Tampereen teknillinen korkeakoulu
Tietotekniikan osasto

Sami Keski-Kasari

Verkkopalveluiden autentikointi yhteisen käyttäjätietokannan avulla

Diplomityö

Aihe hyväksytty osastoneuvoston kokouksessa 10.04.2002
Tarkastajat: Prof. Jarmo Harju
TkT Juha Heinänen

Alkusanat

Olen tehnyt diplomityöni tutkimusapulaisena Tampereen teknillisen korkeakoulun tietoliikennetekniikan laitoksella. Työ on tehty Tampereen Yliopiston Täydennyskoulutuskeskuksen Seinäjoen toimipisteen ja Tampereen teknillisen korkeakoulun tietoliikennetekniikan laitoksen yhteistyöprojektiin. Haluan kiittää työni tarkastajia, prof. Jarmo Harjua ja TkT Juha Heinästä rakentavista kommenteista ja avusta työn rajauksessa. Kiitän myös Heikki Vatiaista oikoluvusta ja sisältöön liittyvistä kommenteista. Lisäksi kiitos Juha Laineelle ja Jussi Lemposelle LaTeX-pohjan virittämisestä, ja Jussi Lemposelle vielä erikseen kiitos LaTeX-ohjeista. Kiitos myös koko WirLabin henkilökunnalle hyvästä työympäristöstä, sekä erityiskiitos Jouni Vuorelalle ja Mika Mustikkamäelle avusta käytännön osuuden toteuttamisessa.

Tampere, 27.05.2002

Sami Keski-Kasari
Lukonmäenkatu 1 C 20
33710 Tampere
Finland
samikk@cs.tut.fi

Tiivistelmä

TAMPEREEN TEKNILLINEN KORKEAKOULU

Tietotekniikan osasto

Tietoliikennetekniikan laitos

Sami Keski-Kasari: Verkkopalveluiden autentikointi yhteisen käyttäjätietokannan avulla

Diplomityö: 54 sivua, 2 liitesivua

Tarkastajat: Prof. Jarmo Harju ja TkT Juha Heinänen

Toukokuu 2002

Avainsanat: AAA, autentikointi, käyttäjätietokanta

Operaattoreiden tarjoamien palveluiden määrän jatkuvasti kasvaessa on käyttäjätietokantojen hallinta muodostumassa ongelmaksi. Lähes jokainen palvelu käyttää omia autentikointikeinojaan, jotka eivät ole keskenään yhteensopivia. Diplomityössä tutkittiin mahdollisuutta yhdistää käyttäjätietokantoja jollain yhteisellä protokollalla. Koska jokaisella operaattorilla on jo valmiina sisäänsoittopalvelussa käytettävä käyttäjätunnuskanta, on niiden hyödyntäminen kannattavaa. Diplomityössä käsitellyssä mallissa asiakkaan ja palvelimen välillä on käytössä halutun palvelun toteuttava protokolla, joka tukee autentikointia. Palvelimen tehtävä on muuttaa autentikointiviestit käytössä olevan AAA (Authentication, Authorization and Accounting)-protokollan mukaisiksi ja lähettää autentikointipyyntö AAA-palvelimelle.

Diplomityön aihealue on melko uusi. Tällä hetkellä määrittelyt eivät ole vielä vakiintuneet, ja lähes jokaisella valmistajalla on käytössään joitain valmistajakohtaisia ratkaisuja. Kehitystyö on kuitenkin käynnissä yhtenäisten standardien aikaansaamiseksi, mikä samalla parantaa eri valmistajien laitteiden välistä yhteensopivuutta. Tässä työssä on pääosin keskitytty näiden yhtenäisten standardien tutkimiseen, eikä valmistajakohtaisia ratkaisuja muutamaa poikkeusta lukuunottamatta ole käsitelty.

Diplomityössä toteutettiin yhdistetty VoIP- ja WLAN-autentikaatiokanta, jonka hallinta tapahtuu WWW-pohjaisella lomakkeella. Yhdistetyn kannan toteutuksessa huomattiin ongelmalliseksi määrittelyjen vakiintumattomuus, mistä johtuen täysin määrittelyjen mukaista toteutusta ei pystytty rakentamaan.

Abstract

TAMPERE UNIVERSITY OF TECHNOLOGY

Department of Information Technology

Institute of Communications Engineering

Sami Keski-Kasari: Authentication of Network Services Using Common User Database

Master of Science Thesis: 54 pages, 2 enclosure pages.

Examiner: Prof. Jarmo Harju and Dr. Juha Heinänen

May 2002

Keywords: AAA, Authentication, User Database

The increasing amount of services offered by network operators is leading to complications due to the fact that almost all services are using different user databases which are incompatible with each other. The possibility of combining user databases was the main target of this thesis. Because all operators already have a user database for their dial-in service, it is reasonable to use it for all services. The architecture considered in this thesis consists of a client, a server, the protocol used for desired service which support authentication; Authentication, Authorization and Accountig (AAA) protocol and an AAA server. The server converts the client's authentication requests to the AAA protocol and sends them to the AAA server.

The field of this thesis is quite new. At this moment specifications are still being defined and almost every manufacturer has proprietary solutions. However there are development efforts to finalize specifications in order to get products compatible with each other. This thesis aims to study these specifications. In addition, a couple of proprietary solutions are described.

A part of the work related to this thesis was the implementation of a combined VoIP and WLAN authentication database which is managed by using a WWW based form. The biggest problems were caused by the lack of finalized specifications and nonproprietary products.

Sisällysluettelo

Alkusanat i

Tiivistelmä ii

Abstract iii

Sisällysluettelo iv

Lyhenneluettelo vi

1 Johdanto 1

2 AAA-palvelut ja protokollat 3

2.1 Point-to-Point Protocol (PPP) 4

2.1.1 Link Control Protocol (LCP) 4

2.1.2 Autentikointiprotokollat 6

2.1.3 Network Control Protocols (NCP) 7

2.2 Extensible Authentication Protocol (EAP) 8

2.2.1 Perustoiminnallisuus 9

2.2.2 EAP ja TLS (Transport Layer Security) 10

2.3 Remote Authentication Dial In User Service (RADIUS) 11

2.3.1 Käyttäjän tunnistus ja valtuutus 11

2.3.2 Tilastointi 14

2.4 Diameter 14

2.4.1 Erot RADIUS-protokollaan 15

2.4.2 Perustoiminnallisuus 15

3 Session Initiation Protocol (SIP) 18

3.1 Komponentit 19

3.1.1 SIP-palvelin 19

3.1.2 SIP-päätelaite 21

3.2 Autentikointi 21

3.2.1 HTTP Digest 22

3.2.2 SIP ja RADIUS 23

3.3 Komponenttien yhteistoiminta 25

3.3.1 Rekisteröinti 25

3.3.2 Yhteyden muodostus 26

4 Langattomat lähiverkot (WLAN) 28

4.1 Perustoiminnallisuus 28

4.2 Autentikointi 30

4.2.1 Wired Equivalency Privacy (WEP) ja autentikointi 30

4.2.2 IEEE 802.1x 32

4.2.3 MS-CHAP EAP -autentikointi 34

4.2.4 EAP SIM-autentikointi 35

5 3G-autentikointi 38
5.1 Radiotielle autentikoituminen 38
5.2 IP Multimedia-palveluun tunnistautuminen 40

6 WirLabin palveluiden toteutus 44
6.1 Yhteisen käyttäjätietokannan toteutus 44
6.2 Käyttäjätietokannan hallinta 45
6.3 SIP-toteutus 48
6.4 WLAN-toteutus 50
6.5 Loppukäyttäjän palvelut 51

7 Yhteenveto 53

Lähdeluettelo 55

Liite A: MD5 -algoritmi 57

Liite B: WEP -algoritmi 58

Lyhenneluettelo

3GPP	3G Partnership Project
AAA	Authentication, Authorization and Accounting
CGI	Common Gateway Interface
CHAP	Challenge Handshake Authentication Protocol
CSCF	Call Session Control Function
DHCP	Dynamic Host Control Protocol
EAP	Extensible Authentication Protocol
EAPOL	EAP over LANs
GPRS	General Packet Radio Service
HLR	Home Location Register
HTTP	Hyper Text Transport Protocol
HSS	Home Subscriber Server
IANA	Internet Assigned Numbers Authority
I-CSCF	Interrogating-CSCF
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IPSec	IP Security
LAN	Local Area Network
LCP	Link Control Protocol
LEAP	Lightweight EAP
MAA	Multimedia-Authentication-Answer
MAR	Multimedia-Authentication-Request
NAS	Network Access Server

NCP	Network Control Protocol
PAP	Password Authentication Protocol
P-CSCF	Proxy-CSCF
PGP	Pretty Good Privacy
PPP	Point to Point Protocol
PSTN	Public Switched Telephone Network
RAA	Registration-Authorization-Answer
RADIUS	Remote Authentication Dial In User Service
RAR	Registration-Authorization-Request
SAA	Server-Assignment-Answer
SAR	Server-Assignment-Request
S-CSCF	Serving-CSCF
SCTP	Stream Control Transmission Protocol
SGSN	Serving GPRS Support Node
SIM	Subscriber Indentifier Module
SIP	Session Initiation Protocol
SMTP	Simple Mail Transfer Protocol
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
WEP	Wired Equivalent Privacy
WLAN	Wireless LAN
WWW	World Wide Web

1 Johdanto

Tämä diplomityö käsittelee verkkopalveluiden autentikointia yhteisen käyttäjätietokannan avulla. Työ on tehty Tampereen teknillisen korkeakoulun ja Tampereen yliopiston täydennyskoulutuskeskuksen Seinäjoen toimipisteen yhteistyöprojektiin. Projektin tarkoituksena oli tutkia mahdollisuuksia yhdistää eri verkkopalveluiden käyttäjän autentikointi, sekä toteuttaa Tampereen yliopiston täydennyskoulutuskeskuksen Seinäjoen toimipisteen WirLab projektiin VoiP- (Voice over IP) ja WLAN- (Wireless LAN) palveluihin käyttäjän autentikointi.

Operaattoreiden tarjoamien palveluiden määrän kasvettua ongelmaksi on muodostonut tiedon monistuminen sekä pirstoutuminen. Operaattorin kannalta on hyödyllistä, että käyttäjän tiedot ovat vain yhdessä keskitetyssä paikassa, jolloin mahdolliset muutokset on helpompi kohdistaa. Lisäksi autentikointia tarvittaessa tiedot löytyvät aina yhdestä paikasta. Palveluiden erilaisen tietorakenteiden ja tallenusmuotojen vuoksi yhteisen toimivan käyttäjätietokannan toteutus vaatii myös yhteisen autentikointiprotokollan määrittelyn. Tällä hetkellä kehityksen kohteena on RADIUS (Remote Authentication Dial In User Service) -protokollan laajentaminen uusien palveluiden vaatimusten mukaiseksi, johtuen sen laajasta käytöstä verkkoon kirjautumisessa. Tulevaisuudessa yhdistävä tekijä saattaisi olla Diameter-protokolla, joka on RADIUS-protokollan jalostetumpi versio.

Palveluita, joissa operaattorit voisivat hyödyntää yhteistä käyttäjätietokantaa, ovat esimerkiksi PPP-yhteyksien autentikoinnit, sähköpostipalvelut, VoIP-palvelut, nykyisin yleistymässä olevat WLAN-palvelut sekä yleisiin tiloihin sijoitettujen verkkoliityntöjen auten-

tikointi. Lisäksi mobiiliverkon laitteiden autentikointi samalla käyttäjätietokannalla tulee mahdolliseksi Diameter-protokollaan tehtyjen parannusten ansiosta.

