAP-om

Kako vidimo proces autentifikacije se odvija u nekoliko koraka:

1. Klijent šalje zahtjev za pridruživanje mreži. Klijent ovime ne dobiva pristup mreži nego samo obavještava pristupnu točku da je tu i da želi postati članom mreže. Pristupna točka mu odobrava pridruživanje mreži, ali mu ne daje pristup uslugama viših slojeva OSI modela. Dakle klijent može slati samo EAP pakete preko nekontroliranog porta.
2. Pristupna točka zahtjeva od klijenta da pošalje svoje korisničko ime i lozinku koje dalje pristupna točka prosljeđuje RADIUS poslužitelju. RADIUS poslužitelj u svojoj bazi korisnika uspoređuje dobivene podatke sa onima iz baze i ukoliko su jednaki šalje klijentu izazov. Ukoliko se podaci ne slažu poslužitelj nalaže pristupnoj točki da odbije klijenta.
3. Pristupna točka šalje klijentu izazov koji on enkriptira te šalje natrag pristupnoj točci. Pristupna točka dalje to prosljeđuje RADIUS poslužitelju koji enkriptira sa svojim ključem poslani izazov te ga uspoređuje sa dobivenim od klijenta. Ukoliko su jednaki poslužitelj dopušta pristupnoj točki da klijentu odobri puni pristup mreži. Na ovaj način je poslužitelj autentificirao klijenta. Važno je zamijetiti da je to jednostrana autentifikacija jer klijent nema načina da autentificira poslužitelj.
4. Pristupna točka obavještava klijenta o uspješnoj autentifikaciji te mu šalje WEP ključ koji će mu služiti za enkripciju podataka između njega i pristupne točke.

EAP konačni automati

U EAP-u su definirane tri vrste konačnih automata:

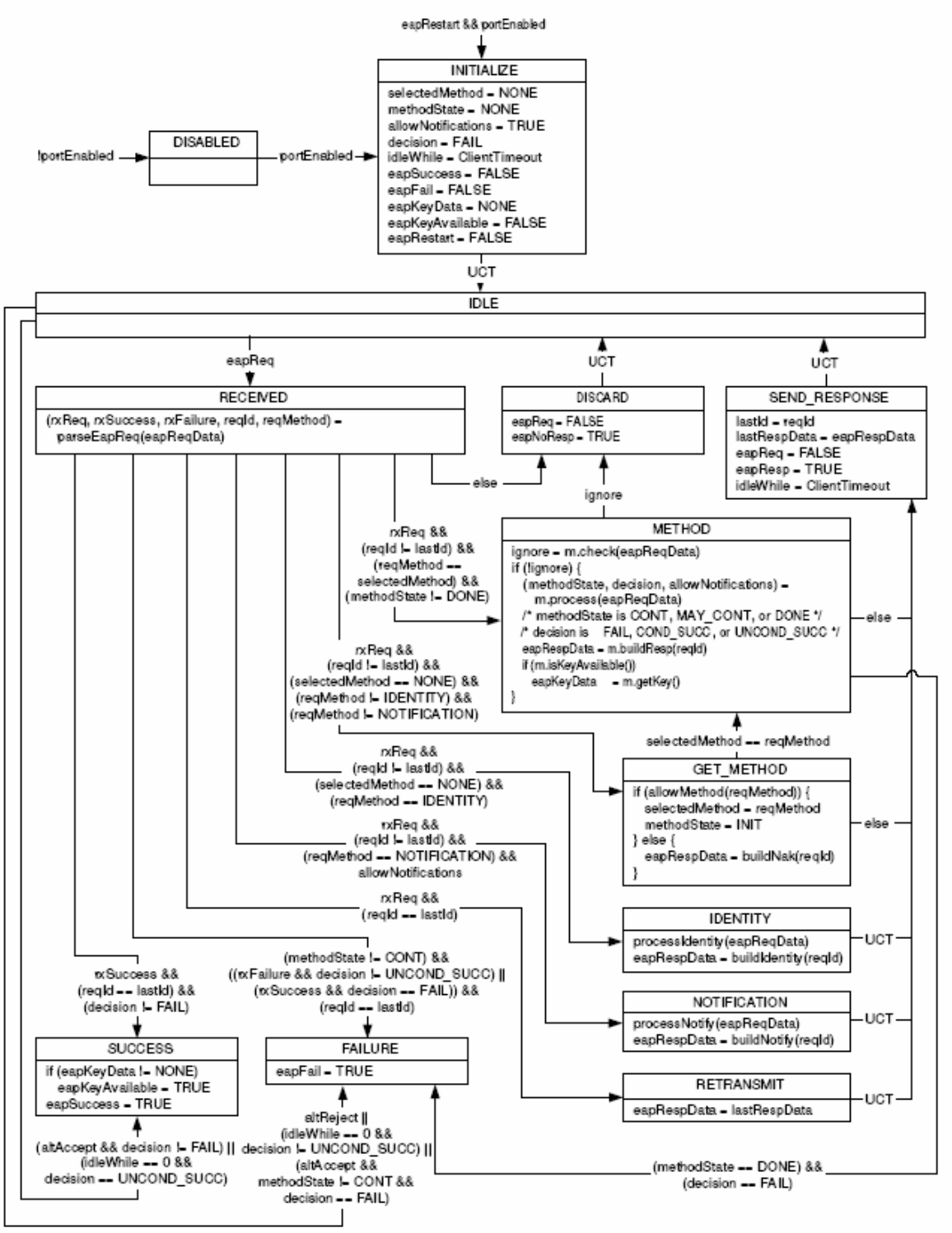
1. Konačni automat na strani klijenta(*peer state machine*)
2. Konačni automat autentifikatora(*standalone authenticator state machine*)
3. Konačni automat pozadinskog autentifikatora(*EAP backend authenticator*)

Svaki od njih je detaljno opisan u dokumentu pod brojem [8]. U daljnjem tekstu bit će dan opis klijentskog konačnog automata.

Konačni automat na strani klijenta(*peer state machine*)

Konačni automat na klijentskoj strani je prikazan na slici 5.11. Kako je vidi automat ima nekoliko stanja:

- **DISABLED**
  U ovom se stanju automat nalazi svaki puta kada se dogodi greška pri komunikaciji sa nižim slojevima ili je tu vezu nemoguće uspostaviti. Automat trenutno prelazi u stanje INITIALIZE kada se ta veza uspostavi.
- **INITIALIZE**
  U ovom stanju se inicijaliziraju sve varijable koje automat koristi.
- **IDLE**
  Ovo je stanje u kojem automat provodi najviše vremena čekajući da se dogode bitni događaji
- **RECEIVED**
  Automat prelazi u ovo stanje kada je primljen EAP paket, zaglavlje paketa se parsira u ovome stanju.
- **GET_METHOD**
  U ovo se stanje dolazi kada dođe zahtjev za novim tipom autentifikacije. Tada se ili započne proces autentifikacije ispravnom metodom ili se generira *Nak response* paket.



Slika 5.11: Konačni automat na klijentskoj strani(*peer state machine*)

- **METHOD**
  U ovom stanju se obavlja metode autentifikacije: autentifikatorov zahtjev je obrađen i generira se prikladni paket kao odgovor.
- **SEND_RESPONSE**
  Ovo stanje je signal nižem sloju OSI modela da je paket(odgovor) spreman za slanje.
- **DISCARD**

Ovo stanje je signal nižem sloju OSI modela da je zahtjev odbačen i da se nikakav odgovor na njega neće slati.

- **IDENTITY**
  Upravlja zahtjevima za *Identity method* te generira odgovore.
- **NOTIFICATION**
  Upravlja zahtjevima za *Notification method* te generira odgovore.
- **RETRANSMIT**
  Brine se za ponovno slanje prethodno generiranog odgovora.
- **SUCCESS**
  Prihvatljivo stanje koje signalizira uspjeh.
- **FAILURE**
  Prihvatljivo stanje koje signalizira neuspjeh.

Također, automat mora imati sučelje prema nižem sloju OSI modela kako bi mogao slati i primati pakete. Niži sloj komunicira sa konačnim automatom preko `eapReqData` varijable i postavljajući signal `eapReq` u vrijednost TRUE. Kada konačni automat želi poslati nešto on postavlja `eapResp` ili `eapNoResp` signal. Ukoliko je postavljen eapResp signal tada se paket koji konačni automat želi poslati nalazi u `eapReqData` varijabli. Tada niži sloj postaje odgovoran za paket tj. mora se pobrinuti da prijemna strana primi ispravan paket. Kada autentifikacija završi, konačni automat će postaviti `eapSuccess` ili `eapFailure` kako bi obavijestio niži sloj o uspjehu ili neuspjehu autentifikacije.

Varijable(komunikacija niži sloj – konačni automat) i njihovo značenje je slijedeće:

- `eapReq(boolean)`
  Ima vrijednost TRUE u nižem sloju, FALSE u konačnom automatu. Signalizira konačnom automatu da je niži sloj dobio zahtjev(*request*).
- `eapReqData(EAP Packet)`
  Ima neku vrijednost kada je eapReq postavljen u vrijednost TRUE. Njegova vrijednost je primljeni zahtjev.
- `portEnabled(boolean)`
  Signalizira da bi konačni automat trebao biti spreman za komunikaciju. Vrijednost ove varijable je postavljena u TRUE kada niži sloj započne komunikaciju sa konačnim automatom. Ako u bilo kojem trenutku se prekine komunikacija nižeg sloja sa konačnim automatom ili sa prijemnikom, ova varijabla se postavlja u FALSE te konačni automat prelazi u stanje DISABLED.
- `idleWhile(integer)`
  Cjelobrojna varijabla u kojoj je pohranjeno vrijeme koje je konačni automat proveo neuposlen. Vrijeme mjeri vanjski brojač vremena.
- `altAccept(boolean)`
  Drugi način obavještavanja o uspjehu autentifikacije. Svrha ove varijable je opisana u dokumentu [9]
- `altReject(boolean)`
  Drugi način obavještavanja o neuspjehu autentifikacije. Svrha ove varijable je opisana u dokumentu [9]

Varijable(komunikacija konačni automat – niži sloj) i njihovo značenje je slijedeće:

- `eapResp(boolean)`

Ima vrijednost TRUE u konačnom automatu, FALSE u nižem sloju. Signalizira nižem sloju da je odgovor spreman.

