SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA
ZAVOD ZA ELEKTRONIKU, MIKROELEKTRONIKU, RAČUNALNE I INTELIGENTNE SUSTAVE

Proširiv autentifikacijski protokol

Jelena Vučak

SEMINARSKI RAD

Zagreb, 2006.

Sadržaj

1. Uvod

Ovaj rad se bavi programskom komponentom pod nazivom *WPA_supplicant* tj. autentifikacijom koju ta komponenta omogućava. Definicija komponente *WPA_suplicant*, njena struktura i okruženje u kojem se komponenta primjenjuje opisani su u 2. poglavlju.

Kao autentifikacijsku metodu *WPA_suplicant* koristi proširiv autentifikacijski protokol tj. EAP (*engl. Extensible Authentication Protocol*). Protokol EAP podržava mogućnost autentifikacije s nekoliko različitih EAP metoda. Svaka od tih metoda posebna je po formatu paketa kojima prenosi poruke, ali i po svojem načinu provođenja autentifikacije. Neke EAP metode i protokol EAP su opisani u 3. poglavlju.

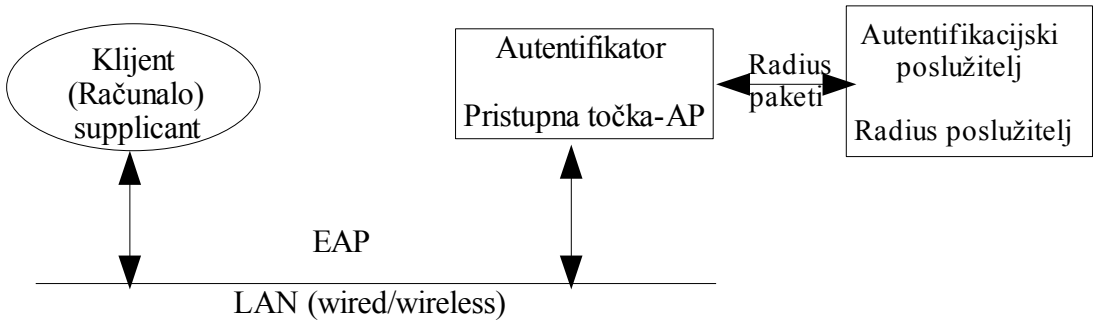
Ostatak ovog dokumenta obuhvaća razradu komponente *WPA_supplicant*. To podrazumijeva opis ostvarenja (implementacije) modula. Moduli su razrađeni u 4. poglavlju. Podijeljeni su u četiri skupine i tim redom su i razrađeni. Započinje se opisom kontrolnog sučelja (ctrl if), potom automata stanja (state_machines), a na kraju je opis sučelja prema mrežnim sučeljima (driver if) i l2_layer sučelja.

Posljednje, 5. poglavlje, prikazuje praktični rad, kojeg čini korištenje *WPA_supplicant-a* pri bežičnom spajanju na dvije pristupne točke, koje koriste različite metode zaštite. Prikazano je spajanje na pristupnu točku koja koristi WPA-PSK (*engl. WPA Pre Shared Key*) metodu zaštite te spajanje na pristupnu točku s EAP-MSCHAPv2 (*engl. EAP Microsoft Challenge Handshake Authentication Protocol version 2*) metodom zaštite.

2. Sigurnosna komponenta *WPA_supplicant*

WPA_supplicant je komponenta koja se izvršava u stanicama klijenata i zadužena je za njihovu autentifikaciju kako bi klijenti mogli pristupiti računalnoj mreži. Naziv je sročen od dviju riječi WPA i supplicant. WPA je skraćenica od *Wi-Fi Protected Access* i označava zaštitnu enkripcijsku metodu, a koristi se za zaštitu bežičnih mreža. Ta je metoda nastala nakon probijanja WEP-a (*engl. Wired Equivalent Privacy*), pa se često spominje i kao WEP2. Bazira se na RC4 algoritmu i pruža bolju enkripciju (od WEP-a) jer se temelji na TKIP (*engl. Temporal Key Integrity Protocol*) protokolu, u kojem se ključ kriptiranja mijenja svakih 10000 paketa. Ova metoda za autentifikaciju koristi EAP (*engl. Extensible Authentication Protocol*). Riječ *supplicant* u nazivu označava komponentu u klijentu, kojoj je zadatak dogovaranje s autentifikatorom oko ključeva koji se upotrebljavaju za zaštitu, te autentifikacija i autorizacija sa bežičnim upravljačkim programima (*engl. wlan drivers*).

Okruženje u kojem se upotrebljava *WPA_supplicant* prikazano je na slici 2.1. Na slici se vidi da se u procesu autentifikacije javljaju tri strane tj. tri sudionika. Prvi sudionik je klijent, kojeg predstavlja računalo (prijenosno ili stolno), a u kojem se nalazi supplicant. Klijent je spojen na lokalnu mrežu (žično ili bežično). Na lokalnu mrežu je spojen i autentifikator, kojeg na slici predstavlja pristupna točka (*engl. access point – AP*). Treći sudionik je autentifikacijski poslužitelj, primjerice radius poslužitelj. EAP protokol se koristi između klijenta i pristupne točke (AP), preko lokalne mreže, dok se između autentifikatora i autentifikacijskog poslužitelja razmjenjuju radius paketi.



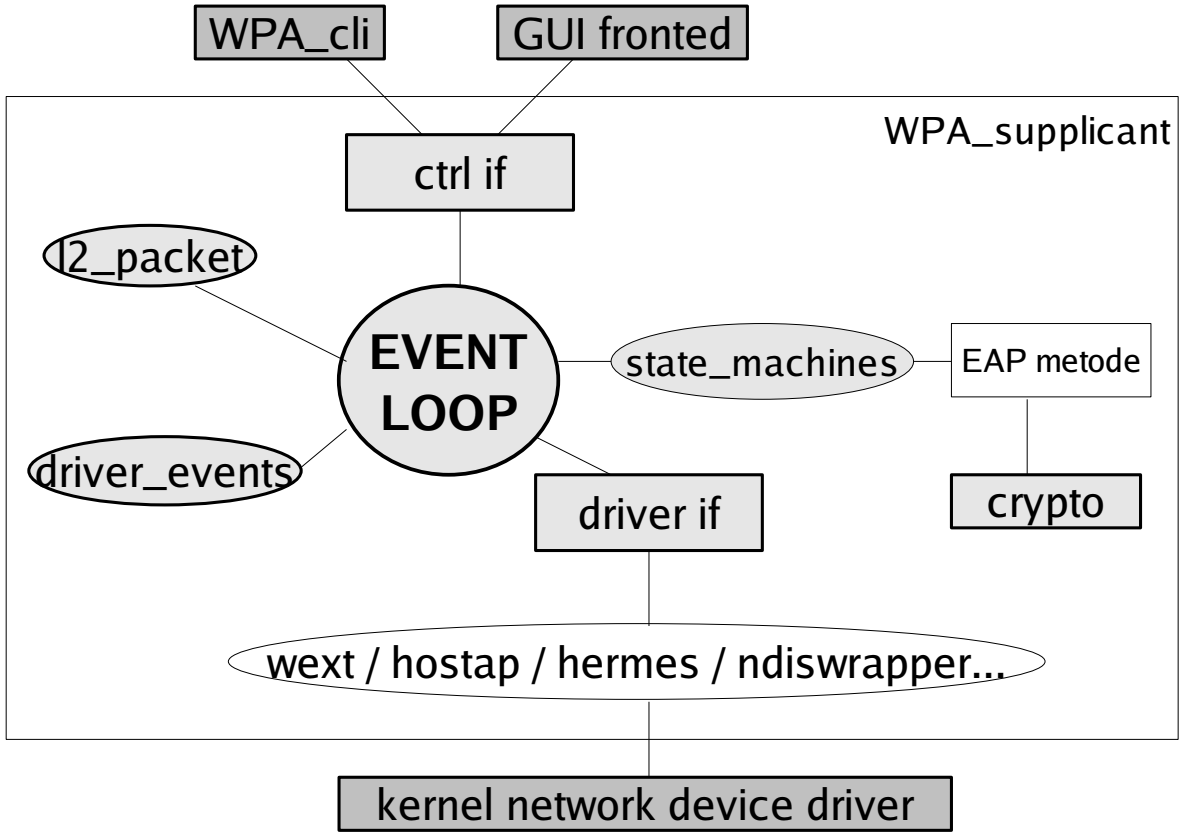
Slika 2.1. *WPA_supplicant* u stvarnosti

Na prikazanoj slici predstavljena je ideja korištenja *WPA_supplicant-a*. Međutim, korištenje *WPA_supplicant-a* je znatno složenije, a programska realizacija poprilično složena. Razlog tomu je što se *WPA_supplicant* sastoji od nekoliko, međusobno povezanih, dijelova. Glavni dio čini autentifikacijski protokol EAP tj. automat stanja EAP protokola (EAP state machine). EAP koristi mnoge metode autentifikacije, a one pak koriste mnoge kriptografske algoritme (EAP metode, crypto, TLS). Uz automat stanja EAP-a, tu su još i automat stanja WPA te EAPOL-a (*engl. EAP over LAN*). Za komunikaciju s vanjskim programima *WPA_supplicant* koristi kontrolno sučelje, kojeg upotrebljavaju programi za konfiguraciju. Tako se razlikuju wpa_cli i wpa_gui. Kontrolnim sučeljem omogućava se vanjskim programima dohvat informacija o rezultatima operacija koje se u *WPA_supplicant-u* događaju (proces autentifikacije). Uz do sada spomenute module, važan modul predstavlja i sučelje prema mrežnim sučeljima (*engl. Driver wrapper implemetation*). Spomenuti modul sadrži skup manjih datoteka s implementiranim generičkim kodom, kojim se zamijene dijelovi upravljačkih programa (*driver-a*) ovisni o raspoloživom sklopovlju, programskoj podršci ili operacijskom sustavu. Posljednji modul važan za spomenuti jest l2_layer packet, koji predstavlja sučelje za razmjenu paketa s 2. slojem. Svi do sada spomenuti moduli, ali i oni koji nisu spomenuti, a važni su za opis rada *WPA_supplicant-a* bit će detaljnije objašnjeni u sljedećim poglavljima.