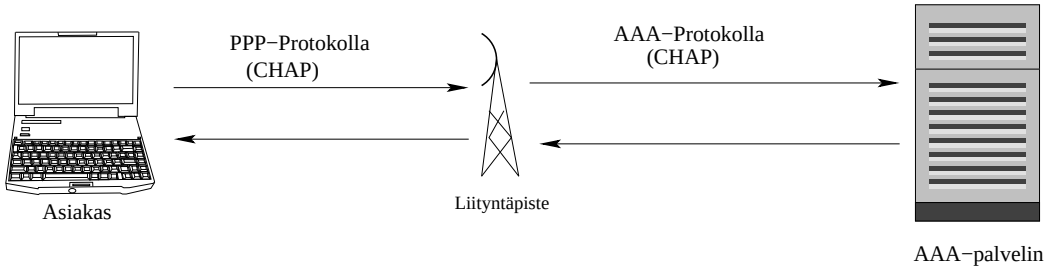
Työ koostuu seitsemästä luvusta ja kahdesta liiteluvusta. Luvussa 2 käsitellään AAA-protokollien kehityksen aloittajaa PPP:tä, sekä sen eri autentikointimenetelmiä. Lisäksi tutustutaan tarkemmin EAP (Extensible Authentication Protocol) -laajennokseen, joka on yleistymässä muuallakin kuin PPP:ssä. Luvussa 2 käsitellään myös AAA-protokollat RADIUS ja Diameter. Luvussa 3 käydään läpi IETF:n (Internet Engineering Task Force) määrittelemä SIP-protokolla, jota tässä työssä käytetään VoIP (Voice over IP)-palvelun toteutuksessa. Luku 4 käsittelee langattomia lähiverkkoja, varsinkin käyttäjän autentikoinnin suhteen. Luvun alussa on lyhyesti kerrottu perustoiminnallisuus, mutta erityisesti luvussa on keskitytty käyttäjän autentikointiin. Luvussa 5 käydään vertailun vuoksi lyhyesti läpi 3G-verkkojen käyttäjän autentikoinnin totetutus. Perustoiminnallisuus 3G-verkoissa on periaatteeltaan sama kuin lähiverkoissa, eli liikenne kulkee IP-protokollan päällä ja multimediapalveluiden signalointi suoritetaan SIP-protokollalla. Kuitenkin 3G:llä on pidempi historia ja GSM-verkoissa on jo totuttu SIM-kortin käyttöön, joten periaatteessa luotettava tunnistaminen on helpompi toteuttaa. Luvussa 6 on kerrottu WirLabin palveluiden toteutuksesta ja luvussa 7 on yhteenveto. Liitteessä A käsitellään yleistä teoriaa autentikointiprotokollissa yleisessä käytössä olevasta MD5-algoritmista ja liitteessä B on esitelty tarkemmin langattoman lähiverkon määrittelyssä käytetty WEP-salausprotokolla.

2 AAA-palvelut ja protokollat

AAA-lyhenne tulee englanninkielisistä sanoista Authentication, Authorization ja Accounting. Näitä vastaavat suomennokset ovat todentaminen tai autentikointi, valtuutus ja tilastointi. Autentikointipalvelu mahdollistaa käyttäjien tunnistuksen. Valtuutuspalvelun avulla käyttäjien saamia palveluja pystytään profiloimaan. Tilastointipalvelun avulla pystytään keräämään käyttäjistä tilastotietoja, kuten esimerkiksi yhteysaikoja. Tässä luvussa AAA-palvelun toteuttavista protokollista käsitellään jo nykyisin operaattoreilla yleisesti PPP-yhteyksissä käytössä olevaa RADIUS-protokollaa, sekä 3G- ja IPv6-verkoissa käyttöön tulevaa Diameter-protokollaa.

AAA-protokollaa käytettäessä palvelu koostuu yleensä kolmesta eri komponentista: asiakkaasta, liityntäpisteestä ja AAA-palvelimesta. AAA-protokollat ovat asiakkaalle näkymättömiä. Asiakkaan ja liityntäpisteen välissä on yleensä käytössä joku alemman tason protokolla, joka tarjoaa käyttäjän autentikointipalvelun. Esimerkiksi tässä luvussa myöhemmin esiteltävä PPP on protokolla, jossa alkuperäisessä määrittelyssä [1] on mukana PAP- ja CHAP-autentikointiprotokollat, sekä jälkeenpäin määritelty [2] EAP-protokolla. Liityntäpiste välittää autentikointiprotokollan parametrit AAA-protokollalla AAA-palvelimelle, joka suorittaa varsinaisen käyttäjän tunnistuksen. Tässä luvussa esiteltyjen AAA-protokollien, RADIUS ja Diameter, pääasiallinen tarkoitus on kuljettaa turvallisesti edellä mainittujen autentikointiprotokollien parametrit AAA-palvelimelle. Näin käyttäjän laite pysyy mahdollisimman yksinkertaisena, eikä sen tarvitse ottaa kantaa käytettävään AAA-protokollaan. Esimerkki AAA-palvelun rakenteesta on esitetty kuvassa 2.1. Kuvassa asiakkaan ja liintyntäpisteen välissä on käytössä PPP-protokolla ja autentikointi-

protokollana CHAP-protokolla.



Kuva 2.1: Esimerkki AAA-palvelun rakenteesta

2.1 Point-to-Point Protocol (PPP)

PPP-protokolla on määritelty IETF:n RFC:ssä 1661 [1]. PPP-protokolla on palveluntarjoajilla yleisessä käytössä Internet-yhteyksien tarjoamisessa. Internetin yleistyessä käyttäjien määrä lisääntyi, ja palveluntarjoajat joutuivat lisäämään soittosarjoja, jolloin käyttäjän todentaminen nousi ongelmaksi. Paikalliset käyttäjätietokannat olivat huonosti skaalautuvia, ja ne jouduttiin kopioimaan kaikkialle. AAA-protokollien kehitys sai alkunsa tästä. PPP-protokollaa voidaan siis pitää AAA-protokollien kehityksen käynnistäjänä. Tämän vuoksi tässä alaluvussa käydään lyhyesti läpi PPP-protokollan perustoiminnallisuus.

2.1.1 Link Control Protocol (LCP)

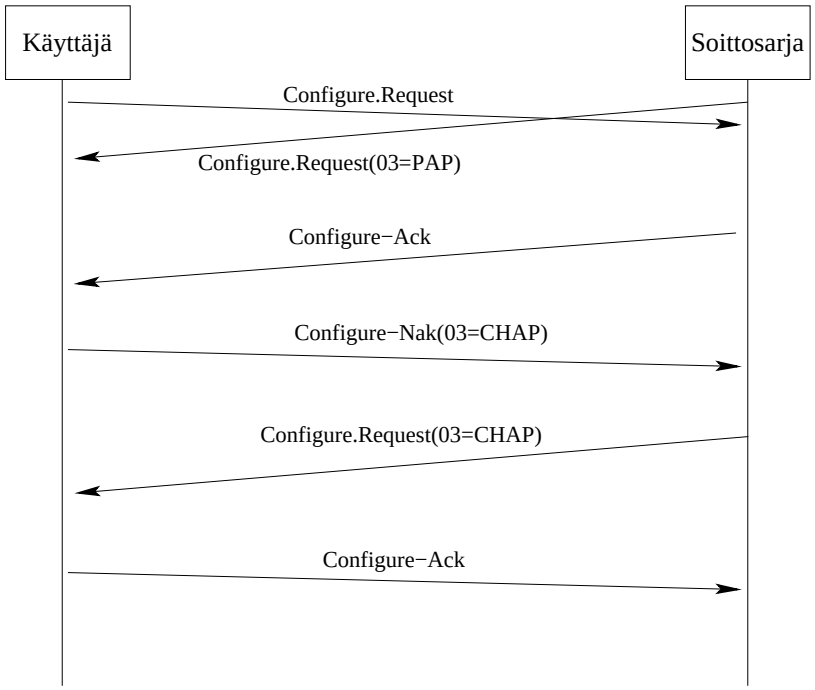
PPP:n voidaan ajatella jakautuvan kahteen pääkokonaisuuteen: linkkikerroksen ja verkkokerroksen hallintaan. Linkkikerroksen hallintaprotokolla esitellään tässä alaluvussa ja verkkokerroksen vastaava seuraavassa alaluvussa. Linkkikerroksen hallintaprotokollan tehtäviä ovat fyysisen yhteyden muodostamisen jälkeen yhteysparametrien neuvottelu osapuolten välille sekä mahdollinen käyttäjän tunnistaminen.

Kun osapuolten välille on saatu fyysinen yhteys, esimerkiksi kantoaalto on tunnistettu, on LCP-protokollan vuoro aloittaa PPP-yhteyden muodostus. Linkkikerroksella yhteysparametrit neuvotellaan aina molempiin suuntiin erikseen ja toisistaan riippumatta. Tämän vuoksi yhteys ei ole välttämättä symmetrinen parametrien osalta. Toisaalta nykyisin monet PPP-yhteyttä käyttävät laitteet eivät muodosta symmetristä fyysisen tason yhteyttä. Tällöin on tärkeää, että voidaan käyttää eri parametreja linkin eri suuntiin.

Mahdollisia linkkikerroksen optioita on useita. Tärkeimpiä niistä ovat optiot numeroilla 01,02 ja 03. Optiolla numero 01 neuvotellaan suurin siirtokehyksen koko. Oletusarvoinen siirtokehyksen koko on 1500, mutta riippuen linkin nopeudesta sitä voidaan joko pienentää tai suurentaa. Optiolla 02 neuvotellaan käytetty yhteyden hallintatapa, esimerkiksi mahdolliset vuon ohjaukset. Optiolla 03 neuvotellaan käyttäjän tunnistusprotokolla. Vaihtoehtoja ovat PAP, CHAP, EAP, sekä muut valmistajakohtaiset ratkaisut. Kyseinen optio voidaan jättää myös pois, jolloin käytössä ei ole tunnistamista ollenkaan. Usein soittosarja pyytää LCP-protokollan optioissa jotain tunnistautumisprotokollaa, mutta käyttäjältä soittosarjalle lähetetyissä viesteissä tätä optiota ei ole. Jos käyttäjä haluaa olla varma myös verkon oikeellisuudesta, pyytää hänkin verkkoa tunnistautumaan. Muissa optioissa on käsitelty erilaisia otsikon ja viestin pakkausmuotoja ja erilaisia laatuparametrien neuvotteluja. Yksi yrityksissä yleisessä käytössä oleva optio on 0D, jolla käyttäjä voi pyytää tunnistautumisen jälkeen soittosarjaa soittamaan takaisin. Tällä takaisinsoittopalvelulla on mahdollista toteuttaa helposti yrityksen etätyöyhteydet, jolloin yhteydestä kertyvä lasku menee suoraan yritykselle.