- `eapNoResp(boolean)`
  Ima vrijednost TRUE u konačnom automatu, FALSE u nižem sloju. Signalizira nižem sloju kako je zahtjev obrađen, ali konačni automat ne želi poslati nikakav odgovor.
- `eapSuccess(boolean)`
  Ima vrijednost TRUE u konačnom automatu, FALSE u nižem sloju. Signalizira nižem sloju kako je konačni automat u stanju SUCCESS.
- `eapFail(boolean)`
  Ima vrijednost TRUE u konačnom automatu, FALSE u nižem sloju. Signalizira nižem sloju kako je konačni automat u stanju FAILURE.
- `eapRespData(EAP Packet)`
  Ova varijabla je postavljena na neku vrijednost kada `eapResp` varijabla ima vrijednost TRUE. Vrijednost varijable je EAP paket.
- `EapKeyData(EAP Key)`
  Ima vrijednost kada je dostupan ključ. Postavlja se tijekom METHOD stanja.
- `EapKeyAvailable(boolean)`
  Postavlja se na TRUE u stanju SUCCESS ukoliko je ključ dostupan.

Postoji i također konstanta:

- `ClientTimeout(integer)`
  Promjenjiva vrijednost vremenskog intervala unutar kojega mora doći valjani zahtjev, inače se prekidaju sve akcije. Vrijednost ovisi o implementaciji.

Također je potrebno sučelje između konačnog automata i EAP metoda. Ovdje će biti opisan način interakcije konačnog automata i EAP metoda. Postoje slijedeći tipovi varijabli:

- ULAZNE: `eapReqData`(uključuje i `reqId`)
- IZLAZNE: `ignore,eapRespData,allowNotifications,decision`
- ULAZNO/IZLAZNE: `methodState, (method-specific state)`

Ukoliko je `methodState`==INIT metoda inicijalizira svoje stanje(*method-specific state*). Nakon toga metoda mora odlučiti hoće li obraditi primljeni EAP paket ili će ga odbaciti. Ukoliko primljeni paket izgleda kao da nije ga poslao autentfikator, metoda može postaviti varijablu `ignore` u vrijednost FALSE. U tom slučaju metoda ne smije mijenjati stanje niti jedne druge varijable.
No ukoliko je paket ispravan i metoda ga odluči obraditi ona čini slijedeće:

- Obnavlja vlastito stanje(*method-specific state*)
- Ukoliko je metoda izvela ključ koji želi dalje poslati ona ga sprema u `eapKeyData` varijablu.
- Kreira odgovor u obliku paketa i sprema ga u `eapRespData` varijablu.
- Postavlja varijablu `ignore` u vrijednost TRUE.

Nadalje metoda mora postaviti vrijednosti varijabli `methodState` i `decision` prema slijedećim pravilima:

- `methodState`=CONT
  Metoda uvijek nastavlja svoj rad u ovoj točki. Varijabla `decision` je uvijek postavljena u FAIL.
- `methodState`=MAY_CONT

U ovoj točci autentifikator može odlučiti ili nastaviti započetu metodu ili prekinuti komunikaciju. Varijabla `decision` govori što napraviti ukoliko je komunikacija prekinuta. Ukoliko trenutna situacija ne zadovoljava klijentovu sigurnosnu politiku postavi `decision` na FAIL inače na COND_SUCC.

- `methodState`=DONE
  Metoda nikada ne nastavlja u ovoj točki. Ukoliko nas je autentifikator obavijestio kako neće dozvoliti pristup klijentu ili klijent ne želi komunicirati sa ovim autentifikatorom(kršenje sigurnosne politike) tada postavi `decision` na FAIL.
  Ukoliko je server dozvolio pristup i slijedeći paket će biti *EAP Success* te ukoliko je klijent voljan pristupiti mreži postavi `decision` na UNCOND_SUCC.
  Inače nije poznata odluka servera no klijent i dalje želi pristupiti mreži pa se postavlja `decision` na COND_SUCC.

Na kraju metoda mora postaviti `allowNotification` varijablu. Ukoliko je novo stanje metode CONT ili MAY_CONT i metoda ne zabranjuje uporabu obavijesnih poruka(*notification messages*) , postavi `allowNotification` varijablu na vrijednost TRUE, inače na FALSE.

Konačni automat također sadrži lokalne varijable. Varijable se dijele na :

- dugoročne(*long-term*) varijable – zadržavaju vrijednost kroz više paketa podataka
- kratkoročne(*short-term*) varijable – ne zadržavaju vrijednost kroz više paketa podataka.

Dugoročne lokalne varijable

- `selectMethod(EAP Type)`
  Postavlja se u GET_METHOD stanju. Označava metodu koja je trenutno u tijeku.
- `methodState(enumeration)`
  Opisana je u gornjem tekstu.
- `lastID(integer)`
  Postavlja se u SEND_RESPONSE stanju. Ima vrijednost *EAP identifier-a* posljednjeg zahtjeva.
- `lastRespData(EAP Packet)`
  Postavlja se u SEND_RESPONSE stanju. Ima vrijednost posljednje poslanog paketa.
- `decision(enumeration)`
  Opisana je u gornjem tekstu.

Kratkoročne lokalne varijable

- `rxReq(boolean)`
  Postavlja se u RECIEVED stanju. Kazuje da li je primljeni paket zahtjev(*EAP request).*
- `rxSuccess(boolean)`
  Postavlja se u RECIEVED stanju. Kazuje da li je primljeni paket zahtjev(*EAP success).*
- `rxFailure(boolean)`
  Postavlja se u RECIEVED stanju. Kazuje da li je primljeni paket zahtjev(*EAP failure).*
- `reqId(integer)`
  Postavlja se u RECIEVED stanju. Ima vrijednost identifikatora trenutnog zahtjeva.

- `reqMethod(EAP Type)`
  Postavlja se u RECIEVED stanju. Govori o metodi trenutnog zahtjeva.
- `ignore(boolean)`
  Postavlja se u METHOD stanju. Kazuje hoće li metoda prihvatiti trenutni zahtjev.

Konačni automat također ima i neke procedure:

- `parseEapReq()`
  Određuje kod, vrijednost identifikatora i tip trenutnog zahtjeva. Također provjerava je li paket dugačak onoliko koliko je zapisano u *length* polju.
- `processNotify()`
  Obrađuje sadržaj *Notification Request* (npr. prikazuje obavijest korisniku ili je sprema u datoteku za logiranje)
- `buildNotify()`
  Generira odgovarajući odgovor na *Notification Request.*
- `processIdentity()`
  Obrađuje sadržaj *Identity Request.*
- `buildIdentity ()`
  Generira odgovarajući odgovor na *Identity Request*.
- `m.integrityCheck()`
  Procedura koja ovisi o vrsti metode, a ima ulogu ispitati valjanost poruke.
- `m.process()`
  Procedura koja parsira i obrađuje zahtjev za tu metodu.
- `m.getKey()`
  Procedura koja dobavlja ključ koji će koristiti EAP ili niži sloj.

Ovdje je opisan konačni automat koji se nalazi u *supplicantu.*

5.1.1 Sigurnosni propusti u 802.1X standardu

EAP, koji je glavni i najvažniji dio 802.1X standarda, prvenstveno je namijenjen za korištenje u "žičanim" lokalnim mrežama i to kao PPP(*point-to-point protocol*). Njegova uporaba u bežičnim mrežama gdje se podaci šalju radijskim signalom donijela je neke nove sigurnosne probleme.

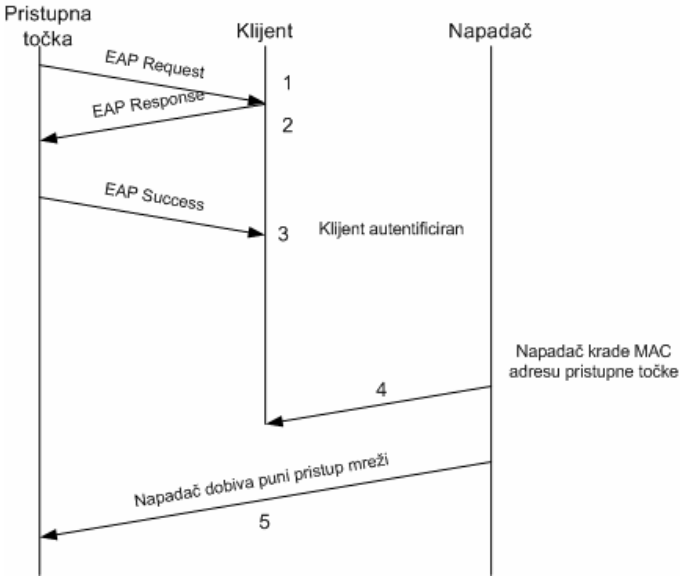
Napad čovjek-u-sredini

Glavni sigurnosni propust u EAP standardu je odsutnost načina na koji bi *supplicant* mogao autentificirati autentifikatora. *Supplicant* i autentifikator komuniciraju porukama koje su ulazni jezik za konačne automate koji su glavni dio *supplicant*a i autentifikatora.
Konačni automati *supplicant*a i autentifikatora su asimetrični: konačni automat na strani autentifikatora kontrolira port samo kada je klijent ispravno autentificiran dok je klijentov port stalno u stanju *autentificiran*. Jednostrana autentifikacija otvara mogućnost napada čovjek-u-sredini u kojem bi se napadač klijentu predstavio kao pristupna točka, a pristupnoj točki kao klijent. Prema standardu konačni automat na autentifikatorovoj strani prihvaća samo *EAP Response* poruke od *supplicant*a, a *supplicant*u odašilje samo *EAP Request* poruke. *Supplicant* ne odašilje nikada *EAP Request* poruke dakle nikada ne autentificira autentifikatora. To je propust i čitave okosnice jer ni viši slojevi ne podržavaju obostranu autentifikaciju.
EAP TLS pruža mogućnost obostrane autentifikacije ali njegovo korištenje kao autentifikacijskog mehanizma nije obavezno. No čak i kada se koristi, pogreška u dizajnu EAP-a omogućava napadaču uspješan napad.