Cilj ovog seminarskog rada je upoznavanje sa *WPA_supplicant-om*, kako bi se napravio temelj koji će omogućiti spajanje *WPA_supplicant-a* s implementacijom IKEv2 protokola tj. korištenjem *WPA_supplicant-a* kao autentifikacijske metode IKEv2 protokola.

Radi se o ostvarenju tzv. *road-warrior* načina rada. U tom načinu rada računalo, koje se može nalaziti bilo gdje na Internetu, pristupa lokalnoj mreži tvrtke na takav način da postaje njen sastavni dio. Korisniku računala izgleda kao da je spojen na mrežu unutar tvrtke. Kako zaposlenik treba pristupiti povjerljivim podacima svoje tvrtke i to putem nesigurne mreže kao što je Internet, koristi se *IPsec*, tj. *IKEv2* (*engl. Internet Key Exchange*) i *ESP* (*engl. Encapsulating Security Protocol)* protokoli kako bi se osigurala sigurna veza. Kako bi se spriječio pristup neautoriziranim korisnicima na lokalnu mrežu tvrtke koristi se autorizacija. IKEv2 podržava autentifikaciju putem dijeljene tajne, odnosno putem certifikata, ali također podržava mogućnost korištenja EAP protokola. Kako bi se realizirala mogućnost autentifikacije putem EAP protokola koristiti će se *WPA_supplicant*. Za ostvarenje ideje korištenja *WPA_supplicant-a* u implementaciji IKEv2 protokola bilo je potrebno detaljno proučiti pojedine module supplicant-a. Rezultati tog proučavanja sažeti su u ovom dokumentu.

U nastavku teksta slijedi opis proširivog autentifikacijskog protokola (EAP protokola), kojeg *WPA_supplicant* koristi, a zatim opis ostvarenja (implementacije) modula. Moduli su razrađivani redom, kojim su prikazani na slici 2.2 koja predstavlja ideju *WPA_supplicant-a*.



Slika 2.2. *WPA_supplicant*

Opisano je kontrolno sučelje (ctrl if), potom automati stanja (state_machines), sučelje prema mrežnim sučeljima (driver if) i l2_layer sučelje.

3. Protokol EAP

EAP *(engl. Extensible Authentication Protocol)*, proširiv autentifikacijski protokol, koji podržava različite autentifikacijske metode, opisan u RFC3748. Koristi se na podatkovnom sloju žičnih ili bežičnih mreža. Do sada je EAP implementiran uz usmjernike i računala povezane preko prospojnih mreža. Kod bežičnih mreža implementira se u pristupnim točkama i preklopnicima *(engl. Switch)*. Prednost ovog autentifikacijskog protokola je njegova proširivost, koja se očituje u izboru autentifikacijske metode koja će se koristiti. One se mogu naknadno dodavati bez potrebe za promjenom kompletnog koda pristupne točke. Korištenjem EAP-a nije potrebno da *Autentifikator* podržava sve autentifikacijske metode, nego se koristi autentifikacijski poslužitelj (*engl. Authentication server*). Taj poslužitelj implementira sve (ili samo neke) metode, a autentifikator se onda ponaša kao prijelazni poslužitelj *(engl. Pass-through)* za metode. Može se reći kako autentifikacijski poslužitelj obavlja autentifikaciju za autentifikatora tj. obavlja autentifikacijske metode za autentifikatora. Uz autentifikatora i autentifikacijskog poslužitelja, kod EAP-a treba poznavati i pojam *Supplicant* (još se naziva i partner (*peer*)). Kako je autentifikator dio koji pokreće EAP autentifikaciju, supplicant je dio koji odgovara autentifikatoru. Dakle, postoji veza između partnera i autentifikatora i omogućena im je komunikacija. Komunikacija podrazumijeva razmjenu poruka između njih, što čini autentifikaciju. Razmjena poruka detaljnije je objašnjena u sljedećem poglavlju.

Prednosti EAP protokola su što, kao što je već rečeno, omogućava upotrebu različitih metoda autentifikacije.

Također, jedna od prednosti je što uređaji za pristup mrežnom poslužitelju (npr. pristupne točke, preklopnici) ne moraju poznavati svaku metodu autentifikacije. Tada se ponašaju kao prijelazni dio za "stražnje" autentifikacijske poslužitelje.

Nedostaci EAP protokola su što, primjerice za korištenje u Point To Point protokolu (PPP) potrebno je dodati novi autentifikacijski tip PPP LCP, a i implementaciju PPP-a je potrebno prilagoditi za taj tip. Ukoliko autentifikator nije odvojen od "stražnjeg" autentifikacijskog poslužitelja, mogu nastupiti poteškoće pri razmjeni ključeva.

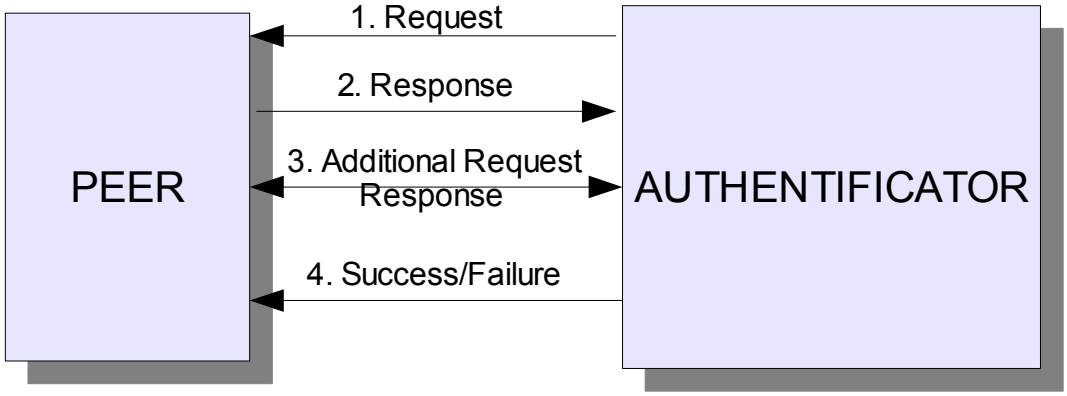
3.1. Razmjena poruka EAP protokolom

Razmjena poruka između partnera i autentifikatora odvija se sljedećim redoslijedom:

1. Autentifikaciju započinje autentifikator slanjem zahtijeva za autentifikacijom (*engl. Request*) partneru. Ta poruka sadrži polje *Type* koji određuje kakva je vrsta zahtijeva (npr. Identity, MD5 – Challenge).
2. Zatim slijedi odgovor partnera (*engl. Response*). Kao i zahtijev, odgovor sadrži polje *Type*, po kojem se prepoznaje da je ta poruka odgovor na prije poslani zahtijev.
3. Autentifikator zatim šalje dodatne poruke (zahtijeve), a partner odgovara na zahtijeve. Izmjena zahtijeva i odgovora nastavlja se do daljnjega tj. dok se ne autentificira partner. Novi zahtijev ne može se poslati dokle god nije stigao odgovor na prije poslani zahtijev. To vrijedi za sve zahtijeve, osim početnog, kojim autentifikator započinje proces autentifikacije.
4. Uspješnom autentifikacijom se smatra kada autentifikator autentificira partnera i tada može poslati poruku o uspješnoj autentifikaciji (*Success*).

   Ukoliko autentifikator nije primio odgovore na nekoliko odaslanih zahtijeva, autentifikacija partnera se smatra neuspješnom i autentifikator tada šalje poruku o neuspješnoj autentifikaciji (*Failure*).

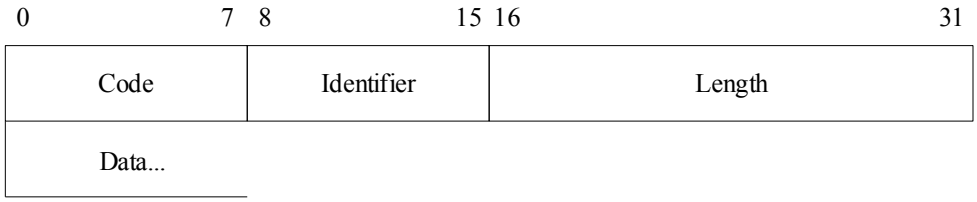
Razmjena EAP paketa prikazana je na slici 3.1



Slika 3.1. Razmjena EAP paketa

3.2. Format EAP paketa

Format EAP paketa prikazan je na slici 3.2.



Slika 3.2. Osnovni format EAP paketa

Prvi oktet EAP paketa zauzima polje *Code*, koje sadrži tip (*Type*) EAP paketa. Moguće vrijednosti polja *Code* su prikazane u tablici 1

Tablica 1: Vrijednosti polja *Code* EAP paketa

1	Request
2	Response
3	Success
4	Failure

EAP protokol podržava samo gore navedene vrijednosti polja *Code*, te svi paketi s vrijednostima tog polja različitim od gore navedenih vrijednosti trebaju biti odbačeni.