Mahdollisista viesteistä tärkeimmät ovat Configure-Request, Configure-Ack, Configure-Nak ja Configure-Reject. Configure-Request-viestillä aloitetaan neuvottelu. Viestiin laitetaan parametreiksi halutut optiot haluttuilla arvoilla. Configure-Ack-viestillä hyväksytään kaikki optiot, sekä niiden arvot. Jos vastapää ei tue jotain optiota, lähettää se Configure-Reject-viestin, jossa on parametrinä kyseinen optio. Tämän jälkeen alkuperäisen lähettäjän täytyy lähettää uusi Configure-Request-viesti, jossa kyseistä optiota ei ole. Vastaanottajan tukiessa jotain optiota, mutta ei pyynnössä saatua arvoa, lähetetään Configure-Nak-viesti. Configure-Nak-viestissä on parametrinä optio sekä arvo, jota vastaanottaja haluaisi käyttää. Tämän jälkeen alkuperäinen lähettäjä päättää haluaako se käyttää kyseistä arvoa. Halutessaan käyttää tätä arvoa, lähettää se uuden Configure-Request-viestin muutetulla option arvolla, muuten yhteyden muodostus keskeytetään.

Kuvassa 2.2 on esitetty neuvottelutapahtuma. Kumpikin puoli lähettää kantoaallon tunnistuksen jälkeen Configure-Request viestin. Käyttäjällä ei ole mitään erityistoiveita optioiden suhteen. Soittosarja haluaa käyttäjäntunnistusta, joten sen parametreissä kulkee optio 03 jonka arvona on PAP-protokollan tunnus. Soittosarja hyväksyy käyttäjän Configure-Request viestin sellaisenaan lähettämällä Configure-Ack -viestin. Käyttäjä haluaa tunnistamisen tapahtuvan CHAP-protokollalla, joten soittosarjalle lähetetään Configure-Nak. Viestissä tunnistusprotokollan option arvoksi on vaihdettu CHAP. Soittosarja hyväksyy CHAP-



Kuva 2.2: Linkkikerroksen parametrin neuvottelutapahtuma

protokollan käytön, jolloin se lähttää uuden Configure-Request -viestin, jossa toivottu tunnistusprotokolla on CHAP. Tämän jälkeen käyttäjä hyväksyy yhteyden ja lähettää kuittauksen. Nyt linkkikerroksen yhteys on muodostettu ja alkaa käyttäjän tunnistamisen osuus.

2.1.2 Autentikointiprotokollat

Tässä yhteydessä käydään läpi PPP:n määrittelyn [1] mukaiset autentikointiprotokollat PAP ja CHAP. EAP-protokolla käsitellään myöhemmin omassa alaluvussaan, koska sitä käytetään muissakin protokollissa kuin PPP:ssä. PAP-protokolla on vanhin tunnistuksessa käytetty protokolla. Siinä ei ole kiinnitetty huomiota tietoturvanäkökohtiin, vaan käyttäjätunnus ja salasana kulkevat selkokielisenä siirtotiellä. CHAP-protokolla on paranneltu versio PAP:sta. Siinä salasana kulkee aiemmin linkkitasolla neuvotellulla algoritmillä salattuna. Usein tämä algoritmi on MD5, jota on käsitelty tarkemmin liitteessä 1.

PAP-protokollassa on ainoastaan kolme erilaista viestityyppiä. Viestityypit ovat Authentication-Request, Authentication-Ack ja Authentication-Nak. Authentication-Request -viestillä käyttäjä ilmoittaa halustaan tunnistautua. Authentication-Request -viestissä on parametreinä käyttäjätunnus ja salasana. Jos viestissä olleet käyttäjätunnus ja

salasana ovat samoja kuin vastapäässä määritellyt, katsotaan tunnistautuminen onnistuneeksi. Tällöin vastaanottaja lähettää Authentication-Ack -viestin, ja tunnistautuminen katsotaan onnistuneeksi. Muulloin se lähettää Authentication-Nak -viestin. Authentication-Nak -viestissä voi olla parametrinä syy tunnistamisen epäonnistumiseen.

CHAP -protokollassa viestityyppejä on 4. Nämä ovat Challenge, Response, Success ja Failure. Toisin kuin PAP-protokollassa, CHAP:issa tunnistamista vaativa osapuoli lähettää ensimmäiseksi Challenge -viestin, jossa se pyytää käyttäjältä käyttäjätunnusta ja salasanaa, sekä antaa haasteen. Salasana salataan saadulla haasteella käyttäen esimerkiksi MD5-algoritmia. Tämän jälkeen käyttäjä lähettää Response -viestin, jossa on käyttäjätunnus sekä salasanasta laskettu tarkistussumma. Jos tarkistussumma tunnistamista pyytäneellä laitteella täsmää katsotaan tunnistaminen onnistuneeksi ja lähetetään Success-viesti. Muuten lähetetään Failure viesti, jossa on parametrinä epäonnistumisen syy.

CHAP-protokollaa ei voi automaattisesti pitää parempana ratkaisuna kuin PAP-protokollaa, vaikka tilanne on usein näin. Kuitenkin CHAP-protokollaa käytettäessa salasanan täytyy olla molemmilla osapuolilla selkokielisessä muodossa. Tällöin erilaiset SecureID-tyyppiset, ainoastaan kerran kelpaavat salasanat eivät välttämättä toimi CHAP:n kanssa. Tällöin vaihtoehdoksi jää käyttää PAP-protokollaa tai jotain muuta ratkaisua, kuten EAP-protokollaa.

2.1.3 Network Control Protocols (NCP)

Verkkokerroksen hallintaan on olemassa useampia hallintaprotokollia, riippuen verkkokerroksella käytössä olevasta protokollasta. Tässä luvussa käsittellään ainoastaan IPCP (IP Control Protocol) -protokolla, joka on IP-protokollan vuoksi yleisimmin käytössä. IPCP on määritelty IETF:n RFC:ssä 1332 [3].

Kun linkkitason optiot on määritelty, sekä mahdollinen tunnistaminen suoritettu onnistuneesti, on verkkotason hallintaprotokollan vuoro muodostaa yhteys loppuun. Voidakseen liikennöidä, laite tarvitsee IP-osoitteen, jolloin tärkein IPCP-protokollan tehtävä on tämän vaatimuksen täyttäminen. IPCP:ssä tähän on kaksi keinoa, osoite saadaan joko itse protokollan avulla tai osoite haetaan esimerkiksi DHCP-palvelimelta PPP-yhteyden avaamisen jälkeen. Toinen tärkeä määriteltävä ominaisuus on pakkausprotokollan käyttäminen. Tämä on hyödyllistä hitailla linkeillä, joiden päätelaitteilla on hyvä laskentateho.

IPCP:ssä on määritelty samat neljä perusviestityyppiä kuin LCP:llekin. Configure-Request-viestissä menee ainoastaan pyydetty IP-osoite ja vastauksessa tulee siihen joko kuittaus tai uusi IP-osoite. Access-Reject-viesti keskeyttää koko protokollan toiminnan, ja linkki avataan ilman IPCP-protokollaa. Laite voi myös itse ehdottaa IP-osoitettaan, jolloin vastaanottaja tarkistaa tietokannastaan tai RADIUS-palvelimelta, että kyseinen asiakas voi käyttää ehdottamaansa IP-osoitetta. Jos osoite on sallittu, lähettää vastaanottaja kuittauksen Configure-Ack-viestillä. Yleisempi tilanne on se, että laitteella ei ole valmiiksi asetettu IP- osoitetta. Tällöin Configure-Request viestissä lähetetään IP-osoite optio arvolla 0. Tällöin vastaanottaja voi joko antaa omasta osoite-poolistaan IP-osoitteen Configure-Nak-viestissä, palauttaa Configure-Ack:n, jolloin hyväksytään asiakkaan ehdottama IP-osoite tai palaluttaa Configure-Reject:in, jolloin IPCP-protokollan toiminta keskeytetään. Ensin mainitussa tilanteessa asiakkaalle annetaan IPCP-protokollalla IP-osoite ja liikennöinti voi alkaa. Kahdessa jälkimmäisessä tapauksessa PPP-yhteys avataan, mutta käyttäjällä ei ole vielä IP-osoitetta, joten se ei voi vielä liikennöidä. Tällöin asiakkaan täytyy käyttää esimerkiksi DHCP:tä IP-osoitteen saamiseen. Asiakas lähettää DHCP-request viestin PPP-linkille ja vastapäässä oleva DHCP-palvelin palauttaa asiakkaalle IP-osoitteen. DHCP-tapaus on yleisimmin käytössä, koska sen avulla käyttäjälle voidaan antaa IP-osoitteen lisäksi nimipalvelimen osoitteet, aliverkon peite, sekä muiden palvelimien osoitteita.

2.2 Extensible Authentication Protocol (EAP)

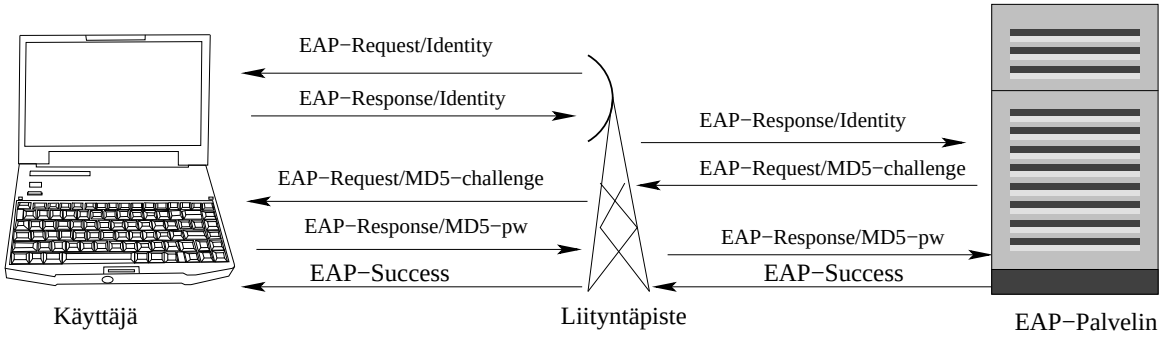
EAP-protokolla on alunperin määritelty laajennukseksi PPP-protokollaan. Sen avulla PPP-yhteyksille on mahdollista helpommin lisätä uusia tapoja käyttäjän tunnistamiseen. Nykyisin protokollan yksinkertaisuuden ja joustavuuden vuoksi se on tulossa käyttöön muissakin ympäristöissä, esimerkiksi langallisissa ja langattomissa lähiverkoissa. Alkuperäisessä IETF:n RFC:ssä 2284 [14] ei ole otettu kantaa protokollan käyttöön muualla kuin PPP-yhteyksillä. Nykyisessä määrittelyvaiheessa olevassa IETF:n draftissa [2] protokollaa on muutettu paremmin yhteensopivaksi muidenkin protokollien kanssa. PPP-taustansa vuoksi EAP-protokollaa käytetään yleensä jonkun AAA-protokollan kanssa, jolloin saadaan automaattisesti todentamis-, valtuutus- ja tilastointipalvelut käyttöön verkkotyppistä riippumatta. Esimerkiksi lähiverkoissa palvelun toteuttaminen muilla keinoilla on on hankalaa.

2.2.1 Perustoiminnallisuus

Tunnistaminen suoritetaan ennen varsinaisen dataliikenteen alkua. Muissa kuin IP-tunneleissa EAP-protokolla ei mene IP:n päällä, vaan alemman kerroksen protokollan päällä. Esimerkiksi PPP:ssä Link Control -protokollan päällä ja lähiverkoissa Ethernetin päällä.