EAP Success poruka se šalje *supplicant*u nakon što je autentifikator primio *RADIUS Access Accept* poruku od autentifikacijskog poslužitelja(RADIUS). Ta poruka obavještava konačne automate na strani *supplicant*a i autentifikatora da je autentifikacija uspješno obavljena. Kada *supplicant* primi *EAP Success*, ona postavi u konačnom automatu *eapSuccess* zastavicu što uzrokuje bezuvjetan prelazak konačnog automata u stanje *Authenticated* bez obzira u kojem se stanju automat prije nalazio. Koristeći ovu činjenicu napadač može obmanuti autentifikatora i tako izvesti napad čovjek-u-sredini. Napadač tada može vidjeti sav promet između klijenta i pristupne točke. Na ovaj način napadač je zaobišao i autentifikacijske metode viših mrežnih slojeva.

Krađa sjednice(Session Hijacking)

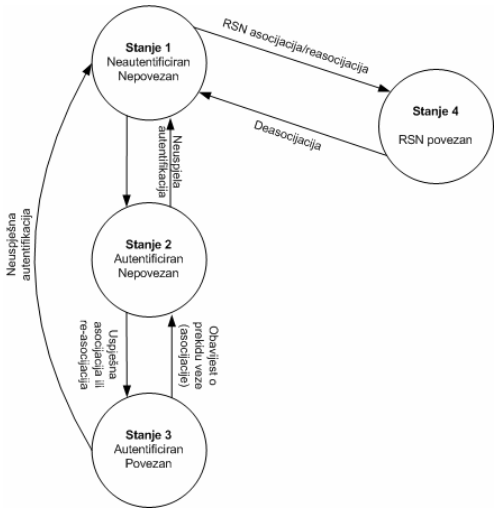
U 802.1X standardu postoje dvije vrste konačnih automata koje utječu na proces autentifikacije: RSN konačni automat i 802.1X konačni automat. Njihov kombinirani rad diktira stanje autentifikacije klijenta. Zbog toga što se poruke šalju u čistom, nekriptiranom, obliku napadač je u mogućnosti neovlašten pristup mreži. Napad se može prikazati slikom 5.12.



Slika 5.12: Prikaz krađe sjednice

Kao što vidimo napad se sastoji od nekoliko koraka:

1. Poruke 1, 2, 3: Klijent se autentificira pristupnoj točki. Iako se sam proces autentifikacije sastoji od nekoliko koraka ovdje su prikazani samo oni najosnoviniji.
2. Poruka 4: Napadač, koristeći MAC adresu pristupne točke, šalje klijentu upravljački okvir(*disassociate managment frame*) koji uzrokuje da klijent pomisli da je izgubio pristup mreži. Ovom porukom RSN konačni stroj prelazi u stanje neasociran(*unassociated*) dok 802.1X konačni automat još uvijek ostaje u stanju autentificiran(*authenticated*). RSN konačni automat ima izgled prikazan na slici 5.13.



Slika 5.13: RSN konačni automat

3. Poruka 5: Napadač dobiva puni pristup mreži koristeći MAC adresu klijenta jer je 802.1X konačni automat(u pristupnoj točci) još u stanju autentificiran.

Nedostatak mehanizma autentifikacije i provjere integriteta paketa
Sam 802.1X standard nema prikladan mehanizam koji bi omogućavao autentifikaciju i provjeru integriteta svakog pojedinog paketa. To je ključna činjenica koja otvara mogućnost napadima na sigurnost bežične mreže. Primjerice prethodno naveden napad, krađa sjednice, je moguć jer poruke između klijenta i pristupne točke nisu prikladno zaštićene. Dok su podatkovni paketi zaštićeni kada se koristi WEP enkripcija(nakon što završi proces autentifikacije), upravljački okviri nisu nikada enkriptirani što ostavlja mogućnost krivotvorenja i izmjene informacija.

5.1.2 Moguća rješenja sigurnosnih propusta u 802.1X standardu

Kao što smo vidjeli 802.1X standard ima neke sigurnosne propuste koji bi se morali ispraviti kako bi dobili zadovoljavajuću sigurnost. Ovdje će biti navedena neka moguća poboljšanja.
Simetrična autentifikacija
Oba entiteta koja sudjeluju u procesu autentifikacije bi se trebala moći međusobno autentificirati. Zbog toga bi trebalo u standard dodati mogućnost simetrične autentifikacije. Dakle klijent bi trebao moći autentificirati pristupnu točku kao i pristupna točka klijenta. Konačni automat *supplicant*a bi trebao postati sličan onome autentifikatora tj. trebao bi se i na klijentskoj strani uvesti model portova kao što sada postoje kod autentifikatora. Također bi RADIUS poslužitelj trebao tretirati klijenta kao što sada tretira pristupnu točku.
Skalabilna autentifikacija
Kako bi omogućio prirodno neograničenu mobilnost korisnika unutar bežične mreže standard bi trebao riješiti problem sa dijeljenim ključem koji je prema sadašnjem standardu ovisan o pristupnoj točki. Dakle svaki puta kada se klijent premjesti u doseg druge pristupne točke on mora dobiti novi dijeljeni ključ.

Iako je postignut velik pomak u odnosu na 802.11x standard i njegove mehanizme kontrole pristupa(*SSID, Open System Authentication, Shared Key Authentication*) i zaštite podataka(*WEP*) još uvijek postoje značajni propusti u sigurnosti 802.1X standarda.

Prethodno opisani napadi pokazali su nekoliko slabosti u standardu. Na sreću ti se propusti dadu lagano ukloniti i time će se značajno povećati sigurnost bežičnih mreža u kojima se koristi 802.1X.

5.2 WEP2

Ovaj standard je još jedan od pokušaja povećanja sigurnosti bežičnih mreža. Kako se iz imena standarda dade naslutiti on je nastao nadograđivanjem WEP-a i s time je naslijedio neke slabosti u dizajnu. IEEE je načinio preinake u duljini ključa koji je proširen na 128 bita(prije 40 bita) te proširivanjem polja u kojemu se nalazi inicijalizacijski vektor koje je sada veliko 128 bita(prije 24 bita). Također donosi i podršku za Kerberos V protokol. No ostao je isti enkripcijski algoritam – RC4 i isti način upravljanja ključevima pa se može zaključiti da WEP2 ne donosi velik pomak u poboljšanju sigurnosti. Dobra stvar je da je WEP2 kompatibilan sa WEP protokolom tako da mrežna oprema uz određenu programsku nadogradnju može koristiti WEP2 protokol.

5.3 IPsec

IPsec(*IP security*) je skup protokola koji je razvila organizacija IETF(*Internet Engineering Task Force*) kako bi zaštitila razmjenu podataka preko IP sloja u mrežnom protokolu koji se koristi na Internetu. IPsec se koristi kod implementacije virtualne privatne mreže(*virtual private network*). Dakle osnovna namjena IPsec protokola je mogućnost uspostave sigurne komunikacije između dva računala preko nesigurnog medija kao što je Internet ili lokalna mreža.
IPsec ima dva osnovna načina rada:

- Transportni način(*transport mode*)
  U ovom načinu rada enkriptira se samo teret(*payload*) koji nosi paket, ne i njegovo zaglavlje(*header*).
- Tunelski način(*tunnel mode*)
  U ovom načinu rada enkriptira se i teret i zaglavlje paketa. Smatra se sigurnijim od transportnog načina.

Također postoje i dva protokola koji se koriste u IPsec-u:

- AH(*Authentification header*)
  Koristi se samo za autentifikaciju svakog pojedinog paketa.
- ESP(*Encapsulating Security Payload*)
  Pruža usluge autentifikacije i enkripcije svakog pojedinog paketa.