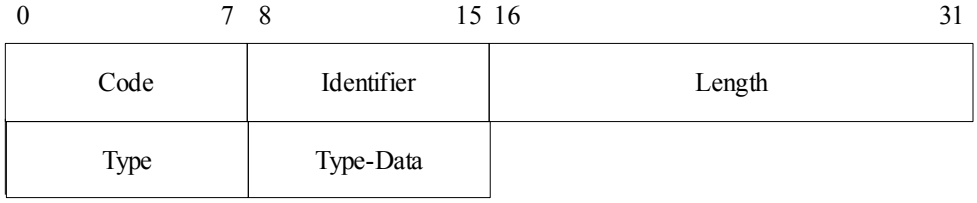
Sljedeći oktet zauzima polje *Identifier*, a zadatak mu je povezivati upite zahtjeva i odgovora na njih.

Polje *Length* sadrži vrijednost koja označava duljinu EAP paketa (u oktetima). Duljina EAP paketa uključuje polja *Code*, *Identifier*, *Length* i *Data*. Svi okteti koji prelaze vrijednost navedenu u polju *Length* se po primitku odbacuju, zanemaruju i smatraju se *padding-om* podatkovnog sloja.

Nakon prva 4 okteta slijede podaci u polju *Data*. Duljina tog polja varira i može poprimiti vrijednosti 0 ili nekoliko okteta. Tip podataka EAP paketa ovisi o vrijednosti u polju *Code*, odnosno o tipu EAP paketa.

EAP paketi se mogu podijeliti u dva "podtipa": Request/Response tip paketa i Success/Failure tip paketa. Request/Response tip paketa u polju *Code* sadrži vrijednost 1(Request) tj. 2(Response).

Request šalje autentifikator, a namijenjen je kao upit partneru za autentifikacijom. Response paket je odgovor partnera autentifikatoru. Na slici 3.3 je prikazan format Request/Response tipa paketa. U odnosu na općeniti format EAP paketa, ovaj se format razlikuje samo u polju *Type*. To polje je duljine jednog okteta, a sadrži heksadecimalnu vrijednost koja označava različite tipove Request/Response paketa. Različitim EAP paketima, određene su i različite EAP metode, o čemu će biti više riječi kasnije.



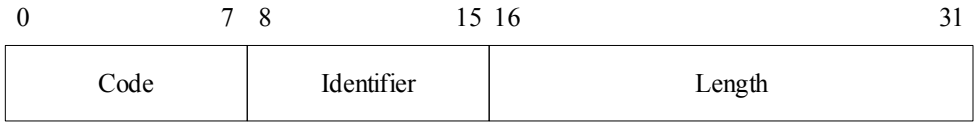
Slika 3.3. Format Request/Response paketa

Neke vrijednosti polja *Type* su prikazane u tablici 2:

Tablica 2: Različiti tipovi EAP paketa, ovisno o polju *Type*

1	Identity
2	Notification
3	Nak (samo za Response pakete)
4	MD5-Challenge
5	OTP (One Time Password)
6	GTC (Generic Token Card)
13	TLS (Transport Level Security)
26	EAP MSCHAPv2
254	Expanded types
255	Experimental use

Sve EAP implementacije moraju podržavati tipove od 1 do 4, a trebale bi i tip 254. Nakon polja *Type*, a ovisno o njemu, slijedi polje *Type-Data*, koje sadrži podatke paketa tipa *Type*. Drugi tip EAP paketa, Success/Failure paket, je prikazan na slici 3.4.



Slika 3.4. Format Success/Failure paketa

Taj tip paketa sadrži samo tri polja, koja su ista i s istom funkcijom, kao i u općem prikazu EAP paketa. Treba napomenuti da Success paket u polju *Code* ima oznaku 3, a Failure paket ima oznaku 4.

U sljedećem poglavlju bit će prikazane neke EAP metode.

3.3. EAP metode

Kao što je već spomenuto EAP metode se razlikuju po tipu paketa tj. po polju *Type* u EAP paketu. Također je već spomenuto da sve EAP implementacije moraju podržavati tipove 1, 2, 3 i 4 tj. Identity, Notification, Nak i MD5-Challenge. Uz njih trebale bi podržavati i tip 254 tj. Expanded types. Uz te, već spomenute tipove, u ovom će poglavlju biti prikazani i neki drugi tipovi autentifikacijskih poruka (metode), koje su implementirane u *WPA_supplicant* programu. To su metode TLS, GTC, SIM, PSK, OTP i druge. Njihov tip se numerira počevši s 4.

3.3.1. Identity

Tip Request/Response paketa. Request predstavlja upit autentifikatora za identitetom partnera, a najčešće se šalje kao prvi paket, na početku autentifikacije. Odgovor partnera se šalje u Response paketu (tipa 1). Preporuča se Response paket tipa 1 (Identity) upotrebljavati za dogovor oko korištenja EAP metoda (sadrži broj/brojeve autentifikacijske/ih metode/a, s kojima partner želi provesti autentifikaciju). Identity Request i Identity Response paketi se šalju u čistom (nekriptiranom) obliku.

3.3.2. Notification

Tip RequestResponse paketa. EAP paket tipa 2 služi za prijenos poruke od autentifikatora do partnera.

3.3.3. NAK

Tip Response paketa, koji se koristi kada je potrebno autentifikatora obavijestiti kako se traženi tip autentifikacije (broj sadržan u Nak paketu) ne može prihvatiti tj. metoda provesti. Ukoliko Nak paket sadrži broj 0, tada to znači da ne postoji niti jedna željena metoda za provedbu autentifikacije, te da autentifikator više ne treba slati Request pakete.

3.3.4. EAP-MD5-Challenge

Tip Request/Response paketa. Request paket predstavlja poziv (*challenge*) partneru, a Response odgovor autentifikatoru, s tim da se koristi MD5 funkcija sažimanja.

3.3.5. EAP-OTP

Metoda EAP-OTP (*One Time Password*), tj. tip autentifikacije koja sprječava napad nastao prisluškivanjem (*engl. Eavesdropping*). Naime, prisluškivanjem mreže mogu se otkriti povjerljive informacije koje služe za autentifikaciju korisnika. Takve informacije su npr. login korisnika, zaporke korisnika i slično. S tim podacima omogućen je napadaču pristup računalnom sustavu, za kojeg nema dopušten pristup. Ova metoda koristi tajni *pass-phrase* korisnika, kojim se generira jednokratna (*single use*) zaporka korisnika. Dodatna sigurnost koju pruža ova metoda je što se ni u kojem trenutku korisnikov tajni pass-phrase ne treba prenositi mrežom i da nikakvi tajni podaci ne trebaju biti pohranjeni na nekom poslužitelju.

3.3.6. EAP-GTC

Metoda EAP-GTC (*Generic Token Card*) je tip koji se koristi s različitim *Token Card* implementacijama, a koje zahtjevaju unose korisnika. Odgovor (Response paket) sadrži Token Card, informacije potrebne za autentifikaciju. Polje *Type* u paketu je postavljeno na vrijednost 6.

3.3.7. EAP-TLS

Metoda EAP_TLS (*Transport Level Security*) za obostranu autentifikaciju omogućava mehanizme za zaštitu integriteta i razmjenu ključeva između dvije krajnje točke. Kao i kod svake autentifikacije EAP-TLS metoda započinje razmjenom Identity Request/Response. Nakon saznavanja identiteta partnera EAP poslužitelj šalje EAP-TLS Start paket, kojim započinje TLS komunikacija (u EAP polju *Type*, je postavljen tip EAP-TLS, vrijednost 13). Zatim slijedi *client_hello_handshake* poruka (unutar EAP Response paketa, također s poljem *Type* postavljenim na EAP-TLS), koja sadržava verziju TLS-a, random broj, skup kriptografskih metoda podržanih od strane klijenta. Poslužitelj odgovara svojim paketom (*server_hello_handshake*) koji sadrži TLS zapise (TLS cerifikat, ključ poslužitelja, zahtjev za certifikatom, itd.). Poslužitelj šalje zahtjev za certifikatom kako bi usporedio identitet partnera kojeg je primio u prvoj poruci (Identity) i identitetom sadržanim u certifikatu. U tu svrhu i partner šalje zahtjev za certifikatom. Svaka uspostava "veze" tj. komunikacije partnera i poslužitelja ima svoj vlastiti SessionID, te skup kriptografskih parametara (metoda, ključeva), koje partner šalje poslužitelju, a ovaj to prihvaća. Ako je postupak autentifikacije uspješno završen autentifikator (poslužitelj) šalje odgovor (Response paket) kojim završava komunikacija s *handshake* porukama. Partner tada šalje TLS paket bez podataka u polju *Data*, a EAP poslužitelj na to odgovara sa Success porukom. U slučaju neuspješne autentifikacije, poslani odgovor sadrži i razloge zbog kojih autentifikacija nije uspjela.

EAP-TLS paketi sadrže i jedno dodatno polje (u odnosu na osnovni EAP paket). To je 8-bitno polje *Flags*, kojim se postavljaju zastavice LMSRRRRR, ovisno o tome da li je u paketu uključeno polje *Length* (L), dodatni fragmenti (M), radi li se o EAP_Start paketu (S), te rezervirani bitovi (RRRRR).