Kuva 2.3: EAP-viestin otsikon rakenne

Kuvassa 2.3 on esitetty EAP-viestin otsikon rakenne. Mahdollisia koodeja on 4. Koodilla 1 ilmoitetaan pyyntö. Sitä käytetään kun halutaan ilmoittaa käyttäjälle tunnistautumisen tarpeesta. Koodia 2 käytetään pyyntöihin vastatessa. Koodilla 3 ilmoitetaan tunnistamisen onnistumisesta ja 4:llä epäonnistumisesta. Tunnistekentän avulla tunnistetaan toisiinsa liittyvät viestit ja pituus kertoo EAP-viestin koko pituuden. Koodeilla 3 ja 4 ei ole datakenttää, koska ne ovat ainoastaan ilmoituksia. Koodeilla 1 ja 2 datakentässä kulkee ilmoitus halutusta toimenpiteestä. Tällöin alussa on 8-bittinen tyyppikenttä, joka kertoo kyseessä olevan pyynnön ja sen jälkeen pyyntöön liittyvä data. Tällä hetkellä määriteltyinä ovat tyyppikentän arvot 1 Identity, 2 Notification, 3 Nak, 4 MD5-Challenge, joita kaikkien toteutusten täytyy tukea. Lisäksi on valmistajakohtaisia tyyppikentän arvoja.



Kuva 2.4: Tunnistautumistapahtuma EAP-protokollalla

Kuvassa 2.4 on kuvattu tunnistautumistapahtuma. Liityntäpisteen ja palvelimen väliset EAP-sanomat kuljetetaan yleensä jonkun AAA-protokollan sisällä, ja usein EAP-palvelin on yhdistetty AAA-palvelimeen. Kun liityntäpisteen ja käyttäjän laitteen välille on muodostunut fyysinen yhteys, lähettää liityntäpiste EAP-request -viestin, jossa se pyytää käyttäjää tunnistautumaan. Tämän jälkeen käyttäjä lähettää EAP-response -viestin, jossa on pa-

rametrina käyttäjän tunniste, esim. käyttäjätunnus. Tämä vaihe voidaan jättää pois, jos liityntäpiste pystyy tunnistamaan käyttäjän muulla tavalla. Vastauksen jälkeen liityntäpiste lähettää tunnistamispyynnön EAP-palvelimelle. Palvelimelta tulee EAP-request, jossa tyyppikentässä on jokin haaste, esimerkiksi MD5-haaste, jonka liityntäpiste välittää käyttäjälle. Haasteen saannin jälkeen käyttäjän laite laskee salasanasta haasteella tarkistussumman, jonka se liittää vastaukseen. Liityntäpiste välittää jälleen viestin palvelimelle, joka laskee samalla haasteella omassa kannassan olevasta salasanasta tarkisteen ja vertaa niitä. Jos tarkistussummat ovat samoja, lähettää palvelin EAP-Success-viestin, jolloin liityntäpiste tietää käyttäjän tunnistautuneen ja välittää viestin käyttäjälle sekä päästää dataliikenteen kulkemaan. Muussa tapauksessa palvelin lähettää EAP-Failure-viestin, jolloin tunnistautuminen ei onnistunut ja käyttäjää ei päästetä verkkoon.

2.2.2 EAP ja TLS (Transport Layer Security)

PPP-taustasta ja yksinkertaisuudesta on myös haittaa. Protokollasuunnittelussa on lähdetty ajatuksesta, että liityntäpisteen ja käyttäjän välillä on fyysisen kerroksen pisteestäpisteeseen -yhteys. Tämän vuoksi protokollaan ei ole määritelty mitään salausta, ja tunnistaminenkin on vain yksisuuntainen, ainoastaan käyttäjä autentikoituu verkkoon. PPP-yhteyksillä ja langallisissa lähiverkoissa tämä toimii, koska mahdollisuus liikenteen kaappaamiseen vaatii pääsyn fyysiseen siirtotiehen. Langattomissa verkoissa tämä oletus on ongelmallisempi, koska tällä hetkellä niissä ei ole toimivaa ja varmaa fyysisen siirtotien salausta. Ongelmallisia ovat myös IP-protokollan päälle rakennetut tunnelit, joissa viestit voivat kiertää julkisen Internetin kautta. Näistä syistä johtuen on lähdetty määrittelemään tapoja, joilla EAP-liikenne kulkisi TLS:n avulla salattuna ja suojattuna.

EAP TLS on IETF:n RFC 2716 [4]. EAP TLS mahdollistaa kaksisuuntaisen tunnistamisen, sekä EAP-liikenteen salauksen ja sisällön suojaamisen. Toiminnaltaan EAP TLS on samanlainen kuin EAP-autentikointi normaalisti, mutta autentikointi tapahtuu sertifikaattien pohjalta, jotka vaihdetaan neuvotteluvaiheessa. Varsinkin EAP-asiakkaan kannalta sertifikaatin varmennus on usein ongelmallinen, koska sillä ei yleensä ole autentikointivaiheessa vielä Internet-yhteyttä, jolloin se ei pysty varmistamaan palvelimen sertifikaatin tuoreutta ja oikeellisuutta. Tämän vuoksi on olemassa kaksi IETF:n draft-määrittelyä: PEAP (Protected EAP) [5] ja EAP-TTLS (EAP Tunneled TLS) [6], joissa muodostetaan TLS:n avulla

salattu ja suojattu yhteys, jonka sisällä autentikointi suoritetaan jollain EAP:n autentikointimenetelmällä, kuten MD5-haasteella. EAP-TTLS laajentaa autentikointimenetelmiä myös EAP:n ulkopuolelle. Siinä on mahdollista muodostaa salattu yhteys EAP-TLS:n avulla ja suorittaa sen sisällä autentikointi muullakin kuin EAP-protokollalla, esimerkiksi PAP:lla tai CHAP:lla.

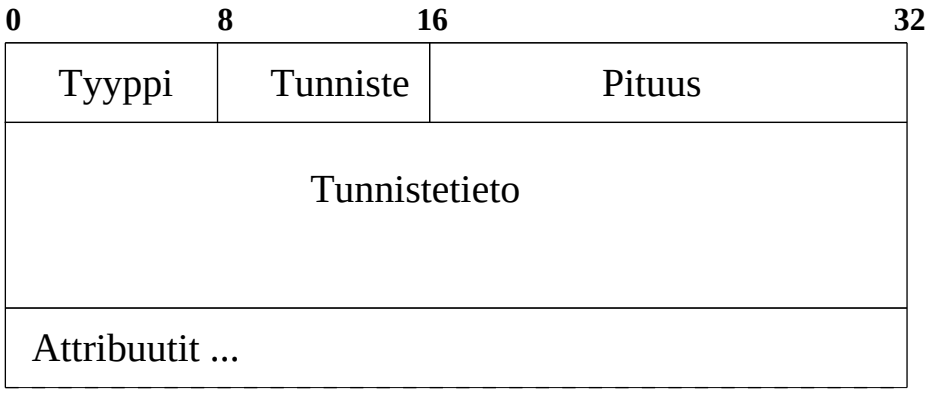
2.3 Remote Authentication Dial In User Service (RADIUS)

RADIUS-protokolla on aikoinaan suunniteltu sisäänsoittopalveluissa tapahtuvaan tunnistukseen, jossa se on nykyäänkin laajassa käytössä. Alunperin Livingston -yhtiössä lähdettiin kehittämään RADIUS-protokollaa. Muut valmistajat kehittivät vastaavia omia protokolliaan. Livingston kuitenkin julkaisi protokollan määrittelyn, jolloin siitä tuli IETF:n RFC. Nykyinen määrittely tunnistuksen ja valtuutuksen osalta on RFC 2865 [7]. Uusin tilastoinnin määrittely on RFC 2866 [8]. RADIUS -protokollan pääasiallinen käyttökohde on operaattorin sisäisessä verkossa, jolloin verkkoa voidaan pitää kohtuullisen luotettavana ja yhden tahon ylläpitämänä. Tämän vuoksi RADIUS-protokollassa voidaan käyttää viestien välitykseen UDP-protkollaa, sekä luoda laitteiden väliset luottamussuhteet kiinteästi.

2.3.1 Käyttäjän tunnistus ja valtuutus

RADIUS -viestit noudattavat kuvassa 2.5 olevaa rakennetta. Tyyppikenttä määrittelee viestin tyypin. Tyyppikentän arvo 1 on Access-Request, 2 on Access-Accept, 3 on Access-Reject ja 11 on Access-Challenge. Tunnistekentän avulla tunnistetaan toisiinsa liittyvät paketit. Tunnistetietokenttä on 16-tavuinen satunnaisluku, jota käytetään palvelimen vastausten tunnistamiseen sekä salasanojen salaamiseen. Attribuutit -kenttä on vaihtelevan pituinen ja sisältää pyyntöön liittyviä attribuutteja, kuten IP -osoitetietoja ja muita tunnistamisessa ja valtuutuksessa tarvittavia ja käytettäviä tietoja.

Tunnistetietokenttää käytetään kahdessa merkityksessä. Access-Request-viesteissä sitä pidetään pyynnön tunnistetietokenttänä, muissa vastauksen tunnistetietokenttänä. Pyynnön tunnistetietokenttänä käytettäessä arvo on 16-bittinen satunnaisluku, jonka avulla käyttäjän



Kuva 2.5: RADIUS-viestin muoto

salasana salataan. Vastauksen tunnistetietokentän arvo saadaan laskemalla MD5-summa koko paketin ja asiakaslaitteen ja palvelimen välisen yhteisen salasanan summan yli. Summalla tässä yhteydessä tarkoitetaan, että tekstikentät on ketjutettu peräkkäin. MD5-algoritmi on tarkemmin esitelty liitteessä A.

Käyttäjän kirjautuessa RADIUS-tunnistusta vaativaan järjestelmään, liityntäpiste lähettää RADIUS-palvelimelle Access-Request -viestin, jossa on parametreina käyttäjätunnus, salasana MD5-salattuna, asiakaslaitteen tunnus sekä portti johon käyttäjä on pyrkimässä.

RADIUS-palvelimen saatua viestin se varmistaa ensin, että kyseisen liityntäpisteen IP-osoite ja salasana on määritelty palvelimelle. Jos määrittely on tehty, pyyntö käsitellään. Ensimmäisenä palvelin kysyy tietokannasta käyttäjätunnusta vastaavat tiedot. Tietokannasta löytyvät vaatimukset, jotka käyttäjän täytyy täyttää. Näihin vaatimuksiin kuuluu aina salasana, mutta ne voivat sisältää muutakin informaatiota, kuten porttinumerot ja liityntäpisteet, jonne käyttäjällä on mahdollista päästä.

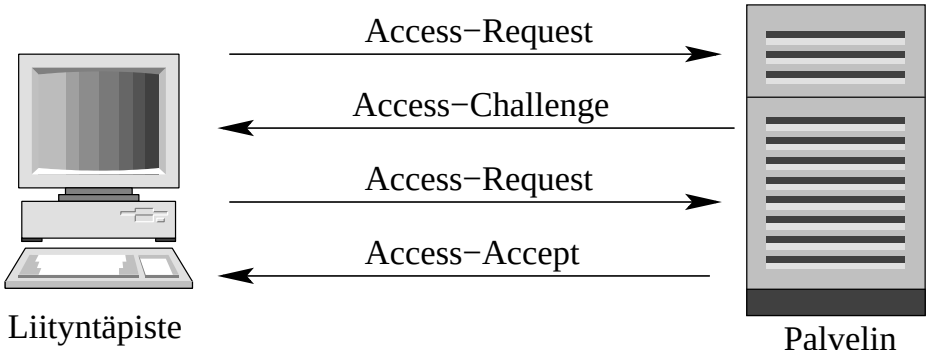
Jos edellä mainitut vaatimukset eivät täyty, palvelin lähettää Access-Reject-viestin. Muussa tapauksessa palvelin voi joko lähettää Access-Challenge-viestin tai hyväksyä yhteyden.