Dakle iz prethodno nabrojenog može se vidjeti da korisnik koji želi autentificirati primljeni paket ima sveukupno četiri različita načina na raspolaganju:

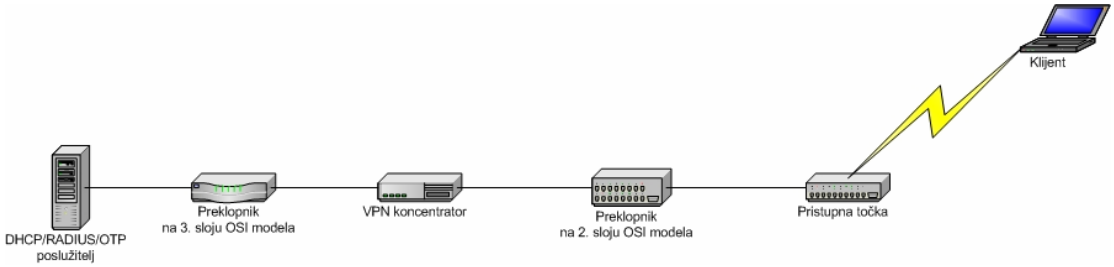
- Transportni način/AH
- Tunelski način/AH
- Transportni način/ESP ukoliko se ne koristi enkripcija
- Tunelski način/ESP ukoliko se ne koristi enkripcija

Međusobna razlika je minorna tako da je praktično svejedno koji će se način koristiti. No to donosi sa sobom i pitanje zašto su nam onda potrebna dva protokola koji se koriste. Misao vodilja kreatora IPsec-a je bila da se AH koristi kada je potrebna autentifikacija, a ESP kada je uz autentifikaciju potrebno i pakete zaštititi enkripcijom. No na prvi pogled se vidi kako je transportni način rada ustvari dio tunelskog načina rada, gledajući iz perspektive mreže. Jedina prednost korištenja transportnog načina rada je u nešto manjem opterećenju mreže, no to i to se dade popraviti ukoliko se u tunelskom modu primjene neke

metode kompresije zaglavlja paketa. Dakle nameće se zaključak kako se transportni način rada može izbaciti iz protokola i može se, bez ikakvih posljedica, koristiti samo tunelski način rada.
U transportnom načinu rada bolje je koristiti AH protokol jer on autentificira i zaglavlje paketa. U tunelskom načinu rada najčešće se koristi ESP protokol.
AH protokol ima i neke nedostatke. Najveći nedostatak to da on autentificira zaglavlja i nižih mrežnih slojeva što je izravno kršenje načela modularnosti u izradi mrežnih protokola. Zbog toga AH protokol mora poznavati sve tipove podataka koji se koriste u zaglavljima nižih mrežnih slojeva. Ukoliko se dogode izmjene u standardu koji opisuje neki od nižih mrežnih protokola to će dovesti do problema sa AH protokolom u IPsec-u. Tunelski način rada u kombinaciji sa ESP-om otklanja navedeni problem ali on je i zahtjevniji na širinu pojasa(*bandwidth*).

5.3.1 Korištenje IPsec-a u bežičnim mrežama

Osnovna ideja korištenja IPsec-a u bežičnim mrežama jeste da bežičnu mrežu smatramo kao i javnu(nesigurnu) mrežu. Dakle bežičnu mrežu treba staviti van intraneta organizacije i na taj način povećati sigurnost mreža. Kada se želi primijeniti IPsec u bežičnoj mreži prvi korak je instalacija klijentskog programa(ukoliko već sami operacijski sustav nema podršku) koji podržava IPsec na svako računalo u bežičnoj mreži. Na taj način se osigurava da klijent mora prije nego što se pridruži bežičnoj mreži uspostaviti IPsec tunel do žične mreže i samo kroz njega on može komunicirati sa drugim računalima. Promet se filtrira na više slojeva- na 2 i 3. sloju OSI modela i na taj način se osigurava komunikacija isključivo preko sigurnog tunela. Model mreže koja koristi IPsec se može vidjeti na slici 5.14.



Slika 5.14: Model mreže sa IPsec-om

Na slici su vidljivi bitni entiteti:

- Klijentska programska i sklopovska podrška koja omogućava komunikaciju klijenta i pristupne točke(mrežna kartica, antena, prikladni pogonski programi…)
- VPN klijentska programska podrška sa osobnim vatrozidom(*firewall*) omogućava uspostavu sigurnog tunela od kraja do kraja tj. od klijentskog računala do VPN koncentratora. Vatrozid štiti korisnika od raznih opasnosti sa mreže.
- Pristupna točka pruža usluge spajanja klijentima, ali i vrši filtriranje po IP adresama između klijenta i mreže.
- Preklopnik na drugom sloju OSI modela spaja lokalnu mrežu sa pristupnom točkom. Neki noviji modeli imaju mogućnost korištenja VLAN ACL(VACL) koja dodaje još jedan sloj u filtriranju adresa.
- Preklopnik na trećem sloju OSI modela ima ulogu usmjeravanja IP paketa prema raznim modulima no pruža i uslugu filtriranja IP paketa sa bežične mreže.
- RADIUS poslužitelj se koristi pri autentifikaciji korisnika žične/bežične mreže. Opcionalno komunicira sa OTP poslužiteljom.
- OTP(*One-time password*) poslužitelj ima ulogu autorizacije OTP informacija koje šalje RADIUS poslužitelj.

- DHCP poslužitelj daje adrese VPN klijentima prije i poslije uspostavljanja VPN-a.
- VPN koncentrator ima ulogu autentificiranja klijenata, a može i dodjeljivati IP adrese klijentima(one koje je dobio od DHCP poslužitelja) .

Kao što vidimo uvođenje VPN-a u neku organizaciju donosi nove, velike investicije. Na svakoj organizaciji je da procijeni isplati li se uvoditi tako kompleksan sustav s obzirom na njihove potrebe.
IPsec otklanja nekoliko vrsta opasnosti:

- **Krađa paketa**
  Krađa paketa je nemoguća jer se koristi jaka enkripcija svih podataka koji se šalju u eter. Novija programska podrška dopušta da se tunel automatski digne čim se korisnik pokuša spojiti na mrežu tj. kada mu se dodijeli ispravna IP adresa. Na ovaj način je izbjegnuto da korisnik mora ručno podizati tunel, on je riješen svih tehničkih detalja.
- **Napad čovjek-u-sredini**
  Ovaj tip napada je onemogućen korištenjem enkripcije i autentifikacije klijenata.
- **Neovlašten pristup**
  Kako je jedino inicijalnim protokolima(DHCP za dodjelu privremene IP adrese, IKE(*Internet Key Exchange*) koji se brine o raspodijeli ključeva te ESP kojim se uspostavlja siguran tunel) dozvoljen prolazak preko filtara nije moguće da bi napadač mogao doći do podataka iz lokalne mreže preko pristupne točke. Dodatno se mogu pojačati mjere sigurnosti na VPN koncentratoru ovisno o vrsti korisnika koji se žele spojiti.
- **Umetanje i krađa IP paketa**
  Napadač može snimiti IP paket ali ga ne može ponovno iskoristiti jer neće proći provjeru i autentifikaciju kroz sve filtre.
- **Umetanje i krađa ARP paketa**
  Moguće je da napadač snimi ARP promet na mreži no ne može doći do nikakvih podataka jer se koristi jaka i sigurna enkripcija(3DES ili u novije vrijeme AES)
- **Otkrivanje topologije mreže**
  Kako su dozvoljeni protokoli samo IKE, DNS, DHCP te ESP napadač ne može prodrijeti sa ICMP(*Internet Control Message Protocol*) paketima u mrežu, a tako ne može ni otkriti topologiju mreže.

Ovome se protokolu zamjera velika kompleksnost. Kompleksnost sustava je u suprotnosti sa njegovom sigurnošću jer što je sustav kompleksniji veća je i mogućnost da se dizajnerima potkradu greške. Sustav je kompleksan ponajprije jer je na njegovom dizajnu radilo mnogo ljudi pa se nastojalo da se sve strane zadovolji nauštrb jednostavnosti dizajna sustava. Naime ranije je napomenuto da se može transportni način rada, uz manje preinake, u potpunosti izbaciti i koristiti samo tunelski način rada. Isto bi se moglo zaključiti i za AH protokol koji vrši samo autentifikaciju korisnika i on bi se mogao zamijeniti sa ESP protokolom koji dozvoljava autentifikaciju i enkripciju. Nadalje većina stručnjaka smatra da bi se trebalo učiniti autentifikaciju u ESP protokolu učiniti obvezatnom, a ne kako je sada, opcionalnom. Opcionalna bi trebala ostati samo enkripcija.
No nakon svega možemo zaključiti da je IPsec/VPN tehnologija najbolji izbor za maksimalnu sigurnost bežičnih računalnih mreža i unatoč visokoj cijeni uvođenja i kompleksnosti sustava trebalo bi je koristiti u svakoj važnijoj bežičnoj računalnoj mreži.

6. BUDUĆI STANDARDI

Uvidjevši razne nedostatke sadašnjih standarda međunarodna standardizacijska tijela su nastavila rad na boljim i sigurnijim standardima. Kao rezultat su nastala dva standarda WPA(*Wi-Fi Protected Access*) i 802.11i.

6.1 WPA(Wi-Fi Protected Access)

WPA je donijelo tijelo pod imenom Wi-Fi Alliance, koje se sastoji od proizvođača mrežne opreme, u suradnji sa IEEE. WPA je nastao kao odgovor na uočene probleme u WEP-u i pri dizajnu se pazilo da se uklone svi nedostatci, a da se pritom i zadrži kompatibilnost sa postojećom mrežnom opremom. WPA koristi TKIP(*Temporal Key Integrity Protocol*) za enkripciju i 802.1X standard sa nekim od uobičajenih EAP protokola za autentifikaciju. Novost je i uvođenje MIC-a (*Message Integrity Check* znanog kao i *"Michael"*) kako bi se spriječilo krivotvorenje paketa. TKIP će biti opisan u slijedećem poglavlju. Enkripcija se, zbog kompatibilnosti, vrši sa RC4 algoritmom.
Prednost WPA je da se može, bez većih troškova, ugraditi i u sadašnju mrežnu opremu. Dovoljno je instalirati nove pogonske programe u pristupnim točkama i klijentskim mrežnim karticama kako bi se prešlo na novi standard. Ukoliko se kupuje nova oprema važno je da ona podržava WPA. Velike korporativne mreže potrebno je i nadopuniti RADIUS poslužiteljem kako bi se autentifikacija mogla vršiti pomoću 802.1X standarda. Također je potrebno i odabrati tip EAP-a koji će se koristiti. No kako velike mreže imaju mnogo klijenata neki proizvođači omogućavaju rad u tzv. miješanom načinu u kojemu se koriste i WEP(klijenti koji nisu instalirali nove pogonske programe svojih mrežnih kartica) i WPA. No preporučuje se da ta faza prelaska bude čim manja kako bi se postigla najbolja sigurnost. Za male kućne i uredske mreže predviđeno je da se autentifikacija vrši preko dijeljenih ključeva kako ne bi morali ulagati u RADIUS poslužitelj.
Može se reći da je WPA jedan korak dalje prema boljem i potpunijem standardu koji bi osiguravao bežične mreže. Donosi mnoga poboljšanja uz prihvatljive materijalne troškove. Puno je isplativiji od današnjih IPsec rješenja i bolji jer radi na drugom sloju OSI modela.