3.3.8. LEAP

LEAP (*Lightweight EAP*) je tip Radius EAP autentifikacijskog protokola, koji se koristi za autentifikaciju bežičnog klijenta (prijenosnog računala ili računala s bežičnom karticom). Autentifikaciju ovom metodom čine poruke (Radius paketi) koje se razmjenjuju između pristupne točke (AP) i Radius poslužitelja. Protokol započinje porukom pristupne točke Radius poslužitelju s imenom korisnika koji je zatražio autentifikaciju. Poslužitelj odgovara porukom Radius Challenge (sadrži random MSCHAP Peer Challenge – PC). Treću poruku šalje pristupna točka poslužitelju s odgovorom na PC poslan prethodnom porukom (PR). Ako je autentifikacija uspjela Radius poslužitelj odgovara Radius Access- Accept porukom. Zatim AP šalje Radius Request poruku koja sadrži APC (Access Point Challenge). Poslužitelj na to odgovara porukom koja sadrži *session* ključ u sljedećem obliku Radius atributa: "leap:session-key=nnnn". Sve poruke su EAP oblika, polje *Type* ima postavljenu vrijednost 17, a polje *Type-Data* je podijeljeno na sljedeće dijelove: 1 oktet s brojem verzije LEAP protokola; 1 oktet postavljen na vrijednost 0x00; oktet s duljinom polja koje slijedi (*Binary-Data*); *m* okteta polja *Binary-Data*; *n* okteta s imenom korisnika koji je zatražio autentifikaciju. LEAP je još poznat i pod nazivom EAP-CISCO, jer se koristi u CISCO-vim usmjerivačima.

3.3.9. EAP-SIM

Metoda, koja za autentifikaciju i raspodjelu ključa koristi GSM SIM modul (*Subscriber Identity Module*). Takva autentifikacija se zasniva na algoritmima SIM modula, koji kao parametre primaju 128-bitni slučajni broj (RAND) i tajni ključ pohranjen na SIM modulu, a kao rezultat vraćaju 32-bitni odgovor (SRES) i 64-bitni ključ K_c, (koji se dalje upotrebljava za enkripciju, ali i za dobivanje drugih ključeva potrebnih za daljnje enkripcije).
Format EAP_SIM paketa izgleda kao i osnovni format EAP paketa (polja: *Code*, *Identifier*, *Length*, *Type*-postavljen na 18), nakon kojeg slijede polja *Subtype* i *Reserved*. Polje *Subtype* je određeno EAP_AKA specifikacijama, a ostatak paketa (Reserved) sadrži atribute definirane poljima *Attribute Type*, *Length* i *Value* (definiraju tip atributa, duljinu te podatke tog atributa).

Postoji 9 tipova EAP_SIM poruka, a oni su: EAP-Request/SIM/Start, EAP-Response/SIM/Start, EAP-Request/SIM/Challenge, EAP-Response/SIM/Challenge, EAP-Request/SIM/Re-authentication, EAP-Response/SIM/Re-authentication, EAP-Response/SIM/Client-Error, EAP-Request/SIM/Notification, EAP-Response/SIM/Notification. Poruke se međusobno razlikuju po atributima koje u sebi moraju/mogu sadržavati. Vrste atributa su: AT_VERSION_LIST, AT_SELECTED_VERSION, AT_NONCE_MT, AT_PERMANENT_ID_REQ, AT_ANY_ID_REQ, AT_FULLAUTH_ID_REQ, AT_IDENTITY, AT_RAND, AT_NEXT_PSEUDONYM, AT_NEXT_REAUTH_ID, AT_IV, AT_ENCR_DATA, AT_PADDING, AT_RESULT_IND, AT_MAC, AT_COUNTER, AT_COUNTER_TOO_SMALL, AT_NONCE_S, AT_NOTIFICATION, AT_CLIENT_ERROR_CODE.

3.3.10. EAP-AKA

Ova se metoda koristi u općim pokretnim telekomunikacijskim sustavima (*UMTS – Universal Mobile Telecommunication System*), a koristi AKA mehanizam (*Authentification and Key Agreement*). Taj se mehanizam zasniva na simetričnoj kriptografiji, a nalazi se na SIM modulima tj. USIM (*UMTS SIM*) modulima, RUIM (*Removable User Identity Module*, slični pametnim karticama). Autentifikacija se temelji na komunikaciji EAP poslužitelja (smještenom na stražnjem autentifikacijskom poslužitelju, koristeći AAA protokol-*Authentication*, *Autorization*, *Accounting*) i autentifikatora. Započinje ramjenom Identity Resquest/Response poruka (kao i kod drugih metoda), a nakon saznavanja identiteta pretplatnika (*subscriber-a*), stvara autentifikacijski vektor, koji služi za daljnju autentifikaciju. Zatim EAP poslužitelj započinje sa AKA algoritmom za daljnju autentifikaciju. Paketi koji se razmjenjuju ovom metodom u polju *Type* imaju zapisanu vrijednost 23.

3.3.11. EAP-PEAP

PEAP (*Protected Extensible Authentication Protocol*) je EAP metoda, s poljem *Type* paketa postavljenim na 25. Paket PEAP metode sličan je paketu EAP-TLS metode. Radi se o metodi koja je vrlo slična TLS metodi, a primjenjuje se kod bežičnih mreža. Ova metoda je podijeljena u dvije faze. U prvoj fazi autentifikacije se uspostavlja sigurnosni tunel između autentifikatora i poslužitelja, koristeći EAP-TLS za autentifikaciju poslužitelja. U drugoj se fazi autentificira klijent, koristeći bilo koju EAP metodu.

PEAP protokol zahtijeva certifikate i to samo za autentifikaciju poslužitelja. To ujedno čini i prvi korak autentifikacije. Zatim klijent treba uspostaviti vezu s autentifikatorom, a autentifikator treba uspostaviti sigurnosni kanal s poslužiteljem. Zatim slijede uobičajene Request/Response poruke između klijenta i autentifikatora te autentifikatora i poslužitelja, a nakon toga EAP-TLS protokol (razmjena poruka EAP-TLS metode). Završetkom EAP-TLS metode uspostavljen je sigurnosni kanal, te završava prva faza autentifikacije. Druga faza započinje razmjenom Identity poruka, a zatim slijedi

autentifikacija klijenta nekom od EAP metoda (MD5, CHAP, SIM itd.). Autentifikacija se odvija sigurnosnim kanalom uspostavljenim u prvoj fazi PEAP protokola. Slijedi računanje ključeva (klijent i poslužitelj računaju potrebne ključeve), a zatim te ključeve prima autentifikator, kao i rezultat autentifikacije. Time se završava autentifikacija: klijent i pristupna točka mogu sigurno razmjenjivati poruke.

3.3.12. EAP-MSCHAPv2

Metoda koja omogućuje obostranu autentifikaciju i metode računanja ključeva, kakve se koriste u MPPE kriptiranju (*engl. Microsoft Point to Point Encryption*). Skraćenica je nastala od naziva *Microsoft Challenge Handshake Authentication Protocol version 2*. Koristi se kao druga faza autentifikacije, nakon obavljenog protokola metode EAP-PEAP. Format paketa ove metode se malo razlikuje od osnovnog formata paketa, jer su dodana polja *OpCode*, *MS-CHAPv2-ID*, *MS-Length* i *Data*. *OpCode* polje definira tip EAP-MSCHAPv2 paketa, koji može biti: 1-Challenge, 2-Response, 3-Success, 4-Failure, 7-Change Password. *Type* polje paketa ove metode postavljeno je na vrijednost 26. Ono što razlikuje ove pakete je polje *Data*.

Tako za Challenge tip paketa ono sadrži podatke Challenge.

Za Response tip paketa polje *Data* sadrži polje *Respons*e, koji se dalje dijeli na 4 podpolja: *Peer-Challenge* (random broj), *Reserved* (mora biti 0), *NT-Response* (password), *Flags* (mora biti 0).

Success tip paketa ima dvije podvrste: Success Request i Success Response. Success Request u polju *Data* sadrži poruku (Message) formata: "S=<auth_string> M=<message>". Success Response paket od dodatnih polja, koje imaju paketi EAP-MSCHAPv2 metode, ne sadrži niti jedno, osim polja *OpCode* (s vrijednošću 3).

Failure tip paketa se isto dijeli na dva podtipa: Request i Response. Failure Request paket sadrži polje *Message* oblika: "E=eeeeeeeeee R=r C=cccccccccccccccccccccccccccccccc V=vvvvvvvvvv M=<msg>". Failure response paket, kao i Success Response paket, od dodatnih polja sadrži samo *OpCode*.

Paket tipa Change Password u polju *Data* ima 6 podpolja, a ona su: *Encrypted-Password*, *Encrypted-Hash*, *Peer-Challenge*, *Reserved*, *NT-Response*, *Flags*. Ovom metodom je podržana zaštita integriteta, ali nije podržana zaštita pouzdanosti.