Haaste/vastaus-tunnistamisessa haaste on ennustamaton satunnaisluku, jolla käyttäjän täytyy salata salasanansa ja lähettää se vastauksena. Usein tämä toimii siten, että kun käyttäjä kirjautuessaan syöttää käyttäjätunnuksensa ja salasanansa niin ensimmäisessä Access-Request-viestissä läheteteään ainoastaan käyttäjätunnus. Access-Challenge-viestin jälkeen asetetaan User-Password-attribuutti laskemalla MD5-summa laitteiden välisen salasanan ja pyynnön tunnistekentän avulla, jonka jälkeen se XOR:ataan käyttäjän antaman salasanan

kanssa.

Kun tunnistus on onnistunut, sekä muutkin vaatimukset täyttyvät, palvelin lähettää Access-Accept-viestin, jossa on parametreina palvelun tyyppi ja muut palvelun määritykset. Näitä ovat esimerkiksi PPP:llä IP-osoite, suurin paketin pituus.

Kuvassa 2.6 on esitetty onnistunut tunnistautuminen, jossa liityntäpiste lähettää Access-Request-viestissä parametrit NAS-Identifier, NAS-Port, Username, ja User-Password, jossa User-Password voi olla tyhjä tai vain jokin merkkijono. Tämän jälkeen palvelin tunnistaa liityntäpisteen lailliseksi, ja löytää käyttäjätunnuksen kannastaan, joten se lähettää Access-Challenge-viestin. Viestin saatuaan liityntäpiste lähettää uudelleen Access-Request-viestin, jossa on User-Password kentän arvo määritellyllä tavalla. Tämän jälkeen molemmissa päissä lasketut MD5-summat täsmäävät ja käyttäjä on autentikoitu, joten palvelin lähettää Access-Accept-viestin.



Kuva 2.6: Tunnistus- ja valtuutusprosessi

RADIUS-palvelin voi välittää myös tunnistautumispyyntöjä eteenpäin toisille RADIUS-palvelimille. Tällaisesta mahdollisuudesta on hyötyä silloin, jos käyttäjällä on mahdollisuus liikkua useamman palveluntarjoajan verkoissa. Tällöin käyttäjän tiedot ovat ainoastaan palvelun myyvän operaattorin RADIUS-kannassa. Asiakkaan ollessa toisen palveluntarjoajan verkossa, kyseisen palveluntarjoajan RADIUS-palvelin ohjaa kyselyt palvelua myyvän operaattorin RADIUS-palvelimelle. Tästä on se hyöty, että käyttäjän ei tarvitse tehdä sopimuksia kuin yhden palveluntarjoajan kanssa, ja palveluntarjoajien väliset sopimukset laajentavat käyttäjän aluetta.

2.3.2 Tilastointi

RADIUS -protokollassa oli alunperin määriteltynä vain käyttäjän tunnistus ja valtuutus. Myöhemmin siihen on lisätty tilastointituki, joka on määritelty omassa RFC:ssään [8].

Tilastointi on varsinkin operaattoreiden kannalta tärkeä ominaisuus. Sen avulla voi esimerkiksi kerätä laskutustietoa yhteysajoista, tilastotietoa siirretyistä tavumääristä ja käytössä olleet IP-osoitteet. Viimeksi mainittuja voi käyttää sekä laskutukseen että yhteysongelmien selvittämiseen.

Tilastoinnin osalta viestin kehyksen muoto on samanlainen kuin kuvassa 2.5. Tyyppikoodi 4 määrittelee, että kyseessä on Accounting-Request-viesti, ja tyyppikoodi 5 Accounting-Response-viestin. Koska RADIUS-Accounting viestissä ei käytetä käyttäjältä vaadittavia salasanoja, pyynnön tunnistetietokentän arvo lasketaan summaamalla tyyppikenttä, tunnistekenttä, pituuskenttä, 16 nollatavua, attribuutit ja laitteiden välinen salasana, jonka jälkeen tämän summan yli lasketaan MD5-summa. Vastauksen tunnistetietokentän arvo lasketaan summaamalla tyyppikenttä,tunnistekenttä, pituuskenttä, tunnistetietokenttä, sekä laitteiden välinen salasana. Tunnistetietokentän arvo on sama kuin pyynnössä ollut tunnistetietokentän arvo.

Haluttaessa käyttää tilastointia, tunnistamisen jälkeen lähetetään Accounting-Request-viesti, jossa on parametreina kayttäjätunnus ja palvelun tyyppi sekä kerrotaan että uusi istunto alkaa. Tämän jälkeen palvelin lähettää kuittauksen Accounting-Response-viestillä. Kun käyttäjä lopettaa istuntonsa, lähetetään Accounting-Request-viesti, jossa kerrotaan että istunto loppuu, sekä istunnon kestoaika ja siirretyt datamäärät.

2.4 Diameter

Diameter-protokolla on huomattavasti skaalautuvampi ja helpommin laajennettavissa oleva protokolla kuin RADIUS. Lisäksi siinä on otettu paremmin huomioon mobiiliverkot, sekä erityisesti niihin liittyvät vaatimukset. Lisäksi Diameter käyttää siirtoprotokollanaan TCP- tai SCTP-protokollaa, jotka takaavat luotettavan tiedonsiirtokanavan, ja mahdollistavat salauksen. Diameter-protokolla on vasta määrittelyvaiheessa, tällä hetkellä uusin määrittely

on IETF:n draft-versio 09 [9].

2.4.1 Erot RADIUS-protokollaan

Diameter-protokollan perusajatus on erilainen kuin RADIUS-protokollan. Diameter- protokolla ei itsessään määrittele parametreja joilla tunnistaminen, valtuutus ja tilastointi tapahtuvat. Diameter-protokolla määrittelee vain viestien muodon sekä lähetystavan. Käytettävät parametrit riippuvat sovelluksesta, ja ne pitää rekisteröidä IANA:lle (Internet Assigned Numbers Authority). Rekisteröinnin yhteydessä IANA antaa sovellukselle ja mahdollisille uusille parametreille tunnistenumerot.

Diameter-protokolla sallii myös palvelimen puolelta aloitetut tunnistamisprosessit, jolloin RADIUS-mallissa esiintynyt asiakas/palvelin -malli hämärtyy. Tämän vuoksi Diameter on asiakkaiden välinen protokolla. Lisäksi osapuolet pystyvät neuvottelemaan käytettävistä parametreistä ja dynaamisesti etsimään protokollaa osaavia laitteita, mikä vähentää tarvittavaa konfigurointia. Diameter -protokollassa on huomioitu myös tarkemmin virhetilanteet, sekä niistä toipuminen.

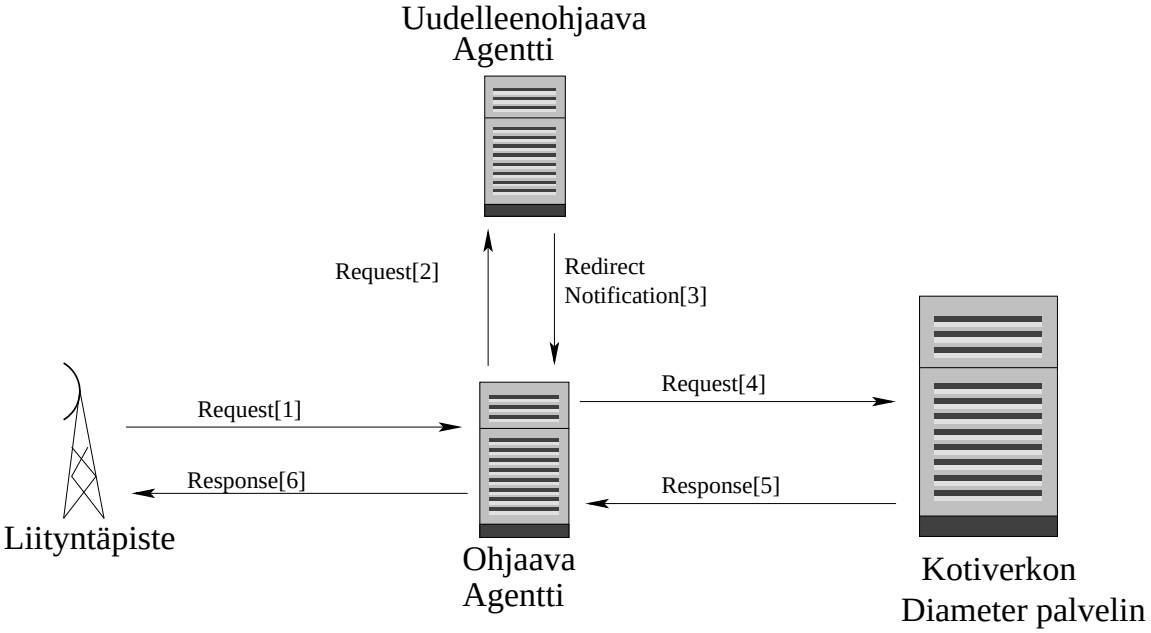
2.4.2 Perustoiminnallisuus

Kuten edellisessä luvussa mainittiin, Diameterin perusprotokollaa käytetään ainoastaan sovelluksen kautta. Sovelluksissa määritellään käytettävät parametrit, sekä tarkoitus johon sitä käytetään. Kaikkien Diameter -toteutusten täytyy kuitenkin tukea perusprotokollaa, koska viestien siirtotapa ja muoto on määritelty siinä. Sovellusten tukeminen on toteuttajan vastuulla, eikä käyttäjän kannalta turhia sovelluksia tarvitse toteuttaa. Tällainen ominaisuus mahdollistaa helpon laajennettavuuden, sekä helpottaa omia valmistajakohtaisia kokeiluja. Tällä hetkellä kaksi merkittävintä sovellusta ovat PPP-yhteyksillä käytettävä NASREQ sekä mobiiliverkoissa käytettävä Mobile-IP. Molemmat sovellukset ovat myös vielä määrittelyvaiheessa.

Diameter -protokollassa on määritelty käsite agentti. Sen tehtävänä on ohjata muuhun kuin omaan verkkoon kuuluvia viestejä, sekä tarvittaessa tehdä muunnoksia Diameterin ja muiden AAA -protokollien välillä. Tällainen toiminnallisuus on ensiarvoisen tärkeää mobiili-

verkoissa, joissa halutaan, että palveluntarjoajan asiakkaat voivat käyttää palveluitaan myös toisen palveluntarjoajan verkossa. Koska asiakkaita on paljon, ei kaikkea tietoa ole järkevä kopioida kaikkialle, vaan käyttäjien pyynnöt on järkevämpi ohjata oman palveluntarjoajan palvelimelle.

Agentit voivat olla joko ohjaavia, välittäviä, uudelleenohjaavia tai muuntavia. Ohjaavat agentit ohjaavat viestit realm -kentän mukaan oikeaan verkkoon. Välittävät agentit toimivat muuten lähes samoin, mutta ne voivat muuttaa viestin sisältöä. Uudelleenohjaavat agentit palauttavat kohdeosoitteen kysyjälle. Ne eivät osallistu itse viestin välittämiseen, vaan se jää kysyjän vastuulle. Muuntavat agentit suorittavat protokollamuunoksia Diameterin ja muiden AAA-protokollien välillä.