6.2 WPA2

WPA2 je nadogradnja na WPA i jedina razlika među njima je što se kao enkripcijski algoritam koristi AES, a ne RC4. AES je prihvaćen kao službeni enkripcijski algoritam NIST-a(*National Institute of Standards and Technology*), ujedno kao i nasljednik DES-a. AES je simetrični algoritam i u ovom standardu se koristi u CCM(*cipher-block chaining mode*) načinu rada. Koristeći ga na taj način osigurano je da bude upotrebljiv i u IBSS(*Independent Basic Service Set*) načinu rada bežičnih mreža kada klijenti komuniciraju izravno jedan s drugim bez posredovanja pristupne točke. Duljina ključeva u AES-u je 128, 192 ili 256 bita.
No WPA2 donosi i značajna materijalna ulaganja u novu mrežnu opremu jer je sadašnja preslaba da bi mogla bez značajnijeg pada performansi omogućiti rad korisnicima. Uzrok tome su veliki sklopovski zahtjevi AES-a. Svaka organizacija bi trebala procijeniti je li isplativo takovo ulaganje u mrežnu opremu.

6.3 RSN(Robust Security Network)

RSN je odgovor IEEE-a na uočene nedostatke 802.11x standarda u pogledu sigurnosti i on obuhvaća dva standarda WPA2 te 802.11i.

RSN se sastoji od dva osnovna podsustava:

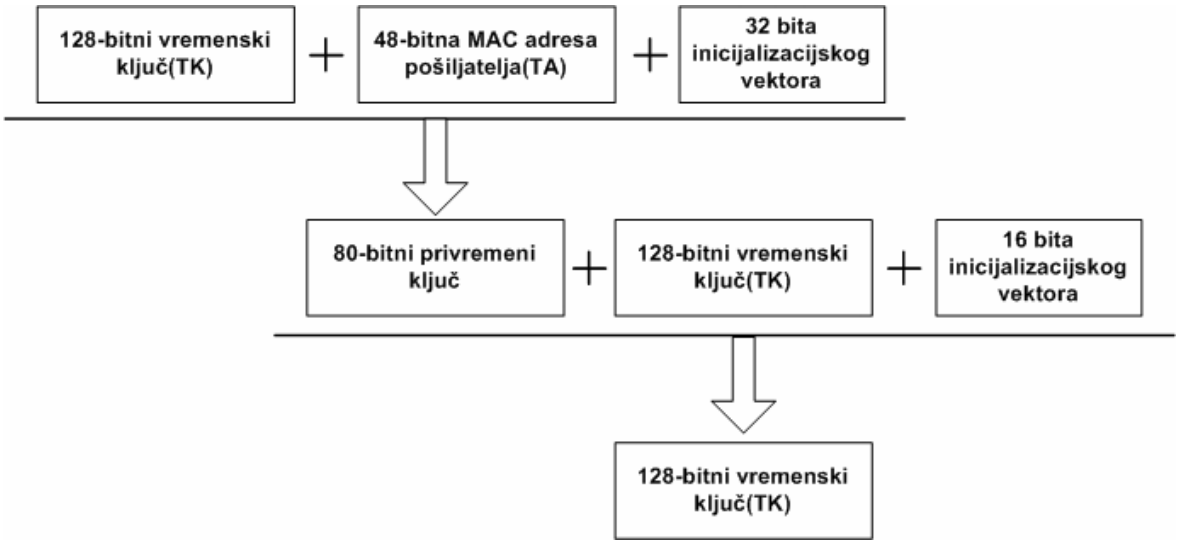
- **Podsustav za enkripciju**
  Ima ulogu zaštite privatnosti i integriteta podataka. Može biti dvojak: TKIP ili AES.
- **Podsustav za autentifikaciju**
  Ovaj podsustav se brine za autentifikaciju korisnika mreže. Pristupna točka i klijent imaju mogućnost dogovoriti se oko načina autentifikacije tj. oko protokola koji će koristiti. Za autentifikaciju i distribuciju ključeva se koristi 802.1X standard iako RSN standard ne propisuje koji standard se treba koristiti.

6.3.1 TKIP(Temporal Key Integrity Protocol)

TKIP je nastao kao zakrpa na postojeći WEP. IEEE uvidjevši sve slabosti u dizajnu WEP-a je odlučio je ukloniti sve nedostatke prethodnog standarda. Uvjeti koji su bili postavljeni su bili da standard mora biti kompatibilan sa WEP-om tj. da se trenutna mrežna oprema može samo programski nadograditi na kako bi se prešlo na novi standard kao i da novi standard mora biti ne toliko zahtjevan na opremu(koja se u 90% slučajeva sastoji od ARM7 ili i386/486 računala). To je ostvareno tako da TKIP koristi onaj dio sklopovlja koji WEP nije koristio tako da se performanse mreže ne degradiraju značajno.
TKIP je uklonio probleme WEP-a slijedećim zahvatima:
Inicijalizacijski vektor je povećan sa 24 na 48 bita. Na ovaj način je onemogućeno da se dva paketa enkriptiraju pomoću istog inicijalizacijskog vektora.
Dinamička raspodjela ključeva, miješanje ključeva i bolja zaštita integriteta paketa. Za razliku od WEP-a u ovome se protokolu velika pažnja posvetila kompleksnom problemu raspodjele ključeva. Izbjegava se i problem kriptografski slabih ključeva koji se pojavljivao u WEP-u.
Proces dobivanja ključeva se sastoji od dvije faze:

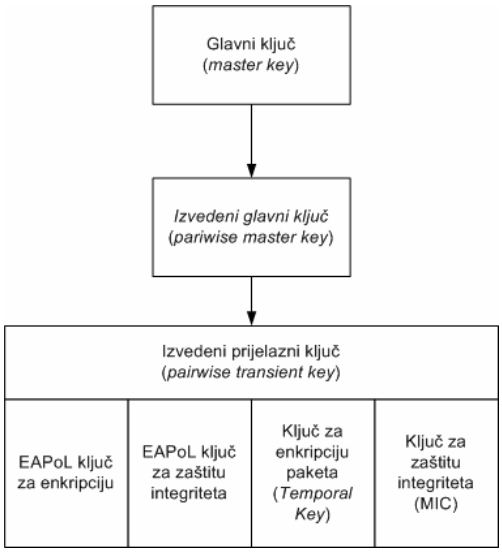
- Vremenski ključ(*temporal key*) veličine 128 bita se miješa sa klijentskom MAC adresom(*transmitter address*) veličine 48 bita te sa najznačajnijih 32 bita inicijalizacijskog vektora i na taj način se dobije privremeni ključ veličine 80 bita. Vremenski ključ poznaju i pošiljatelj i primatelj paketa.
- Privremeni ključ koji je dobiven u prošloj fazi se miješa ponovno sa vremenskim ključem te sa preostalih 16 bita inicijalizacijskog vektora.

Na ovaj način je dobiven 128 bitni ključ koji će koristiti RC4 algoritam za enkripciju jednog i samo jednog paketa. Prikaz opisanog se vidi na slici 6.1.



Slika 6.1: Prikaz procesa miješanja ključeva(*key mixing*)

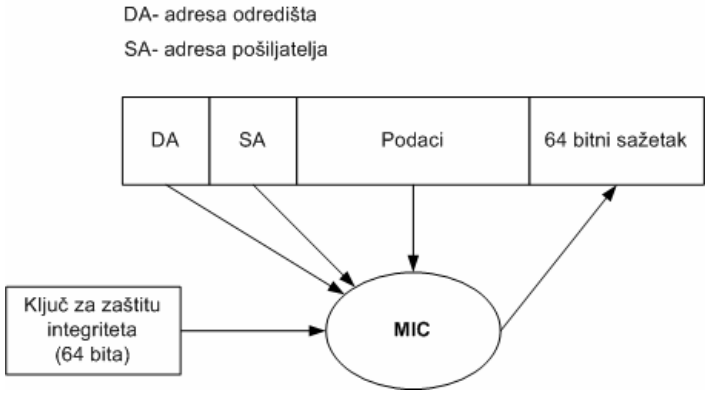
Način dobivanja inicijalizacijskog vektora je drugačije izveden nego kod WEP-a. Inicijalizacijski vektor se dobiva tako da se uzmu prvi i treći byte od staroga inicijalizacijskog vektora te se uvećava za jedan byte. Inicijalizacijski vektor se postavi na nulu kada se vremenski ključ promijeni ili se inicijalizira.
Za sigurnost je veoma bitan način biranja ključeva koji će se koristiti. TKIP ima ugrađen mehanizam odabira ključeva. Kada se klijent autetificira tada se dobije glavni ključ(*Master Key*). Iz glavnog ključa se izvodi izvedeni glavni ključ(*Pairwise Master Key - PMK*) koji će se koristiti u daljnjoj derivaciji ključeva. Iz njega se izvodi prijelazni ključ(*Pairwise Transient Key -PTK*) koji služi pri dobivanju ključeva potrebnih za enkripciju i zaštitu integriteta poruka i podataka. Grafički prikaz je na slici 6.2.