3.3.13. EAP-PSK

Temelji se na simetričnoj kriptografskoj metodi AES-128, što metodu EAP-PSK (*Pre Shared Key*) čini jednostavnom za implementirati. PSK predstavlja 128-bitni ključ, kojeg znaju samo EAP poslužitelj i EAP partner (peer), te služi za računanje daljnjih ključeva (*Authentification Key-AK* i *Key Derivation Key-KDK*). Iz KDK se dalje računaju drugi potrebni ključevi. Uz PSK, metoda koristi i identifikatore poslužitelja i partnera (ID_S i ID_P). Autentifikacija ovom metodom se zasniva na AKEP2 protokolu, a čine ju četiri poruke. U slučaju uspješne autentifikacije EAP-PSK metoda stvara zaštićeni kanal za komunikaciju obiju strana (pri tome koristi EAX metodu). Zaštićeni kanal omogućuje prijenos poruka s kriptiranim tekstom (u daljnjem tekstu - PCHANNEL). Prvu EAP-PSK poruku šalje poslužitelj partneru (server -> partner). Ta poruka ima (uz uobičajena polja EAP paketa) još dodatna dva polja: *RAND_S* (random broj) i *ID_S* (*server's NAI*). Drugu poruku šalje partner poslužitelju, a njena dodatna polja su: *RAND_S* (isti kao u 1. poruci), *RAND_P* (random broj), *MAC_P* (MAC adresa partnera), *ID_P* (peer's NAI). Treću poruku šalje opet poslužitelj partneru, a čine ju dodatna polja: *RAND_S* (isti kao u 1. poruci), *MAC_S* (MAC adresa poslužitelja), *PCHANNEL*. *PCHANNEL* polje se sastoji od podpolja: *Nonce N*, *Tag*, *zastavica R*, *zastavica E*,

Reserved. Četvrtu poruku šalje partner poslužitelju, s poljima RAND_S (isti kao i u 1. poruci) te PCHANNEL.

3.3.14. EAP-PAX

Metoda EAP-PAX (*Password Authenticated Exchange*)za autentifikaciju koristi dijeljeni ključ, a može se podijeliti na dva "podprotokola": PAX_STD i PAX_SEC. PAX_STD obavlja obostranu autentifikaciju koristeći dijeljeni tj. zajednički ključ, a PAX_SEC nadopunjuje prije spomenuti protokol, pružajući zaštitu identiteta koristeći javni ključ poslužitelja. EAP-PAX ima svojstva, koja zadovoljavaju EAP zahtjeve za sigurnošću bežičnih mreža, kao npr.: obostrana autentifikacija; otpornost napadima posrednikom (*engl. man-in-the-middle attack*); zaštita identiteta; autentificirana zaštita podataka itd. Paketi ove metode se prepoznaju po polju *Type* postavljenim na 46.

3.3.15. EAP-FAST

EAP-FAST autentificira i klijenta i autentifikacijskog poslužitelja, koristeći dijeljenu tajnu, poznatu pod nazivom PAC (*Protected Access Credential*). Metoda se može podijeliti na dva dijela: prvi - pripremni i drugi – autentifikacijski. U prvom dijelu se PAC šalje klijentu i poslužitelju i taj se dio obavlja samo jednom (ručno ili automatski). Automatsko slanje stvara tunel, kojim putuju kriptirane poruke i na taj način se osigurava autentifikacija klijenta i sigurna dostava PAC-a klijentu. Taj je mehanizam manje siguran od ručnog slanja, ali je sigurniji od LEAP mehanizma (prepoznaje i onemogućuje napade posrednikom (*engl. man-in-the-middle-attack*). Nakon dobivanja informacija sadržanih u PAC dijeljenoj tajni, slijedi autentifikacija klijenta i poslužitelja, stvaranje kriptiranog tunela, obnavljanje dijeljene tajne (refresh PAC), autentifikacija klijenta koristeći neku EAP metodu te objava autorizacije klijentu (opcionalno).

4. Programsko ostvarenje *WPA_supplicant-a*

U ovom će poglavlju biti prikazan način programskog ostvarenja *WPA_supplicant-a.* Programsko ostvarenje obuhvaća razradu pojedinih modula koji čine *WPA_supplicant*, a prikazani su na slici 2.2. Ti moduli su: kontrolno sučelje, automati stanja, sučelje prema mrežnim sučeljima te l2_layer sučelje.

4.1. Kontrolno sučelje

Kontrolno sučelje služi vanjskim programima za upravljanje i primanje obavijesti o statusu operacija, koje izvodi *WPA_supplicant deamon*. Implementirano je u `wpa_ctrl.c` datoteci, a tu datoteku za svoj rad koriste `wpa_cli.c` te `wpa_gui.c`. Kako bi se ostvarila komunikacija programa s kontrolnim sučeljem potrebno je ostvariti vezu programa sa sučeljem. To se ostvaruje posebnim funkcijama, o čemu će biti više riječi u daljnjem tekstu. *WPA_supplicant* koristi kontrolno sučelje za dvije vrste komunikacije: *naredbama* i *unsolicated messages* ("ne zahtjevanim") porukama. Naredbe čine parovi poruka: zahtijev (vanjskog) programa i odgovor *WPA_supplicant-a. Unsolicated* poruke su poslane kontrolnom sučelju od strane *WPA_supplicant-a*, bez da je program poslao zahtijev za tim porukama. Kako bi program primio te poruke, treba biti ostvarena veza s kontrolnim sučeljem. Kad se kontrolno sučelje koristi za oba načina komunikacije, najbolje bi bilo ostvariti dvije veze s kontrolnim sučeljem. Tada bi se jedna veza koristila za komunikaciju naredbama, a druga za komunikaciju *unsolicated* porukama. Time bi se ujedno i spriječila mogućnost da se *unsolicated* poruka ubaci između zahtijeva programa i odgovora *WPA_supplicant-a*, što je moguće ukoliko se koristi samo jedna veza za komunikaciju. *wpa_cli* predstavlja primjer korištenja samo jedne veze za oba načina komunikacije, a *wpa_gui* predstavlja primjer korištenja dviju veza za komunikaciju. Po završetku komunikacije s kontrolnim sučeljem potrebno je zatvoriti otvorenu vezu.

4.1.1. Funkcije kontrolnog sučelja

Kao što je već spomenuto, postoje funkcije kojima se ostvaruje veza programa s kontrolnim sučeljem, ali i funkcije kojima se šalju naredbe, *unsolicated* poruke, poruke za zatvaranje veze s kontrolnim sučeljem:

Tablica 3: Funkcije kontrolnog sučelja

`wpa_ctrl_open()`	otvaranje veze s kontrolnim sučeljem
`wpa_ctrl_request()`	slanje naredbi
`wpa_ctrl_attach()`	stvaranje veze s kontrolnim sučeljem, kako bi se mogle primati *unsolicated* poruke
`wpa_ctrl_close()/wpa_ctrl_detach()`	zatvaranje veze s kontrolnim sučeljem

Wpa_cli i *wpa_gui* koriste gore navedene funkcije za komunikaciju s *WPA_supplicant-om*.

4.1.2. Naredbe kontrolnog sučelja

Kao što je u prethodnom odjeljku spomenuto u C datoteci `wpa_ctrl.c` nalaze se funkcije kontrolnog sučelja. Funkcija `wpa_ctrl_request()` služi za slanje naredbi kontrolnom sučelju,a neke naredbe kontrolnog sučelja i njihove karakteristike su:

STATUS: naredba kojom se dobiva informacija o trenutnom EAP/EAPOL/WPA statusu. Ispis nakon izvršene ove naredbe sadrži tekst s postavljenim vrijednostima za pojedine varijable: *bssid*, *ssid*, *IP address*, *Supplicant Status*, *WPA state*, *EAP state* itd. Slična ovoj je i naredba STATUS – VERBOSE (sadrži ispis više varijabli i njihovih statusa).

BSSID <network id> <BSSID>: naredba za stavljanje zadanog BSSID-a za određenu mrežu (*network id*).

PREAUTH <BSSID>: naredba kojom se omogućava preautentifikacija sa predanim *BSSID-om*.

LOGON/LOGOFF: *logon/logoff* EAPOL automata stanja.

RECONFIGURE: naredba kojom se *WPA_supplicant-u* zadaje da (ponovno) učitava svoj konfiguracijski file.

ATTACH/DETACH: naredbe kojima se stvara/ukida veza za slanje i primanje *unsolicated* poruka.

REASSOCIATE: naredba za ponovnim spajanjem na AP.

TERMINATE: završetak procesa *WPA_supplicant-a*.

DISCONNECT: prekid veze (s AP-om).

SCAN: naredba za traženjem novih *BSS-ova.*

AP SCAN <ap_scan value>: naredba kojom se mijenja vrijednost varijable *ap_scan*. Ako je ta vrijednost postavljena na 0, *WPA_supplicant* ne zahtjeva traženje AP-ova. Ako je ta vrijednost jednaka 1, zahtjeva se traženje AP-ova i rezultati dobiveni tim traženjem se koriste pri odabiru odgovarajućeg AP-a za spajanje. Posljednja vrijednost koju AP može poprimiti je 2, a tada se zahtijeva od upravljačkih programa (*driver-a*) odabiranje pristupne točke na koju će se spajati.

SELECT/ENABLE/DISABLE/ADD/REMOVE/SET NETWORK <network id>: naredbe kojima se odabire/omogućava/onemogućava/dodaje/skida/postavlja neka mreža, koja se može izabrati iz liste mreža dobivenih naredbom LIST_NETWORKS ili u nju dodati (naredba ADD).

Postoje i naredbe koje omogućavaju interaktivni rad *WPA_supplicant-a*. To znači da je tim naredbama omogućeno *WPA_supplicant-u* da traži neke dodatne zahtjeve od vanjskog programa, kao što su: EAP zaporku, korisničko ime, novu zaporku, pin i slično. Sve takve naredbe upućene od *WPA_supplicant-a* programu započinju prefiksom (WPA_CTRL)_REQ. Odgovor od strane programa započinje prefiksom (WPA_CTRL)_RSP. Takve naredbe su sljedećeg oblika:

```
# CTRL-REQ-<field name>-<network id>-<human readable text>
# CTRL-RSP-<field name>-<network id>-<value>
```

Primjer naredbe poslane od strane *WPA_supplicant-a* GUI-u:

```
# CTRL-REQ-PASSWORD-1-Password needed for SSID test-network
```

Odgovor GUI-a *WPA_supplicant-u* je sljedeći:

```
CTRL-RSP-PASSWORD-1-secret
```

4.2. Automati stanja

EAP autentifikacija se ostvaruje nekom od EAP metoda, a njih čine poruke koje se šalju između EAP partnera i autentifikatora. Odabir metode autentifikacije prepušta se *EAP switch-u.* EAP *switch* se

nalazi i na strani partnera i na strani autentifikatora. Kako i partner i autentifikator podržavaju više EAP metoda za autentifikaciju, EAP *switch* je potreban za "dogovaranje" partnera i autentifikatora koju će koristiti. Automati stanja se temelje na EAP *switch* modelu, što znači da automate čine stanja i varijable za "komunikaciju" EAP *switch-a* i EAP metoda.