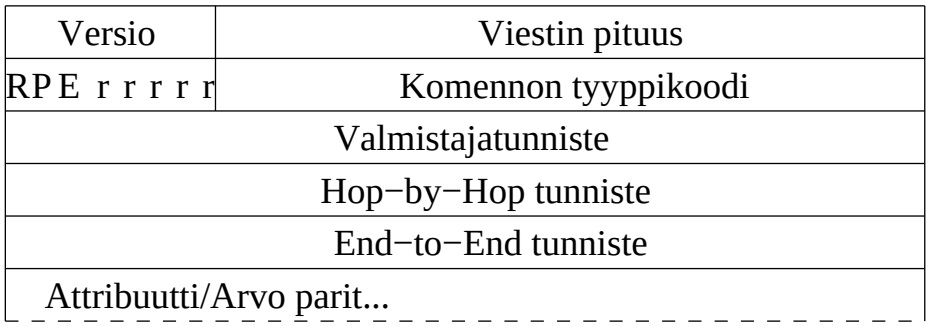


Kuva 2.7: Viestin ohjaus

Kuvassa 2.7 on esitetty ohjaavan ja uudelleenohjaavan agentin toiminta. Asiakkaan halutessa rekisteröityä, viesti lähetetään liityntäpisteelle. Liityntäpiste katsoo parametrina saadusta realm -kentästä, lähetetäänkö viesti paikalliselle palvelimelle vai agentille. Jos realm osoittaa eri verkkoon, lähetetään autentikointipyyntö agentille, kuvassa tämä on esitetty nuolella 1. Viestin saatuaan agentti katsoo omasta realm -kannastaan viestin kohteen, tässä esimerkissä kohde on uudelleenohjaava agentti, jolle viesti lähetetään. Uudelleenohjaava agentti katsoo nyt puolestaan tauluistaan kohteen, mutta koska se on uudelleenohjaava, ei se itse lähetä viestiä eteenpäin vaan palauttaa edellisellä agentille osoitteen, johon viesti täytyy

lähettää. Kuvassa esitetty nuolella 3. Tämän jälkeen ohjaava agentti tietää käyttäjän kotiverkon Diameter-palvelimen osoitteen ja lähettää viestin sille. Kuvassa esitetty nuolella 4. Tämän jälkeen vastaus kulkee suoraan ohjaavan agentin kautta takaisin liityntäpisteelle. Kuvassa esitetty nuolilla 5 ja 6.

Kuten ylläolevasta esimerkistä käy ilmi, ei liityntäpisteen, agenttien ja kotiverkon Diameter-palvelimen välillä ole valmiiksi luotua luottamussuhdetta kuten RADIUS -protokollalla. Lisäksi liikenne saattaa kulkea usean eri verkon kautta, joten liikenne täytyy saada myös salattua. Luottamussuhteen luomiseen on olemassa CMS security-sovellus, jonka avulla osapuolet saavat luotua luottamussuhteen attribuutti/arvo-parien turvaamiseksi. Lisäksi sovelluksessa määritellään keinot, joilla Diameter-asiakas voi pyytää tietyn realm:in Diameter-palvelinta luomaan luottamussuhteen. Luottamussuhde voidaan siis luoda asiakkaan, palvelimen tai agentin pyynnöstä. Liikenteen salaamiseen on määritelty kaksi protokollaa IPsec ja TLS. IPsec:iä on tarkoitus käyttää oman verkon sisällä ja TLS:ää verkkojen välillä.



Kuva 2.8: Diameter-viestin otsikon rakenne

Kuvassa 2.8 on Diameter-viestin otsikon rakenne. Versio-kenttä määrittelee käytetyn version, tällä hetkellä ainoastaan versio 1 on sallittu. Viestin pituus -kentässä on viestin koko pituus, myös otsikkokenttä. R-lipun ollessa 1, kyseessä on pyyntö, muulloin vastaus. P-lippu kertoo voiko viestiä lähettää agenttien kautta. E-lipulla ilmoitetaan virheestä. Loput 5 bittiä on varattu myöhempää käyttöä varten. Komennon tyyppikoodilla määritellään viestin tyyppi. Jokaiselle komennolle on varattu oma numero, ja niitä voi anoa IANA:lta lisää. Valmistajatunnus pitää asettaa jos käytetään jotain valmistajakohtaista ratkaisua, muulloin kentän arvo on 0. Hop-by-Hop tunnistetta käytetään tunnistamaan pyynnöt ja vastaukset hyppyjen välillä ja End-to-End tunnistetta loppupisteiden välillä. Attribuutti/arvo -parit ovat vastaavia kuin RADIUS-protokollan attribuutit, eli tunnistukseen, valtuutukseen ja tilastointiin liittyvät parametrit määritellään niissä.

3 Session Initiation Protocol (SIP)

SIP on rakenteeltaan HTTP:n (Hypertext Transfer Protocol) ja SMTP:n (Simple Mail Transfer Protocol) kaltainen protokolla. Protokollan avulla on mahdollista muodostaa ja hallita laitteiden välisiä istuntoja. Alunperin protokolla on suunniteltu multimediaistuntojen luomiseen, mutta myöhemmin sitä on useissa sovelluksissa laajennettu toisenlaisiinkin istuntoihin. Tässä luvussa käsittelemme SIP-protokollaa reaaliaikaisten multimediayhteyksien luonnissa. Multimediayhteyksillä tarkoitetaan äänen ja kuvan välittämistä verkossa. Reaaliaikaisilla multimediayhteyksillä tarkoitetaan esimerkiksi IP-puheluita jossa puhetta, kuvaa tai molempia välitetään reaaliaikaisesti. SIP-protokolla on alunperin määritelty IETF:n RFC:ssä 2543 [10]. Sitä on myöhemmin laajennettu ja tarkennettu. Uusin IETF:n draft on versio 09 [11].

```
INVITE sip:bob@biloxi.com SIP/2.0
   Via: SIP/2.0/UDP pc33.atlanta.com;branch=z9hG4bK776asdhds
   Max−Forwards: 70
   To: Bob <sip:bob@biloxi.com>
   From: Alice <sip:alice@atlanta.com>;tag=1928301774
   Call−ID: a84b4c76e66710@pc33.atlanta.com
```

Kuva 3.1: SIP-viestin rakenne

Esimerkki SIP-viestin otsikon rakenteesta on esitetty kuvassa 3.1. Jokaisessa viestissä ensimmäisellä rivillä kerrotaan protokolla, versio, sekä viestin tyyppi. Esimerkissä viestin tyyppi on INVITE eli yhteyden avauspyyntö. Lisäksi protokollana käytetään SIP:iä ja ver-

siota 2.0, joka on tällä hetkellä ainoa versio. VIA-kenttään jokainen palvelin merkitsee oman osoitteensa, jotta vastaukset palautuvat samaa reittiä. Max-Forwards parametrillä kerrotaan palvelimien lukumäärä, joiden kautta viesti voidaan välittää. To ja From -kentät ovat merkitykseltään samoja kuin sähköpostissakin. To -kenttä kertoo viestin kohteen, ja From-kenttä viestin lähettäjän. Call-ID-kentällä erotetaan yhteydet. Jokaisessa yhteyden avauksessa täytyy olla eri Call-ID. Lisäksi viestissä voi olla muita vapaaehtoisia alunperin HTTP:ssä määriteltyjä parametrejä. INVITE-viestissä on edellä olevien kenttien lisäksi myös SDP-kentät, joilla käyttäjät neuvottelevat yhteysparametrit. Neuvoteltavia yhteysparametrejä ovat esimerkiksi käytettävä protokolla, porttinumerot ja istunnon tyyppi.

3.1 Komponentit

SIP-protokollassa on kaksi pääkomponenttia, SIP-palvelin ja SIP-asiakaslaite. SIP-protokolla on siis perinteisen asiakas-palvelin arkkitehtuurin mukainen. Kuitenkin asiakaslaite on rakenteeltaan sellainen, että se ei välttämättä tarvitse erillistä ulkopuolista palvelinta. Tämä mahdollistaa suorien yhteyksien muodostamisen, mutta toisaalta rajoittaa laajempaa käyttöä. SIP-palvelin jakaantuu myös toiminnallisuutensa perusteella kolmeen eri kategoriaan. Toiminnallisuuden erot liittyvät palvelimen tapaan käsitellä pyyntöjä.

3.1.1 SIP-palvelin

SIP-palvelin tarjoaa käyttäjille mahdollisuuden muodostaa yhteyksiä riippumatta vastapuolen sijainnista. Yhteyden avauspyynnöt voidaan lähettää SIP-palvelimelle. SIP-palvelin suorittaa kyselyn tietokantaan tai johonkin muuhun palveluun, jonka perusteella se saa tiedon käyttäjän sijainnista. Tiedon saatuaan se lähettää viestin oikeaan kohteeseen. Ohjaustavasta riippuen SIP-palvelimet jaetaan kahteen pääkategoriaan: uudelleenohjaaviin ja välittäviin palvelimiin. Välittävät palvelimet jaetaan vielä kahteen osakategoriaan, tilallisiin ja tilattomiin välittäviin palvelimiin.

Käytettäessä uudelleen ohjaavaa SIP-palvelinta, käyttäjän yhteydenmuodostuspyyntö lähetetään palvelimelle. Palvelin suorittaa käyttäjän etsinnän, mutta se ei välitä yhteyden muodostuspyyntöä eteenpäin, vaan palauttaa käyttäjälle tiedon kohteen sijainnista. Lopullinen

yhteydenmuodostus jää siten asiakkaan tai asiakkaan laitteen tehtäväksi. Saatuaan tiedon, asiakas lähettää yhteydenmuodostuspyynnön suoraan kohteelle, ei SIP-palvelimelle. Tällä tavalla toimiva palvelin sovittaa SIP-protokollan tietokantaprotokolliin, sekä muihin palvelun etsintään tarkoitettuihin protokolliin.

Palvelimen ollessa välittävä, yhteyden avaus suoritetaan vastaavasti kuin uudelleenohjaavassakin palvelimessa. Yhteyden avauspyyntö lähetetään SIP-palvelimelle, ja SIP-palvelin suorittaa käyttäjän etsinnän. Sen sijaan palvelin ei lähetä käyttäjälle osoitetta takaisin, vaan lähettää avauspyynnön kohteelle. Tässä vaiheessa on tilallisen ja tilattoman palvelimen välillä eroja. Tilaton palvelin ainoastaan välittää pyyntöjä. Se ei jää tarkkailemaan yhteydenmuodostuksen kulkua, eikä ole vastuussa uudelleenlähetyksistä. Tällöin uudelleenlähetykset ja muu tarkkailu jää käyttäjien laitteiden vastuulle. Tilaton palvelin ei myöskään tue muita kuin kahden osapuolen välisiä yhteydenmuodostuksia, koska useamman osapuolen välisissä yhteyksissä alkuperäisellä lähettäjällä ei ole mahdollisuutta tietää kaikkia kohteita.

Tilalliset välittävät palvelimet ovat rakenteeltaan monimutkaisempia. Ne myös mahdollistavat enemmän toimintoja kuin muut edellä kuvatut ratkaisut. Tilalliset välittävät palvelimet mahdollistavat esimerkiksi useamman osapuolen yhteydet. Tilallisia palvelimia käytettäessä avauspyyntö lähetetään edelleen palvelimelle, ja palvelin etsii kohteen yhteystiedot. Saatuaan tiedot, palvelin lähettää pyynnön kohteelle. Palvelimella on tilakone, joka seuraa yhteydenmuodostuksen tilaa. Palvelin suorittaa tarvittaessa uudelleenlähetykset. Tämä yksinkertaistaa asiakaslaitteiden toiminnallisuutta, sekä sallii yhteydenmuodostukset useammalle kohteelle. Jos kohteita on useampi kuin yksi selvittää laite kaikkien yhteystiedot. Tämän jälkeen palvelin lähettää avauspyynnöt kaikille kohteille, ja tekee jokaiselle oman tila-automaattinsa. Käyttäjälle toiminnallisuus ei näy muuten kuin, että avauksen vastausviestejä tulee useammalta kuin yhdeltä laitteelta.