Slika 6.2: Prikaz odabira ključeva

Postoje i dvije vrste ključeva koje se koriste u protokolu. Ključ za enkripciju je dijeljen između pristupne točke i klijenta dok je ključ za zaštitu integriteta različit te manje duljine-64 bita

Bolja zaštita integriteta se postiže MIC(*Message Integrity Code,* negdje se naziva i *Michael*) algoritmom. Korištenje pravih kriptografskih funkcija za zaštitu integriteta(*hash* funkcije) nije dolazilo u obzir jer sklopovlje koje se nalazi u mrežnoj opremi ne bi to moglo podnijeti bez značajnijeg pada performansi mreže. Algoritam ne smije trajati duže od 5 instrukcija po oktetu podataka. No to donosi i neke probleme jer je osjetljiv na napade- naime postoji 2^{29} napada na njega. U ovaj algoritam je ugrađena i logika kojom se brani od aktivnih napada. Glavna pretpostavka je da paketi na prijemnu stranu stižu po redu pa kada prijemna strana otkrije da paket koji je dobila nije onaj koji je očekivala, ona ga odbaci te promijeni vremenski ključ i postavi na nulu inicijalizacijski vektor. Također se odredi da će se generiranje novih ključeva odvijati svake minute. Na taj način se učinkovito sprječavaju aktivni napadi. Shematski se algoritam može prikazati slikom 6.3.



Slika 6.3: Shematski prikaz rada MIC-a

TKIP uz već spomenute, ima ugrađene i druge zaštitne mjere:

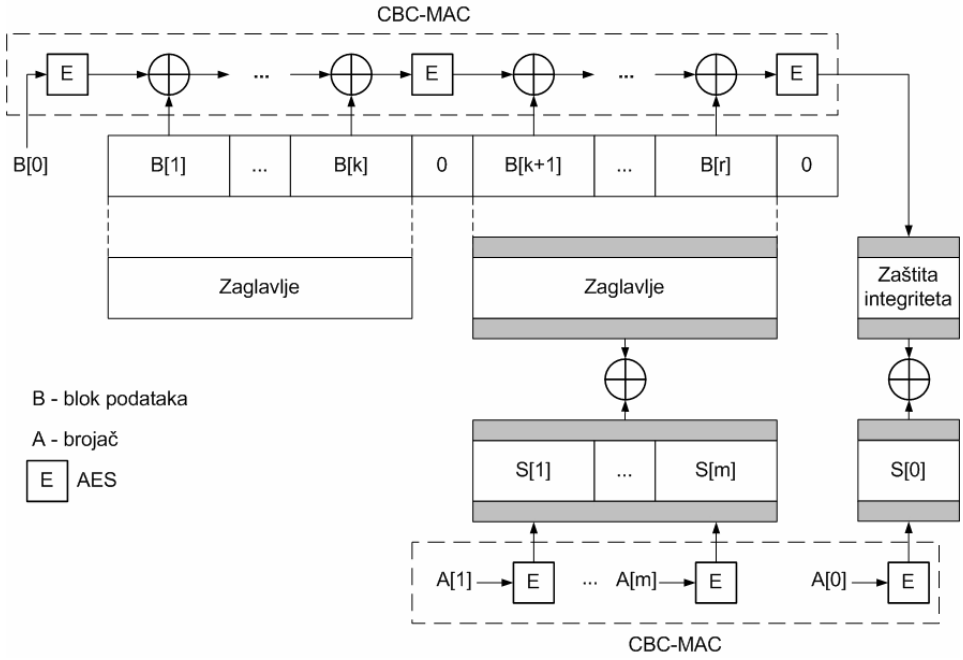
- Ukoliko se iscrpi brojač paketa(veličine 16 bita), a u međuvremenu se klijent i pristupna točka ne uspiju dogovoriti oko novog vremenskog ključa(*Temporal Key*) tada se prekida veza.
- Ukoliko se ne uspije obnoviti vremenski ključ komunikacija se zaustavlja ili se trajno prekida.

Kako se vidi MIC je, iako u kriptografskom smislu još uvijek loše rješenje u odnosu na prave funkcije za zaštitu integriteta, no još uvijek je značajan korak naprijed u odnosu na CRC32 funkciju koja se koristi u WEP-u.

Bežična mreža po svojoj prirodi osigurava korisnicima gotovo neograničenu mobilnost. U tome ih ne bi trebao sprječavati ni standard pa je u TKIP ugrađen mehanizam koji dopušta da klijenti mijenjaju pristupne točke bez da se moraju ponovno autentificirati. Dakle postoji mehanizam kojima pristupne točke međusobno komuniciraju. TKIP ima ključ kojim enkriptira poruke koje šalje svim stanicama koje su priključene na određenu pristupnu točku i to je tzv. *multicasting key*. Ovaj ključ se šalje klijentu kada se uspostavi veza između njega i pristupne točke. Kada klijent mijenja pristupnu točku ona šalje novi ključ enkriptiran pomoću staroga svim klijentima koji su na nju priključeni. Kada svi klijenti dobiju novi ključ pristupna točka započinje koristiti novi ključ u komunikaciji sa svim klijentima. U standardu je podržano da istovremeno se može pohraniti do četiri ključa.
TKIP je veliko unaprjeđenje sigurnosti u odnosu na WEP. No kako je zbog sklopovlja ograničen ne predstavlja zadovoljavajuće trajno rješenje problema sigurnosti bežičnih mreža i kao takav može biti samo međukorak između WEP-a i robusnog standarda koji bi zadovoljio sve sigurnosne zahtjeve koji se postavljaju pred njega.

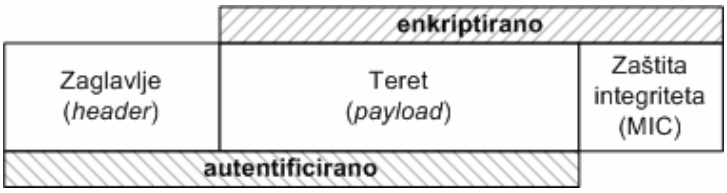
6.3.2 CCMP

CCMP se smatra boljim i trajnim rješenjem problema zaštite podataka u bežičnim računalnim mrežama. CCMP se temelji na AES-u(*Advanced Encription Standard*) u CCM(*Counter Mode Encryption with CBC-MAC Data Origin Authenticity*) načinu rada. CCM način rada podrazumijeva da se koristi jedan ključ za enkripciju i zaštitu integriteta podataka tj. u CCM načinu rada se paket enkriptira i autentificira odjednom. CCM je specijalno dizajniran za 802.11i standard i predviđen je za rad samo sa blokom podataka i ne postoje planovi da se preinači da radi sa tokovima podataka. CCM radi sa 128 bitnim blokom podataka. U CBC-MAC načinu rada se pomoću AES-a enkriptira prvi blok podataka te se provodi operacija ekskluzivno-ili sa drugim blokom podataka te se onda ponovno enkriptira i tako dalje sve do zadnjeg bloka podataka. Na taj način je dobiven 64 bitni rezultat koji se dodaje na kraj paketa podataka koji se šalju i uloga mu je zaštita integriteta. To se može prikazati slikom 6.4.



Slika 6.4: Prikaz rada CCM-a

Nakon toga se teret paketa skupa sa dodanom zaštitom integriteta enkriptira AES-om u CTR(*counter mode*) načinu rada. U CTR načinu rada se proizvoljno odabire neka vrijednost, koja se naziva brojač(*counter*), i nju se enkriptira AES-om. Nakon toga se provodi operacija ekskluzivno-ili dobivene vrijednosti i bloka podataka koji se enkriptira. Nakon svakog enkriptiranog bloka se brojač povećava za 1 te se dalje nastavlja proces enkripcije. Za enkripciju tereta se kao brojač uzimaju vrijednosti 1,2,3,… dok je za enkripciju bitova zaštite integriteta brojač jednak 0. CTR način rada ima funkciju zaštite privatnosti podataka.



Slika 6.5: Shematski prikaz enkriptiranog i zaštićenog paketa podataka

CCM način rada može ostaviti proizvoljan dio teksta u čistom obliku tj. neće ga autentificirati. To je važno jer neka polja u zaglavlju paketa se mijenjaju kako paket putuje mrežom(npr. pri usmjeravanju se mijenja MAC adresa odredišta) pa se neće dogoditi da zbog toga što je paketu mijenjana adresa on bude odbačen kao lažan.
Sada protokol CCMP koristi CCM kako bi zaštitio privatnost i integritet podataka. Ključ koji se koristi u enkripciji/dekripciji podataka je 128 bitni vremenski ključ(*Temporal key*) kojega znaju i klijent i pristupna točka.
802.11i standard uvjetuje obavezno korištenje CCMP-a u svim njegovim implementacijama.

Kako vidimo CCMP je ogromno poboljšanje sigurnosti u odnosu na TKIP, a posebno na WEP. On nagovještava novu eru u sigurnosti bežičnih računalnih mreže. Jedini nedostatak je to što se ne može implementirati u postojeću opremu nego se ona mora zamijeniti novijom opremom koja će biti dovoljno sposobna nositi se sa AES-om bez degradacije performansi.