Automati stanja EAP protokola se mogu podijeliti na 4 vrste: Automat stanja partnera (*EAP peer state machine*), Automat stanja samostalnog autentifikatora (*EAP stand-alone authenticator state machine*), automat stanja (stražnjeg) autentifikatora (*EAP backend authenticator state machine*) (za AAA poslužitelje) te Automat stanja potpunog autentifikatora (*EAP full authenticator*). Automati stanja *EAP peer_sm* i *EAP state-alone authenticator_sm*, prikazuju implementaciju EAP protokola, kakva je objašnjena u RFC[3748]. Druga dva automata stanja opisuju EAP-AAA (A*uthentication, Autorization, Accounting*) protokol, prikazanog i objašnjenog u RFC[3579].

4.2.1. Automat stanja partnera

Razmjena poruka između nižeg sloja i automata stanja partnera (*EAP peer state machine*) temelji se na stavljanju poruke u paket *eapReqData* te postavljanjem varijable *eapReq*=TRUE. Nakon toga *EAP peer_sm* procesira poruku i postavlja varijablu *eapResp* ili *eapNoResp*. Kad autentifikacija završi *EAP peer_sm* šalje *eapSuccess* ili *eapFailure*, kako bi obavijestio niži sloj o uspješnosti autentifikacije. Ova se razmjena paketa (niži sloj – automat i obratno) odvija kroz sučelje niži sloj – automat. Postoji i drugo sučelje: automat – EAP metoda. Tim sučeljem EAP metoda prima od automata paket i odlučuje da li ga prihavaća ili ne. Svaka EAP metoda ima svoj automat stanja i mijenja stanja ovisno o ulazima (primljenim paketima, postavljenim varijablama...). Stanja kroz koja *EAP peer_sm* može prolaziti su *DISABLED*, *INITIALIZED*, *IDLE*, *RECEIVED*, *DISCARD*, *RETRANSMIT*, *SUCCESS*, *FAILURE*, *GET_METHOD*, *METHOD*, *SEND_RESPONSE*, *IDENTITY*, *NOTIFICATION*. Osim stanja kroz koja može autentifikator prolaziti, automat stanja karakteriziraju i varijable, koje su podijeljene u dvije grupe: varijable za "komunikaciju" nižeg sloja i autentifikatora, te varijable za "obrnutu komunikaciju" (od autentifikatora k nižem sloju).

4.2.2. Automat stanja samostalnog autentifikatora

Kao i kod *EAP peer sm-a* postoje dva sučelja. Prvo je između nižeg sloja i automata stanja. Niži sloj sprema poruku u paket *eapRespData* te postavlja *eapResp*=TRUE, te na taj način prosljeđuje poruke *EAP stand-alone authenticator-u*. Kada autentifikator završi procesiranje poruka šalje jedan od sljedećih signala: *eapReq, eapNoReq, eapSuccess* ili *eapFail*, ovisno o uspješnosti autentifikacije. Drugim sučeljem odvija se interakcija između automata i EAP metode. EAP metoda prima paket, te ukoliko ga odluči zadržati koristi ga u svom protokolu autentifikacije, a u protivnom ga odbacuje s odgovarajućom porukom. Stanja kroz koja može prolaziti *EAP stand-alone authenticator* su: *DISABLED*, *INITIALIZE*, *IDLE*, *RECEIVED*, *INTEGRITY_CHECK*, *METHOD_RESPONSE*, *METHOD_REQUEST*, *PROPOSE_METHOD*, *SELECT_ACTION*, *SEND_REQUEST*, *DISCARD*, *NAK*, *RETRANSMIT*, *SUCCESS*, *FAILURE*, *TIMEOUT_FAILURE*. Kao i kod automata stanja partnera (*EAP peer state*) postoje dvije vrste varijabli: varijable za komunikaciju niži sloj – autentifikator, te suprotnu komunikaciju, autentifikator – niži sloj.

4.2.3. Automat stanja autentifikatora

Ovaj automat stanja (*EAP backend authenticator state machine*) čine dva dijela. U prvom dijelu se poruke tj. paketi prenose preko autentifikatora, a drugi dio čini EAP metoda tj. autentifikacija. Postoje dva sučelja: sučelje između nižeg sloja i automata te sučelje između automata i EAP metoda. Kroz prvo sučelje putuju poruke nižeg sloja u paketima *aaaEapRespData*, a niži sloj postavlja varijablu

aaaEapResp. Tim sučeljem putuju i paketi *aaaEapReqData* koje šalje automat nakon završenog procesiranja poruke te postavlja jednu od varijabli: *aaaEapReq*, *aaaEapNoReq*, *aaaSuccess*, ili *aaaFail*. Drugo sučelje slično je drugom sučelju kod *EAP stand-alone authenticator state machine* (poglavlje 4.2.2). Moguća stanja ovog automata su ista kao i kod *EAP stand-alone authenticator state machine*, s dodatkom još jednog stanja: *PICK_UP_METHOD*.

4.2.4. Automat stanja potpunog autentifikatora

Posebnost ovog automata stanja (*EAP full authenticator*) je što pruža mogućnost *pass-through* načina rada i ima sučelja za komunikaciju s nekoliko nižih slojeva. Jedno od njih je isto onom opisanom kod automata stanja *EAP stand-alone authenticator-a*, sučelje prema prvom nižem sloju (*EAP transport layer*). Drugo sučelje je prema drugom nižem sloju, AAA (*Authentication*, *Autorization*, *Accounting*) sloju. Stanja ovog automata su: *INITIALIZE_PASSTHROUGH, IDLE2, IDLE, RECEIVED2, AAA_REQUEST, AAA_IDLE, AAA_RESPONSE, SEND_REQUEST2, DISCARD2, RETRANSMIT2, SUCCESS2, FAILURE2, TIMEOUT_FAILURE2*.

4.3. Sučelje prema mrežnim sučeljima

Sučelje prema mrežnim sučeljima (*engl. Driver Wrapper Implementation*) predstavlja skup dodatnih ".c" datoteka (pisanih u jeziku C) s generičkim kodom. Spomenuti generički kod je implementacija svih funkcija koje upravljački programi (*drivers*) koriste, a koje (možda) nisu implementirane među njihovim osnovnim funkcijama. Kako je *WPA_supplicant* zamišljen kao *hardware-ski* i *driver-ski* neovisan te neovisan o operacijskom sustavu, ove datoteke služe upravo za dijelove, koji su ovisni o implementaciji upravljačkih programa. Takvi dijelovi se zamijenjuju generičkim kodom implemetiranim u datotekama. Sve datoteke tj. funkcije *driver-a* implementirane u tim datotekama moraju definirati strukturu `wpa_driver_ops`, koja se koristi u `wpa_supplicant.c` prilikom pozivanja funkcija upravljačkih programa.

```
const struct wpa_driver_ops wpa_driver_hostap_ops = {
.name = "hostap",
desc = "Host AP driver (Intersil Prism2/2.5/3)",
.get_bssid = wpa_driver_hostap_get_bssid,
.get_ssid = wpa_driver_hostap_get_ssid,
.set_wpa = wpa_driver_hostap_set_wpa,
.set_key = wpa_driver_hostap_set_key,
.set_countermeasures = wpa_driver_hostap_set_countermeasures,
.set_drop_unencrypted = wpa_driver_hostap_set_drop_unencrypted,
.scan = wpa_driver_hostap_scan,
.get_scan_results = wpa_driver_hostap_get_scan_results,
.deauthenticate = wpa_driver_hostap_deauthenticate,
.disassociate = wpa_driver_hostap_disassociate,
.associate = wpa_driver_hostap_associate,
.set_auth_alg = wpa_driver_hostap_set_auth_alg,
.init = wpa_driver_hostap_init,
.deinit = wpa_driver_hostap_deinit,
.set_operstate = wpa_driver_hostap_set_operstate,
;
```

Neki primjeri datoteka s generičkim kodom, koje su uključene u sučelje ove *WPA_supplicant* implementacije:

`driver_wext.c` – datoteka koja sadrži implementirane funkcije *Generic Linux Wireless Extensions*.

`driver_ndis.c` – datoteka, koja sadrži cjelokupno sučelje za Windows NDIS sučelje *driver-a*.

`driver_hostap.c` – uključuje cjelokupnu implementaciju za Linux upravljačke programe, koji koriste *WPA_supplicant*.