Edellä kuvatuissa tapauksissa kohde löytyi aina saman palvelimen takaa, kuin missä lähettäjäkin oli. Se ei kuitenkaan ole vaatimus, vaan kohde voi olla yhden tai useamman palvelimen takana. Tällöin toiminnallisuus on muuten samanlainen, paitsi SIP-palvelimet lähettävät avauspyynnöt toiselle palvelimelle, kunnes päästään oikealle palvelimelle. Oikean SIP-palvelimen osoite löytyy käyttäjän osoitteen verkko-osan perusteella. Koska oletusarvoisesti jokaiselle signaloinnin viestille etsitään reitti uudelleen, saattaa se aiheuttaa ongelmia erinäisten verkkokomponenttien suhteen. Tämän vuoksi ensimmäisessä avausviestissä

voidaan viestiin liittää tieto palvelimista joiden läpi se on mennyt, ja käyttää tätä tietoa myöhemmissä viesteissä saman reitin pitämiseksi.

3.1.2 SIP-päätelaite

Päätelaite jakaantuu kahteen osaan, päätelaitteen asiakkaaseen ja palvelimeen. Asiakasosuus huolehtii yhteyden avaus- ja hallintaviestien lähettämisestä. Palvelinosuus toimii samaan tapaan kuin SIP-palvelin, lukuunottamatta viestien välittämistä tai uudelleenohjaamista. Kun asiakas haluaa muodostaa yhteyden, päätelaitteen asiakas muodostaa yhteyden avauspyynnön, sekä lähettää sen eteenpäin. Tämän jälkeen se jää odottamaan vastausta. Kun yhteyden avauspyyntö pääsee kohteelle, sen palvelinosa ottaa viestin vastaan ja prosessoi sen. Ensimmäiseksi se katsoo, onko kyseistä käyttäjää olemassa sillä palvelimella. Jos kyseinen käyttäjä löytyy, tarkistaa se halutaanko käyttäjäntunnistusta. Jos käyttäjä haluaa, että vastapuoli tunnistautuu, päätelaitteen palvelin lähettää autentikoitumispyynnön. Kun yhteyden avauspyynnön lähettänyt asiakasosuus saa autentikoitumispyynnöön, lähettää se uuden viestin tarvittavilla tiedoilla. Kun mahdollinen autentikointi on suoritettu, palvelin ilmoittaa käyttäjälle, että käyttäjään yritetään muodostaa yhteyttä. Käyttäjä päättää hyväksytäänkö vai hylätäänkö yhteyden muodostus. Jos käyttäjä hyväksyy yhteyden avauspyynnön, lähettää palvelin tiedon yhteyden hyväksymisestä vastapuolelle. Tällainen toiminnallisuus mahdollistaa kahden pisteen väliset suorat yhteydet, eikä välissä olevaa palvelinta välttämättä tarvita.

3.2 Autentikointi

Koska yhteyden muodostus tapahtuu IP-protokollan päällä, täytyy tietoturvanäkökohtiin kiinnittää suurempaa huomiota kuin perinteisestä puhelinverkossa. Tämän vuoksi käyttäjän autentikointi on yksi tärkeä osuus protokollassa. Koska SIP on muutenkin rakenteeltaan HTTP:n kaltainen, käytetään käyttäjän autentikoinnissakin HTTP-perustaista Digest metodia. Kuitenkaan metodia ei kiinteästi sidottu, vaan metodeita voidaan myöhemmin lisätä. Alkuperäisessä SIP:n määrittelyssä [10] oli mukana PGP:llä tapahtuva autentikointi, mutta nykyisessä versiossa [11] se on poistettu. Toinen nykyisestä määrittelystä poistettu autentikointiprotokolla on selkokielisiin salasanoihin perustuva HTTP Basic. Poistettujen au-

tentikointitapojen tilalle uuteen määrittelyyn on otettu mukaan IPSec ja TLS. Ne tarjoavat luotettavan autentikoinnin ja liikenteen salauksen. Kuitenkin näissä metodeissa täytyy olla avaimet jaettuna etukäteen tai erillinen palvelin kyseiseen tarkoitukseen. Tämä vaikeuttaa huomattavasti niiden käyttöä.

3.2.1 HTTP Digest

HTTP Digest on perinteinen haaste/vastaus -autentikointimenetelmä. Salasanaa ei koskaan kuljeteta linkillä, vaan salasanasta lasketaan satunnaisluvulla tarkistussumma molemmissa päissä. Salasanan oikeellisuus varmistetaan näillä tarkistussummilla. Käytännössä autentikointi tapahtuu siten, että autentikointia haluava osapuoli generoi satunnaisluvun, jonka se lähettää haasteena toiselle osapuolelle. Haasteen avulla autentikoituva osapuoli laskee tarkistussumman käyttäjän salasanasta, ja lähettää sen vastauksena autentikointia halunneelle osapuolelle. Vastauksen saatuaan autentikointia halunnut osapuoli laskee samalla satunnaisluvulla omassa kannassaan olevasta salasanasta tarkistusumman ja vertailee sitä vastauksessa saatuun.

HTTP Digestissä haaste annetaan joko 401 Unauthorized tai 407 Proxy Authentication Required -viesteillä. 401-viesti lähetetään siinä tapauksessa, että käyttäjä haluaa yhteyttä kyseiseen laitteeseen. 407-viestin tapauksessa käyttäjä on muodostamassa yhteyttä välityspalvelimen kautta, mutta ennen viestin välitystä välityspalvelin haluaa käyttäjän autentikoituvan. 401-viestissä haaste on WWW-Authenticate otsikkokentässä. WWW-Authenticate otsikkokenttä sisältää satunnaisluvun nonce-parametrissä. Lisäksi otsikon parametrinä täytyy aina olla realm, joka kertoo käyttäjälle vähintään koneen nimen johon hän on yrittämässä. Määrittelyssä on mainittu muitakin valinnaisia parametrejä, joita ovat esimerkiksi opaque, stale, algorithm ja qop. Opaque on palvelimen määrittelemä merkkijono, jonka tunnistautuvan osapuolen täytyy palauttaa sellaisenaan takaisin. Tämän parametrin avulla on mahdollista tunnistaa samaan istuntoon kuuluvat viestit. Istunto voidaan sitoa myös nonce-parametriin. Stale-parametrin avulla palvelin voi kertoa, että edellinen salasana oli oikein, mutta nonce oli väärä. Tällöin ei tarvitse pyytää käyttäjältä salasanaa uudelleen, vaan pelkästään lähettää oikealla nonce-arvolla uusi viesti. Algorithm-parametrillä kerrotaan algoritmi, jota tarkistussumman laskemisessa käytetään. Qop-parametri on aikaisemmassa määrittelyssä ollut pakollinen, mutta nykyisessä [17] se ei enää ole. Kyseisen para-

metrin avulla on mahdollista ilmoittaa suojauksen laatu. Arvolla auth on kyseessä vain tunnistautuminen, mutta arvolla auth-int mukana on myös sisällön suojaaminen. Auth-int:iä käytettäessä otsikon arvoista lasketaan tarkistussumma, jolloin käyttäjä voi olla varma että tieto on alkuperäisessä muodossa. 407-viestissä otsikkokentän nimi on Proxy-Authenticate, mutta parametrit ovat samoja kuin 401-tapauksessakin.

Saatuaan 401- tai 407-viestin laite lähettää saman pyynnön uudelleen, mutta lisää siihen Authorization-otsikkokentän. Otsikkokentässä on vähintään käyttäjätunnus, realm-, nonce- ja response-parametrit. Realm- ja nonce-parametrit ovat samat kuin autentikoitumispyynnössä. Response-parametrissa on käyttäjätunnuksen, noncen ja salasanan avulla laskettu tarkistussumma. Tätä tarkistussummaa vertailemalla autentikoitumista vaativa osapuoli varmistaa salasanan oikeellisuuden. Jos alkuperäisessä pyynnössä oli opaque -parametri, täytyy Authorization -otsikkokentässäkin se olla. Lisäksi sen arvon täytyy olla sama kuin pyynnössä. Jos pyynnössä oli qop-parametri, täytyy vastauksessa olla parametrit qop, cnonce ja nc. Qop kertoo käytetyn metodin (auth tai auth-int). metodia. Nc-parametri kertoo saatujen noncejen lukumäärän. cnonce on vastaava kuin opaque, mutta tunnistautuva osapuoli muodostaa sen. Kun käytössä on sekä opaque että cnonce, molemmat osapuolet voivat varmistua toisistaan.

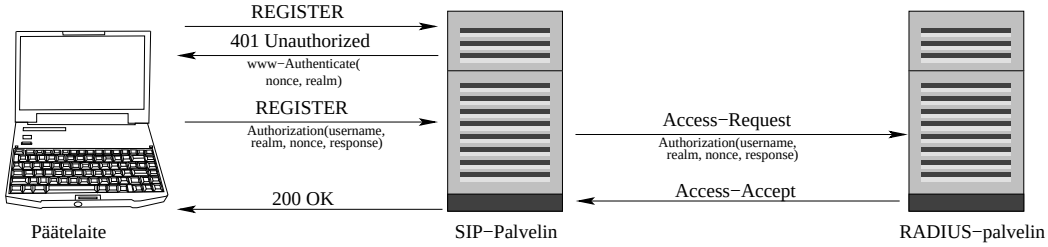
HTTP Digest ei tarjoa minkäänlaista suojaa itse liikenteelle, eikä suojaus ole enää nykyisellä laskentateholla vahva. Kuitenkin se on parempi kuin pelkkä selkokielisten salasanojen lähettäminen. Lisäksi se on melko helppo toteuttaa, eikä se vaadi erillistä avaintenvaihtoa. Sovelluksen vaatiessa hyvää salausta ja sisällön oikeellisuuden varmistamista, ei HTTP Digest ole oikea ratkaisu, mutta pelkkään käyttäjän autentikointiin se on riittävä.

3.2.2 SIP ja RADIUS

Vaikka SIP:issä käytettävä HTTP Digest on samaan tapaan haaste-vastaus-mallilla toimiva protokolla kuten RADIUS:kin, ei se suoraan sovellu käytettäväksi. Käytettävät algoritmit ja tarvittavat parametrit ovat erilaisia. Kuitenkaan protokollat eivät eroa merkittävissä määrin, joten RADIUS-protokollaan on mahdollista saada HTTP Digest tuki. RADIUS-protokollan HTTP Digest on vielä varhaisessa määrittelyvaiheessa [16], mutta siitä on jo muutamilla valmistajilla toteutuksia. Käytännössä tuen saaminen vaatii muutaman uu-

den attribuutin määrittelemisen, joiden avulla Authentication- otsikossa olevat parametrit saadaan kuljetettua palvelimelle. Lisäksi HTTP Digestin perinteisestä asiakas-palvelin -autentikointimenetelmästä täytyy luopua. Tässä mallissa SIP-palvelin toimii liityntäpisteen roolissa. Se tekee muunnoksen HTTP Digestin ja AAA -protokollan välillä. Käytännössä se laittaa HTTP Digestin parametrit AAA -protokollan parametriksi, ja vastauksissa se tekee muunnoksen toisin päin. Tällainen malli ei vaadi käyttäjän laitteeseen mitään muutoksia. Aikaisemmin oli myös määrittely, joka olisi tehnyt SIP-asiakaslaitteista RADIUS-kelpoisia. Tästä on kuitenkin tällä hetkellä luovuttu.