6.3.3 WRAP

WRAP je bio prvi prijedlog za korištenje u 802.11i standardu no došlo je do nekih pravnih problema pri njegovoj implementaciji u 802.11i standard. Naime čak tri tvrtke su podnijele zahtjev za patentiranje WRAP-a što je uzrokovalo da se umjesto WRAP-a koristi CCMP.
WRAP koristi, kao i CCMP, AES no u OCB(*Offset Codebook*) načinu rada. OCB se smatra jednako sigurnim kao i CCMP način rada AES-a. Iako su neki proizvođači ugradili podršku većina ih ipak koristi CCMP. Korištenje WRAP-a je opcionalno u standardu.

6.3.4 Prethodno postavljeni ključevi(Pre-shared keys)

Standard bi trebao omogućiti autentifikaciju i u ad hoc mrežama računala(računala ne komuniciraju preko pristupne točke već izravno jedno sa drugim). To je i ostvareno u 802.11i standardu tako da je uveden koncept prethodno postavljenih ključeva(*Pre-shared keys*) tako da nema potrebe koristiti 802.1X protokol tj. u mreži nije potreban RADIUS poslužitelj. Postoje dva načina na koja se može ostvariti koncept prethodno postavljenih ključeva. Prvi način je da postoji samo jedan ključ koji svi poznaju i kojime štite podatke i taj način se smatra nesigurnim. Drugi sigurniji način uključuje to da svaki par računala u mreži ima svoj ključ i njega koriste pri međusobnoj komunikaciji.
Ovaj način ima primjenu osim u ad hoc mrežama i u malim kućnim i uredskim mrežama kojima se ne bi isplatilo ulaganje u RADIUS poslužitelj.

7. PRAKTIČNI RAD

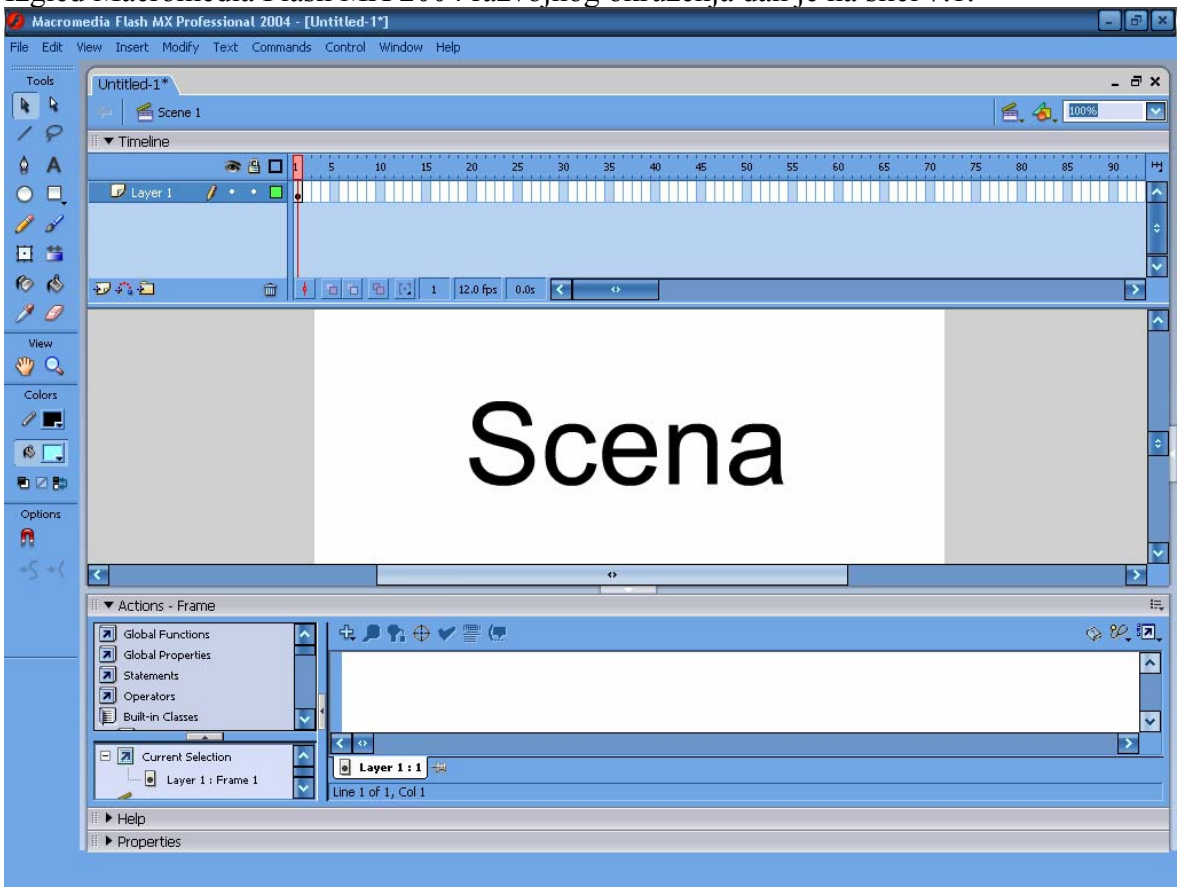
Praktični rad se sastoji od nekoliko prezentacija izrađenih u Flash tehnologiji. Naglasak je stavljen na procese autentifikacije i oni su shematski prikazani prezentacijama.

7.1 Macromedia® Flash™ MX 2004

Macromedia Flash MX 2004 je skup alata koji omogućavaju izradu Flash aplikacija svih namjena. Aplikacije izrađene u Flashu pokreću se pomoću Macromedia Flash Player-a koji postoji za sve platforme i besplatan je za sve korisnike. Flash animacija se sastoji od niza okvira (*frame*) unutar kojih se nalazi jedan ili više slojeva (*layer*). Unutar slojeva se nalaze razni objekti (npr. slike, kontrole za interakciju sa korisnikom, razna tekstualna polja itd.).
Glavni dijelovi Macromedia Flash MX 2004 razvojnog okruženja su:

- Vremenska linija (*timeline*)
  Prikazuje poredak okvira u vremenu te omogućava njihovo raspoređivanje u vremenu.
- Scena (*stage*)
  Scena je zapravo izgled trenutnog okvira. Sastoji se od najmanje jednog sloja. Na sloju se nalaze razni objekti.
- Alatna traka (*tools*)
  Sadržava razne alate za izradu i editiranje vektorske grafike.
- Traka za dodavanje akcija (*actions*)
  Omogućava pridruživanje određenih akcija nekom objektu unutar određenog okvira. Akcije se opisuju u posebnom jeziku – ActionScript.

Izgled Macromedia Flash MX 2004 razvojnog okruženja dan je na slici 7.1.



Slika 7.1: Izgled Macromedia Flash MX 2004 razvojnog okruženja

7.2 ActionScript 2.0

ActionScript je programski jezik temeljen na objektno-orijentiranoj paradigmi (od verzije 2.0), a koristi se pri izradi aplikacija u Flashu. Pomoću njega se postiže dodatna funkcionalnost aplikacija rađenih u Flashu.
Za primjer uzimo izradu jednostavne navigacije unutar Flash prezentacije. Kako bi korisnik bio u mogućnosti upravljati izvođenjem prezentacije potrebno je unutar aplikacije dodati navigacijske tipke (*button*). Primjerice, uzmimo tipku koja korisniku omogućava pokretanje aplikacije (*play button*). Kako bi se aplikacija pokrenula kada korisnik klikne na tipku *play* potrebno je odgovarajućem okviru, na kojem se nalazi tipka *play*, dodati slijedeći kod:

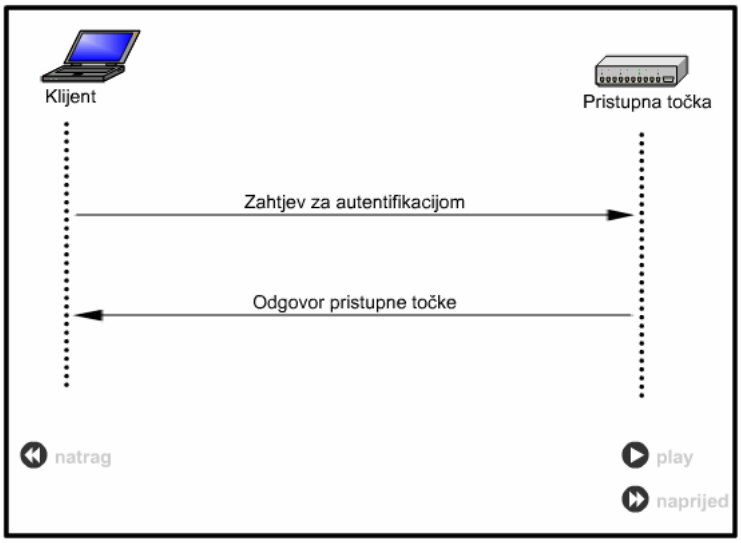
```
play_btn.onRelease=function(){
//play_btn je ime instance. OnRelease je događaj koji pokreće
//izvođenje funkcije
      play();
      //ugrađena naredba koja započinje izvođenje aplikacije
}
```

Sada korisnik klikom na tipku *play* pokreće izvođenje aplikacije. Na sličan način je moguće ostvariti i ostale tipke koje se koriste u ovom primjeru jednostavne navigacije.

7.3 Prezentacije

Ovdje će biti navedene i kratko opisane svaka od prezentacija koje su izrađene kao praktični rad. Njihova svrha je shematski prikazati pojedini dio seminara.