Postoje neki zahtjevi za implementaciju upravljačkih programa (*driver*):

1. Zahtjev da se kao enkripcijski algoritmi koriste TKIP i CCMP. TKIP je skraćenica za *Temporal Key Integrity Protocol*, enkripcijski algoritam, sličan WEP-u, zapravo dizajniran kako bi zamijenio WEP. Za kriptiranje koristi RC4, a podržava provjeru integriteta i mehanizam *re-keying-a.* CCMP je skraćenica za *Counter Mode with Chiper Block Chaining Message Authentication Code Protocol*, enkripcijski algoritam, koji je također nastao kako bi zamijenio WEP, ali za razliku od TKIP-a koristi AES (*engl. Advanced Encryption standard).*
2. Zahtjev za generiranjem WPA IE (*Information Element*), koji bi se trebao generirati na temelju kriptografskih svojstava i *key management-a.* Ukoliko driver omogućava navedeni zahtjev, *WPA_supplicant* je konfiguriran s varijablom *ap_scan* postavljenom na vrijednost 2.
3. Zahtjev za podrškom odabira pristupnih točki (*engl. Access Point -AP*). Odabir bi se trebao provoditi na temelju informacija iz *Beacon* okvira, koje pristupna točka odašilje tj. iz *Probe/Response* okvira. To znači da driver treba omogućavati odabir AP-a. U konfiguracijskoj datoteci to je označeno varijablom *ap_scan* postavljenom na vrijednost 1. Također, *driver* treba omogućavati povezivanje na odabrane *BSS-ove*
4. Zahtjev za *driver events-ima* podrazumijeva primanje povratnih informacija (*callbacks*), nakon što se dogodi pojedini događaj (*event*) vezan uz *driver.* To znači kako bi *WPA_supplicant* trebao dobiti informaciju o npr. uspostavljenoj asocijaciji/deasocijaciji *driver-a* i slično, događajima definiranim u funkciji `wpa_supplicant_svents()`.

Odabir pristupne točke (AP), na koju će se spajati, ovisi o postavljanju vrijednosti varijable *ap_scan.* Moguće vrijednosti te postupci pri spajanju, su sljedeći:

- *ap_scan*=1:
  - *WPA_supplicant* zahtjeva traženje mogućih AP-ova s SIOCSIWSCAN
  - driver daje rezultate tj. povratnu informaciju o pronađenim AP-ovima s SIOCGIWSCAN
  - *WPA_supplicant* čita rezultate s SIOCGIWSCAN
  - *WPA_supplicant* na temelju rezultata određuje koji AP će se koristiti
  - *WPA_supplicant* konfigurira *driver* kako bi se spojio na odabrani BSS (SIOCSIWMODE, SIOCSIWGENIE, SIOCSIWAUTH, SIOCSIWFREQ, SIOCSIWESSID, SIOCSIWAP)
- *ap_scan*=2:
  - *WPA_supplicant* konfigurira *driver* da se spoji s SSID (SIOCSIWMODE, SIOCSIWGENIE, SIOCSIWAUTH, SIOCSIWESSID)

4.4. L2_layer sučelje

L2_layer sučelje služi *WPA_supplicant - u* za slanje i primanje paketa s 2. sloja (podatkovnog sloja). Implementacija tog sučelja ostvarena je u nekoliko ".c" datoteka. Tako postoji implementacija l2 sučelja za Linux - `l2_packet_layer.c`, zatim implementacija za FreeBSD – `l2_packet_freebsd.c`, te implementacija koja koristi libpcap/libdnet biblioteke – `l2_packet_pcap.c`.

Implementaciju `l2_packet_layer.c` čini struktura *l2_packet_data*, te sljedeće funkcije:

- `int l2_packet_get_own_addr (struct l2_packet_data *l2, u8 *addr)`

- `l2_packet send (struct l2_packet_data *l2, const u8 *dst_addr, u16 proto, const u8 *buf, size_t len)`
- `l2_packet_data * l2_packet_init (const char *ifname, const u8 *own_addr, unsigned short protocol, void(*rx_callback)(void *ctx, const u8 *src_addr, const u8 *buf, size_t len), void *rx_callback_ctx, int l2_hdr)`
- `void l2_packet_deinit (struct l2_packet_data *l2)`
- `int l2_packet_get_ip_addr (struct l2_packet_data *l2, char *buf, size_t len)`
- `void l2_packet_notify_auth_start (struct l2_packet_data *l2)`

5. Korištenje programa *WPA_supplicant*

U ovom pogavlju bit će prikazana primjena *WPA-supplicant-a* za dvije različite metode autentifikacije. Prvi primjer pokazuje bežično spajanje prijenosnog računala na pristupnu točku (AP), koja kao metodu zaštite koristi WPA-PSK (OSLwlan). Drugi primjer isto prikazuje bežično spajanje prijenosnog računala na pristupnu točku, no ovaj put pristupna točka, kao metodu zaštite koristi EAP-MSCHAPv2 (FERwlan). Programska komponenta preuzeta je sa sljedeće adrese: http://hostap.epitest.fi/wpa_supplicant/. Za pokretanje tog programa potrebno je bilo napisati odgovarajuće konfiguracijske datoteke. To ujedno čini praktični dio ovog rada: pokretanje *WPA_supplicant-a*. Sadržaj konfiguracijskih datoteka prikazan je i objašnjen u nastavku teksta.

5.1. Pristup mreži zaštićenoj WPA-PSK metodom

Pristupna točka u OSL-u (*Open source laboratory*) kao metodu zaštite koristi WPA-PSK. Kako bi se npr. prijenosno računalo spojilo na mrežu preko te pristupne točke, potrebno je bilo na računalu pokrenuti *WPA*_supplicant. Konfiguracijska datoteka sadrži sve parametre potrebne za metodu WPA-PSK. Konfiguracijska datoteka *wpa_supplicant.conf* izgleda ovako:

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
eapol_version=1
ap_scan=1
fast_reauth=1
network={
       ssid="osl-private-wlan"
       scan_ssid=1
       psk="šifra_za_pristup_pristupnoj_točki"
       proto=WPA
       key_mgmt=WPA-PSK
       pairwise=TKIP
       group=TKIP
}
```

Prva linija konfiguracijske datoteke navodi ime direktorija u kojem je *WPA_supplicant* stvorio *socket*, kako bi mogao primati zahtjeve vanjskih programa (npr. GUI) za statusnom informacijom i konfiguracijom. Tim *socket-om* se služi i `wpa_cli` prilikom spajanja s *WPA_supplicant-om.*

Zatim se definira da samo *root* ima ovlasti koristiti kontrolno sučelje. To je definirano linijom *ctrl_interface_group=0*, gdje 0 označava gid *root-a*. Ukoliko se želi da i ostali korisnici (ne samo *root*) imaju pristup kontrolnom sučelju (a time imaju i mogućnost mijenjati konfiguraciju mreže), umjesto 0 se napiše ime grupe korisnika kojima se to omogućuje.

WPA_supplicant je napravljen s podrškom EAPOL verzije 2, ali mnoge pristupne točke ne podržavaju tu verziju te se s linijom *eapol_version=1* omogućuje korištenje *WPA_supplicant-a* i s takvim pristupnim točkama.

Sljedećom linijom se postavlja vrijednost *ap_scan* na 1, što znači da će *WPA_supplican*t započeti traženje mogućih pristupnih točaka i odabrati najbolju za spajanje na nju.

Također i vrijednost *fast_reauth* se postavlja na 1, čime se omogućuje brza ponovna autentifikacija (koja je omogućena za sve EAP metode, no ukoliko se postavi na 0 postaje onemogućena).

Slijedi *network* dio (mrežni dio) konfiguracijske datoteke. On je isti za sve pristupne točke koje imaju isti *SSID*. Najprije se definira SSID, tj. ime mreže, a zatim se varijabli *scan_ssid* postavlja vrijednost na 1. Time se definira traženje pristupnih točaka koje ne prihvaćaju *broadcast* SSID tj. traženje pristupnih točaka se obavlja sa SSID specifičnim *probe-request* okvirima. U sljedećoj liniji navodi se šifra *psk* (*pre-shared key*) tj. WPA ključ. Kako se radi o WPA-PSK metodi zaštite pristupa na mrežu potreban je ključ. Varijablom *proto* navodi se lista prihvaćenih/mogućih protokola (WPA), a

varijablom *key_mgmt* (*key_management*) navodi se lista mogućih autentifikacijskih protokola za razmjenu ključeva (WPA-PSK). Zatim se definira lista enkripcijskih algoritama (*unicast* i *broadcast cipher*), postavljeno na TKIP (*engl. Temporal Key Integrity Protocol*).

5.2. Pristup mreži zaštićenoj EAP-MSCHAPv2 metodom

Pristupne točke za pristup FERwlanu kao metodu zaštite koriste EAP-MSCHAPv2. Konfiguracijska datoteka *wpa_supplicantFER.conf* za bežični pristup tim pristupnim točkama izgleda ovako:

```
ctrl_interface=/var/run/wpa_supplicant
ctrl_interface_group=0
eapol_version=1
ap_scan=2
fast_reauth=1
network={
       ssid="FERwlan"
       key_mgmt=IEEE8021X
       eap=PEAP
       phase2="auth=MSCHAPv2"
       identity="loginNaFER-u"
       password="šifra"
}
```

Prvih nekoliko linija konfiguracijske datoteke isti je kao i u konfiguracijskoj datoteci za pristup mreži zaštićenoj WPA-PSK metodom. Naziv direktorija u kojem je stvoren *socket* za komunikaciju s vanjskim programima je */var/run/wpa_supplicant*, samo *root* ima ovlasti koristiti kontrolno sučelje, *eapol_version* je postavljen na 1, a omogućena je i ponovna autentifikacija. Razlika je jedino u varijabli *ap_scan*, koja je za pristup ovoj mreži postavljena na 2. Time se definira da *driver* započinje traženje mogućih pristupnih točaka i zatim odabire najpovoljniju pristupnu točku za spajanje na nju.

U *network* (mrežnom) dijelu konfiguracijske datoteke, najprije se definira ime mreže (SSID), a potom i varijabla *key_mgmt. Key_mgmt* je postavljen na vrijednost *IEEE8021X*, čime se definira EAP autentifikacija i dinamičko generiranje ključeva. Varijabla *eap* navodi listu mogućih EAP metoda i postavlja se na vrijednost PEAP. Zatim se kao metoda 2. faze autentifikacije navodi MSCHAPv2. Ukoliko je mreža zaštićena EAP-MSCHAPv2 metodom, autentifikacija se mora odvijati u dvije faze: u prvoj se radi o EAP-PEAP metodi, a s tim je povezana (i dolazi u drugoj fazi) metoda EAP-MSCHAPv2. Slijede dvije varijable koje određuju korisnika koji se želi spojiti na mrežu: *identity*, kao identitet korisnika i *password*, kao šifra korisnika.

5.3. Pokretanje *WPA_supplicant-a*

Nakon što su napisane konfiguracijske datoteke, *WPA_supplicant* se pokreće iz komandne linije (iz direktorija u kojem se nalazi konfiguracijska datoteka) naredbom:

```
# ./wpa_supplicant/wpa_supplicant -i eth1 -c ./wpa_supplicantFER.conf -D wext
```

Tako se npr. prijenosno računalo spaja na pristupnu točku za koju je napisana konfiguracijska datoteka *WPA_supplicantFER.conf* (u ovom primjeru: pristupna točka na FER-u, metoda zaštite: EAP-MSCHAPv2). Za spajanje na neku drugu pristupnu točku tj. točku s nekim drugim SSID-om, potrebno je napisati odgovarajuću konfiguracijsku datoteku i pokrenuti gore navedenu naredbu s imenom te datoteke. Iz druge konzole pokreće se naredba:

```
# dhclient eth1
#
```

Ta naredba traži od DHCP poslužitelja dodjelu IP adrese klijentu, koji je zatražio spajanje na pristupnu točku (u ovom primjeru: prijenosnom računalu). Prilikom procesa spajanja, u konzoli se ispisuje status istoga:

```
Trying to associate with SSID 'FERwlan'
Associated with 00:12:01:bf:86:a0
CTRL-EVENT-EAP-STARTED EAP authentication started
CTRL-EVENT-EAP-METHOD EAP vendor 0 method 25 (PEAP) selected
EAP-MSCHAPV2: Authentication succeeded
EAP-TLV: TLV Result - Success - EAP-TLV/Phase2 Completed
CTRL-EVENT-EAP-SUCCESS EAP authentication completed successfully
CTRL-EVENT-CONNECTED - Connection to 00:12:01:bf:86:a0 completed (auth)
```

Iz ispisa se vidi koji je SSID pristupne točke na koju se želi spojiti (FERwlan) i koja je MAC adresa (00:12:01:bf:86:a0). Slijedi popis daljnjih radnji potrebnih za autentifikaciju (*eventi*): poruka o započetoj EAP autentifikaciji, te o broju autentifikacijske (EAP) metode koja je odabrana (25 – PEAP). Ukoliko je autentifikacija uspjela slijedi poruka *Authentification succeeded*, te započinju daljnje faze autentifikacije (ukoliko su potrebne, ovisno o metodi zaštite pristupu mreži). Na kraju svih faza autentifikacije slijedi poruka o uspješno završenoj autentifikaciji (Success poruka).

6. Zaključak

U prethodnim poglavljima opisan je *WPA_supplicant*. Rad te programske komponente temelji se na proširivom autentifikacijskom protokolu - EAP protokolu. Spomenuti protokol ima nekoliko vrsta metoda, koje se koriste za autentifikaciju klijenta. Svaka od tih metoda je posebna: po formatu paketa koji se u procesu autentifikacije razmjenjuju, ali i po svojem načinu provođenja autentifikacije. Upravo raznolikost EAP metoda čini jednu od prednosti ovog protokola. Također, ne zahtijeva se od pristupnih točaka ili usmjernika da podržavaju sve EAP metode; oni mogu poslužiti kao *pass-through* do autentifikacijskog poslužitelja, koji podržava sve EAP metode i može obaviti proces autentifikacije.

Srž komponente *WPA_supplicant* čine automati stanja. Ovisno o tijeku autentifikacije automati prolaze kroz svoja stanja, pritom koristeći EAP metode. Uz automate stanja, implementaciju *WPA_supplicant-a* čine i moduli, koji čine sučelje prema mrežnim sučeljima (*driver wrapper implementation*) te l2_layer sučelje. Moduli sadrže generički kod, koji implementira sve funkcije koje *driveri* koriste. L2_layer sučelje je sučelje prema 2. sloju (podatkovnom sloju) i služi za razmjenu paketa s 2. slojem.

Primjena *WPA_supplicant-a* isprobana je na bežičnom spajanju prijenosnog računala, s Linux operacijskim sustavom, na dvije različite pristupne točke tj. na pristupne točke s različitim metodama zaštite. Pri tome je bilo potrebno napisati odgovarajuće konfiguracijske datoteke, za svaku metodu tj. pristupnu točku. To podrazumijeva odabir odgovarajućih varijabli i dodjele vrijednosti koje odgovaraju metodi zaštite dotične pristupne točke. Nakon što su konfiguracijske datoteke ispravno napisane, uspješno su bile i uspostavljene bežične veze prema pristupnim točkama.

Daljnji rad na ovoj temi obuhvaća primjenu *WPA_supplicant-a* u *road warrior* načinu rada, kao što je objašnjeno u uvodnom poglavlju. Sada, kada je poznata implementacija *WPA_supplicant-a*, te način rada tj. autentifikacije nekom od EAP metoda (koju *WPA_supplicant* koristi), cilj je iskoristiti *WPA_supplicant* u IKEv2 protokolu. Ideja je koristiti EAP protokol za autentifikaciju *road warrior-a*, ali kako već postoji gotov program – *deamon* (*WPA_supplicant*), koji koristi taj protokol pri autentifikaciji, povezat će se IKEv2 protokol i *WPA_supplicant*.

7. Literatura

1. L. Budin, M. Golub, *Operacijski sustavi 2*, predavanja iz predmeta Operacijski sustavi 2, Zagreb 2005.
2. V. Glavinić, *Mreže računala*, (radni materijal za predavanja iz predmeta Mreže računala), Zagreb 2004.
3. B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, H. Levkowetz Ed., Extensible Authentication Protocol (EAP), 2004., URL: http://www.faqs.org/rfc/rfc3748.txt
4. B. Aboba, D. Simon, PPP EAP TLS Authentication Protocol, 1999., URL: http://www.faqs.org/rfc/rfc2716.txt
5. J. Arkko, H. Haverinen, Extensible Authentication Protocol Method for 3rd Generation, Authentication and Key Agreement (EAP-AKA), 2006., URL: http://www.faqs.org/rfc/rfc4187.txt
6. M. Bauer, Paranoid Penguin - Securing WLANs with WPA and FreeRADIUS, Part I, 2005., URL: http://www.linuxjournal.com/article/8017
7. M. Bauer, Paranoid Penguin - Securing Your WLAN with WPA and FreeRADIUS, Part II, 2005., URL: http://www.linuxjournal.com/article/8095
8. F. Bersani, H. Tschofenig, The EAP-PSK Protocol: a Pre-Shared Key EAP Method, draft-bersani-eap-psk-07, 2005., URL: http://www.tschofenig.com/drafts/draft-bersani-eap-psk-07.html
9. T. Clancy, W. Arbaugh, EAP Password Authenticated Exchange, draft-clancy-eap-pax-07, 2006., URL: http://www.ietf.cnri.reston.va.us/internet-drafts/draft-clancy-eap-pax-07.txt
10. N. Haller, C. Metz, P. Nesser, M. Straw, A one time password (OTP), 1998., URL: http://www.faqs.org/rfc/rfc2289.txt
11. H. Haverinen, Ed., J. Salowey, Ed., Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM), 2006., URL: http://www.faqs.org/rfc/rfc4186.txt
12. Jouni Malinen, Linux WPA/WPA2/IEEE 802.1X Supplicant, URL: http://hostap.epitest.fi/wpa_supplicant/
13. J. Vollbrecht, P. Eronen, N. Petroni, Y. Ohba, State Machines for Extensible Authentication Protocol (EAP), Peer and Authenticator, 2005., URL: http://www.faqs.org/rfc/rfc4137.txt
14. G. Zorn, Microsoft PPP CHAP Extensions, Version 2, 2000., URL: http://www.faqs.org/rfc/rfc2759.txt

Dodatak A: Popis kratica

AAA – Authentication, Autorization, Accounting

AES – Advanced Encryption Standard

AKA – Authentication and Key Agreement

AP – Access Point

EAP – Extensible Authentication Protocol

EAPOL – EAP Over LAN

ESP – Encapsulating Security Protocol

GTC- Generic Token Card

IKE – Internet Key Exchange

LEAP – Lightweight Extensible Authentication Protocol

MSCHAP – Microsoft Challenge Handshake Authentication Protocol

OTP – One Time Password

PAC – Protected Access Credential

PAX – Password Authenticated Exchange

PEAP – Protected Extensible Authentication Protocol

PPP – Point to Point Protocol

PSK – Pre Shared Key

RFC – Request For Comments

SIM – Subscriber Identity Module

SSID – Service Set Identifier

TKIP – Temporal Key Integrity Protocol

TLS – Transport Level Security

WEP – Wired Equivalent Privacy

Wi-Fi – Wireless - Fidelity

WPA – Wi-Fi Protected Access