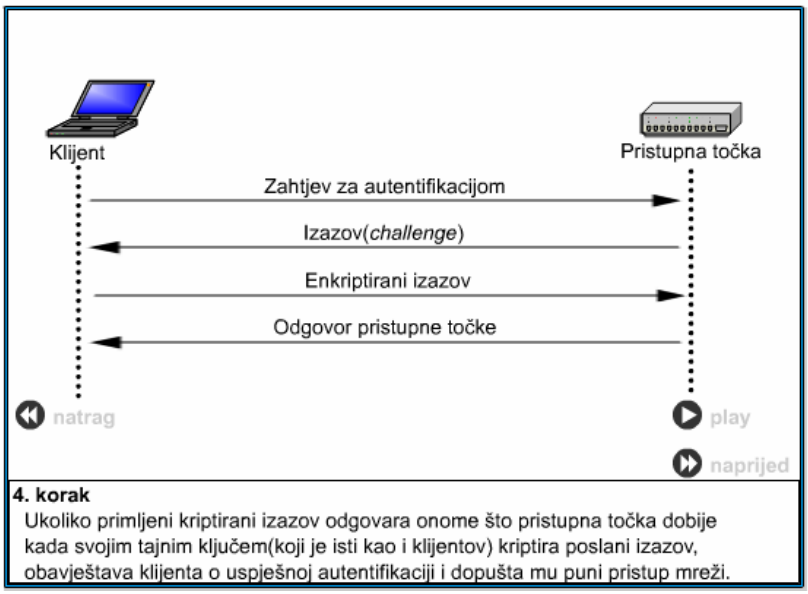
Autentifikacija otvorenog sustava (*Open System Authentication*)
Autentifikacija otvorenog sustava je detaljnije opisana u poglavlju 3.3.1. U prikazu autentifikacije otvorenog sustava razlikujemo dva entiteta: klijenta i pristupnu točku. Na slici 7.1 je prikazan dio prezantacije.



Slika 7.2: Dio prezentacije autentifikacije otvorenog sustava

Autentifikacija temeljena na dijeljenoj tajni (*Shared Key Authentication*)
Autentifikacija temeljena na dijeljenoj tajni je opisana u poglavlju 3.3.2. Kao i kod prethodne autentifikacije sudjeluju dva entiteta: klijent i pristupna točka. Ova prezentacija se po izvedbi razlikuje od prethodne jer je osim osnovnih objekata(klijent i pristupna točka

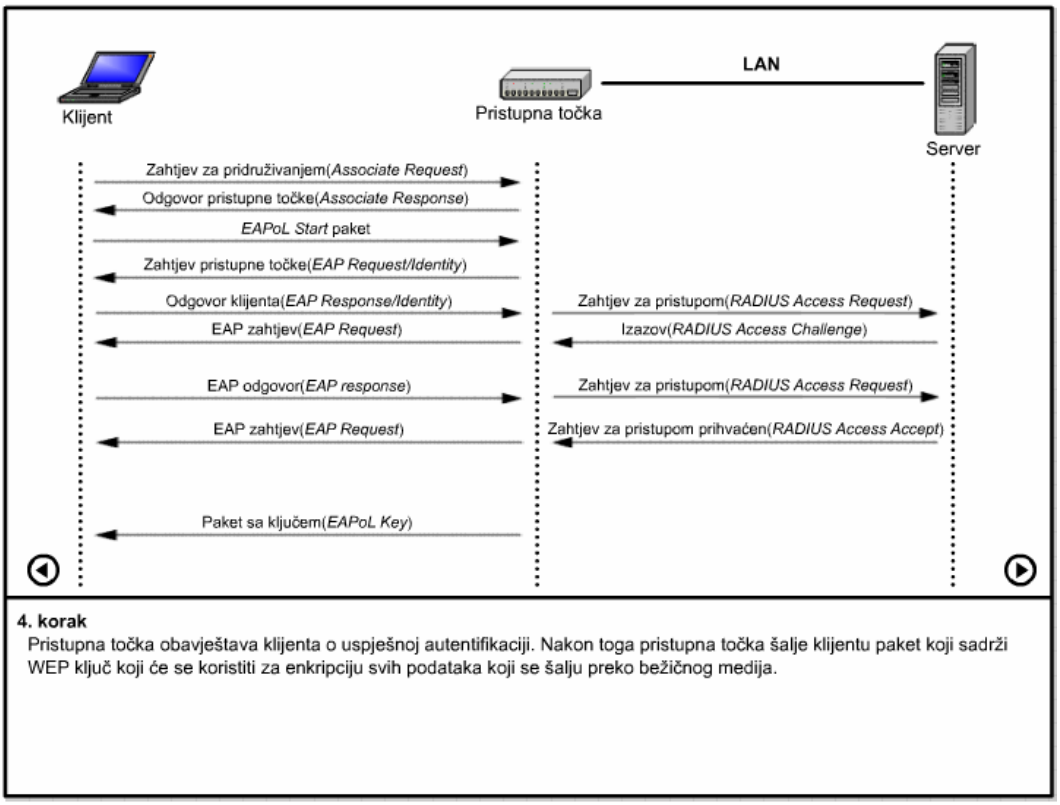
te poruke koje razmjenjuju) i navigacije, dodana i ploča sa opisom komunikacije između klijenta i pristupne točke. Dio prezentacije je prikazan na slici 7.3.



Slika 7.3: Dio prezentacije autentifikacije dijeljenim ključem

EAP autentifikacija

Proces autentifikacije korištenjem EAP-a je opisan unutar poglavlja 5.1. U procesu autentifikacije sudjeluju tri entiteta: klijent (*supplicant*), pristupna točka te pozadinski autentifikacijski server. Prezentacija shematski prikazuje proces autentifikacije. Dio prezentacije je prikazan na slici 7.4.



Slika 7.4: Dio prezentacija EAP autentifikacije

Na slici su vidljivi entiteti koji sudjeluju u procesu autentifikacije kao i navigacijski elementi te ploča sa opisom.

8. ZAKLJUČAK

U sadašnjem trenutku bežične računalne mreže su najveći sigurnosni rizik za svakoga tko ih koristi. Uzrok tome svakako treba tražiti u trenutno važećim standardima 802.11a, 802.11b i 802.11g koji kao uporište sigurnosti uzimaju WEP(*Wired Equivalent Privacy*). WEP nije zadovoljio niti jedan od tri cilja s kojim je stvoren: pouzdana autentifikacija korisnika, zaštita privatnosti podataka te autorizacija korisnika. Impresivan broj različitih mogućih napada, od kojih su neki i praktično uspješno izvedeni, samo potvrđuje činjenicu o nesigurnosti sadašnjeg standarda. Zbog toga se većina organizacija koje intenzivno koriste bežične mreže u svome poslovanju odlučila na skupo uvođenje IPsec tehnologije koja, iako ne bez mana, ipak značajno povećava sigurnost bežične mreže. IEEE uvidjevši sve propuste u važećim standardima je počela ubrzano raditi na novim prijedlozima za poboljšanje sigurnosti bežičnih mreža. Prvi rezultat toga je i donošenje 802.1X standarda koji uvelike poboljšava način autentifikacije korisnika i time povećava ukupnu sigurnost. Taj standard kao svoju osnovicu koristi EAP koji omogućava velik broj različitih metoda autentifikacije korisnika mreže. Slijedeći korak u evoluciji sigurnosti je WPA standard koji je donesen od strane udruženja proizvođača mrežne opreme za bežične računalne mreže(*Wi-Fi Alliance*). WPA je otklonio sve sigurnosne propuste u WEP-u, a pri tome uz manje preinake pogonskih programa radi i na postojećoj opremi za bežične računalne mreže što je još jedan plus s obzirom da nisu potrebna velika ulaganja(kao npr. kod uvođenja IPsec tehnologije) kako bi se povećala sigurnost. Iako veliko poboljšanje WPA se smatra samo međukorakom između 802.11x standarda i najnovijeg 802.11i standarda koji se nameće kao konačno rješenje problema sigurnosti bežičnih računalnih mreža. Sam 802.11i standard je dio većeg RSN(*Robust Security Network*) standarda. 802.11i koristi kao enkripcijski algoritam AES(*Advanced Encription System*), a kao mehanizam autentifikacije se, iako 802.11i standard ne propisuje, koristi 802.1X standard. No za uvođenje 802.11i standarda u mreže je potrebno zamijeniti svu sadašnju bežičnu mrežnu opremu(pristupne točke i mrežne kartice) sa novom jer je sadašnja preslaba.
Može se zaključiti da korisnike bežičnih računalnih mreža unatoč svemu ipak očekuje svjetla, sigurnija, budućnost.

Literatura:

[1] Fluher, Scott, Itsik Mantin and Adii Shamir, *Weaknesses in the Key Scheduling Algorithm of RC4* ,www.drizzle.com/~aboba/IEEE/rc4_ksaproc.pdf.
[2] Nikita Borisov, Ian Goldberg, David Wagner, *The Insecurity Of 802.11*
[3] Arbaugh, W., Mishra, A., "An Initial Security Analysis of the 802.1X Standard", http://www.cs.umd.edu/%7Ewaa/1x.pdf.
[4] Niels Ferguson, Bruce Schneier, *A Cryptographic Evaluation Of Ipse,* http://www.counterpane.com/
[5] Judy Ma, Ashley Tan, *Wi-Fi Security Overview,* SIMS 219, listopad 2003
[6] Aboba, B. *"WEP2 Security Analysis: IEEE 802.11-00/253"*, http://www.drizzle.com/~aboba/IEEE/11-01-253r0-IWEP2SecurityAnalysis.ppt
[7] Stark, T. *"WEP2, Credibility Zero."*, www.dnai.com/~thomst/wireless003.html.
[8] EAP Working Group,"State Machines for Extensible Authentication Protocol (EAP) Peer and Authenticator draft-ietf-eap-statemachine-03", http://www.ietf.org/internet-drafts/draft-ietf-eap-esteem-01.txt
[9] EAP Working Group, "Internet draft.ietf-eap-rfc2284bis"
[10] J. Walker, *"802.11i Overview part 1"*