Tarvittavan RADIUS-attribuutin täytyy pystyä kuljettamaan HTTP Digest-parametrit RADIUS-palvelimelle. RADIUS:ta käytettäessä asiakkaan toiminta ei eroa mitenkään normaalista, ilman RADIUS:ta tapahtuvasta toiminnasta. Käyttäjä lähettää ensin pyynnön laitteelle jolle haluaa yhteyden. Kohde vastaa kyselyyn joko 401 tai 407 -viestillä. Käyttäjän laite muodostaa uuden pyynnön, jossa on autentikaatio-otsikkokenttä ja lähettää sen kohteelle. Sen sijaan, että kohde katsoisi omasta tietokannastaan käyttäjätunnuksen muodostaa se RADIUS-viestin. Tähän Access-Request viestiin se laittaa saamansa tunnistusotsikon parametrit Digest-attribuutteina ja käyttäjätunnuksen User-Name attribuuttiin. RADIUS-palvelin laskee tietokannassaan olevasta salasanasta HTTP Digestissä käytettävällä algoritmillä tarkistussumman ja vertaa sitä saatuun vastaukseen. Jos tarkistussumma oli oikein, lähettää palvelin Access-Accept -viestin. Kun tunnistautumista pyytänyt laite saa kyseisen viestin, katsoo se tunnistautumisen onnistuneeksi ja normaali liikennöinti voi alkaa. Kuvassa 3.2 on esitetty onnistunut rekisteröityminen RADIUS-autentikoinnilla.

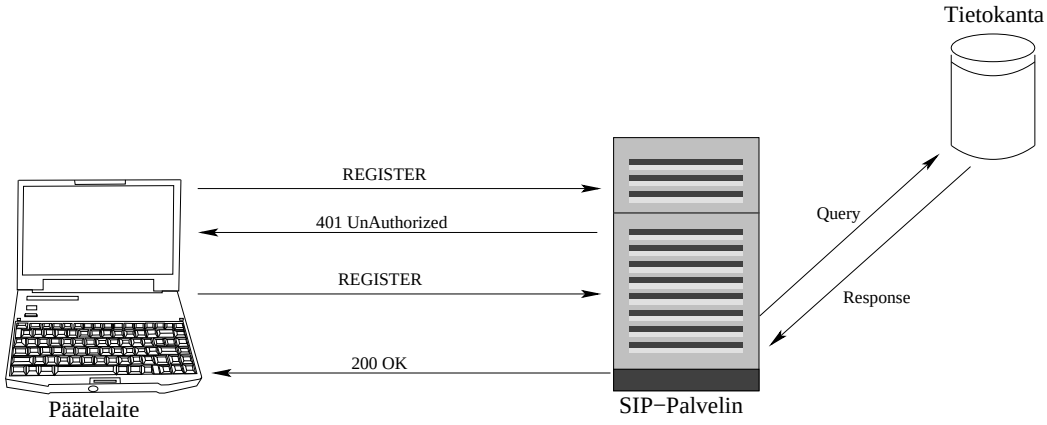


Kuva 3.2: Onnistunut rekisteröintyminen RADIUS-autentikoinnilla

3.3 Komponenttien yhteistoiminta

Tässä alaluvussa on käsitelty edellisissä kohdissa olleiden komponenttien välinen yhteistoiminta sekä käyttäjän tunnistus yhdistettynä SIP-viestinvälitykseen. Käyttäjän autentikointia tarvitaan kahdessa vaiheessa. Rekisteröinnin yhteydessä tarkastetaan, että käyttäjällä on oikeus rekisteröidä tunnus, sekä salasanan oikeellisuus. Yhteyden muodostuksessa tarkistetaan salasanan oikeellisuuden lisäksi, että käyttäjällä on oikeus lähettää viestejä haluamansa käyttäjän nimissä.

3.3.1 Rekisteröinti



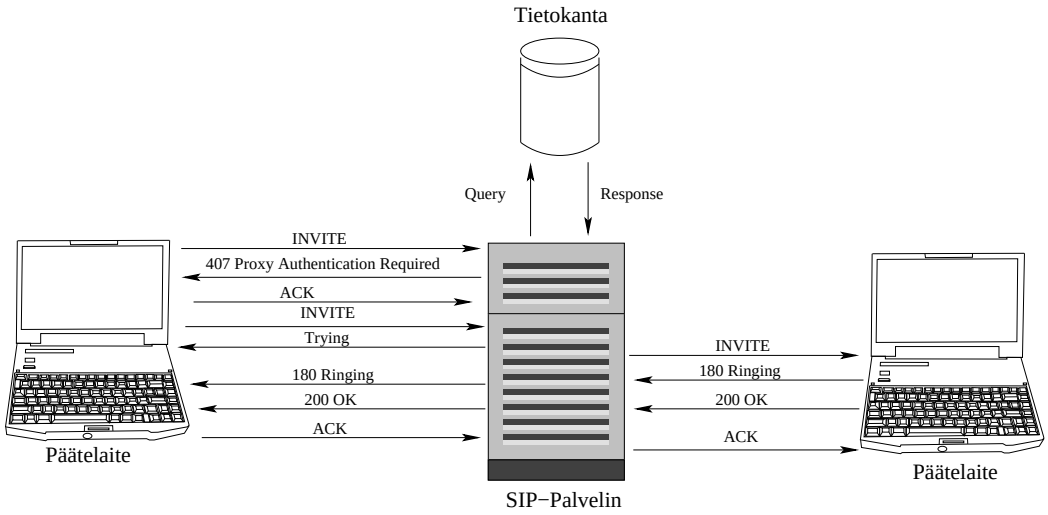
Kuva 3.3: Onnistunut rekisteröinti

Kuvassa 3.3 on esitetty onnistunut rekisteröintitapahtuma. Kuvassa esitetyssä tapauksessa on käytetty käyttäjän autentikointia. Aluksi käyttäjä lähettää REGISTER-viestin, jossa hän ilmoittaa palvelimelle SIP-osoitteensa. SIP-osoitteen lisäksi viestissä on IP-osoite ja portti, josta käyttäjä on tavoitettavissa. Koska käytössä on autentikointi, ei palvelin hyväksy suoraan rekisteröitymispyyntöä vaan vaatii käyttäjältä käyttäjätunnuksen ja salasanan. HTTP Digestissä ei salasanaa lähetetä selkokielisenä, minkä vuoksi palvelin antaa haasteen 401-viestissään. Viestin saatuaan päätelaite laskee MD5-tarkistussumman annetulla haasteella, ja liittää sen uuteen REGISTER-viestiin. Uuden REGISTER-viestin saatuaan palvelin pyytää käyttäjän salasanaa tietokannalta, minkä jälkeen se laskee MD5-summan samalla satunnaisluvulla. Jos lasketut tarkistussummat täsmäävät, ja käyttäjällä on oikeus kyseisen tunnuksen rekisteröintiin, katsotaan rekisteröityminen onnistuneeksi ja lisätään käyttäjän rekisteröitymistiedot palvelimen sijaintitietokantaan. Lopuksi palvelin lähettää käyttäjälle

tiedon onnistuneesta rekisteröitymisestä OK-viestillä.

Vaikka tässä esimerkissä rekisteröinti suoritettiin SIP-palvelimelle, se ei ole välttämättömyys. SIP-palvelussa voidaan käyttää myös muita sijaintietopalveluita. Toinen vaihtoehto rekisteröitymiselle olisi ollut rekisteröinnin suorittaminen erilliselle rekisteröintipalvelimelle. Rekisteröintipalvelin lisää käyttäjän yhteystiedot erilliselle sijaintipalvelimelle, jota myös SIP-palvelin käyttää viestien ohjaamiseen. Tälläinen toiminnallisuus mahdollistaa käyttäjien sijaintitietokantojen yhdistämisen.

3.3.2 Yhteyden muodostus



Kuva 3.4: Onnistunut yhteyden muodostus

Kuvassa 3.4 on esitetty onnistunut yhteyden muodostus. Kuten rekisteröinnin yhteydessä, on tässäkin esimerkissä käytössä käyttäjän autentikointi. Yhteydenmuodostuksen aluksi päätelaite lähettää INVITE-viestin, jossa To-kentässä on kohteen osoite. From-kentässä on osoite, jota käyttäjä haluaa käyttää yhteydenmuodostuksessa. Se voi olla joko käyttäjän oma osoite tai kolmannen osapuolen tapauksessa jonkun muun osoite. Koska käytössä on käyttäjän tunnistus, lähettää palvelin 407-viestin. Erona rekisteröintiin on se, että INVITE -viestiä ei ole suunnattu palvelimelle, joten se ei voi vastata 401-viestillä. Rakenteeltaan 407-viesti on samanlainen kuin 401:kin, eli siinä on käyttäjän salasanalle haaste. Käyttäjän laite kuittaa saadun autentikoitumispyynnön ACK-viestillä, jotta SIP-palvelin tietää käyttäjän saaneen viestin ja mahdolliset uudelleenlähetysajastimet voidaan poistaa. Tämän jälkeen käyttäjä lähettää uuden INVITE-viestin, jossa on autentikaatiokentät mukana. Saa-

tuaan viestin palvelin tarkistaa, että salasanan ja käyttäjätunnus on oikein. Lisäksi palvelin tarkistaa, että käyttäjällä on oikeus käyttää From-kentän osoitetta yhteyden muodostukseen.

Tässä esimerkissä on käytössä välittävä SIP-palvelin, joten se ohjaa autentikoitumisen jälkeen INVITE -viestin oikealle kohteelle. Kohteen osoitteen se saa joko omasta sijaintitietokannasta tai erilliseltä sijaintitietopalvelimelta. Löydettyään kohteen, lähettää se INVITE-viestin eteenpäin. Mahdollisesti se myös kertoo siitä avauspyynnön lähettäneelle käyttäjälle Trying-viestillä. Kun kohdelaite saa viestin, se lähettää Ringing-viestin. Tämäkään viesti ei ole pakollinen, mutta sen avulla yhteyden avaajaa voidaan informoida kohteen löytymisestä. Kun kohde hyväksyy yhteyden, lähettää hänen laitteensa OK-viestin. SIP-palvelin välittää OK-viestin yhteyden aloittajalle. Jos yhteysparametrit hyväksytään molemmissa päissä, lähettää päätelaite vielä ACK-viestin. Tällä viestillä molemmat varmistuvat yhteysparametrien oikeellisuudesta. Lisäksi viestin jälkeen tilallinen SIP-palvelin tietää yhteyden muodostuksen onnistuneen, ja se voi poistaa tila-automaattinsa.

Jos käytössä olisi ollut uudelleenohjaava palvelin, olisi INVITE-viestin ohjaamisen sijasta palvelin palauttanut Redirect- viestin. Viestissä olisi ollut kohteen sijaintitedot. Viestin jälkeen kaikki loputkin viestit olisi lähetetty suoraan päätelaitteiden välillä. Muuten yhteyden muodostus olisi tapahtunut samalla tavalla.

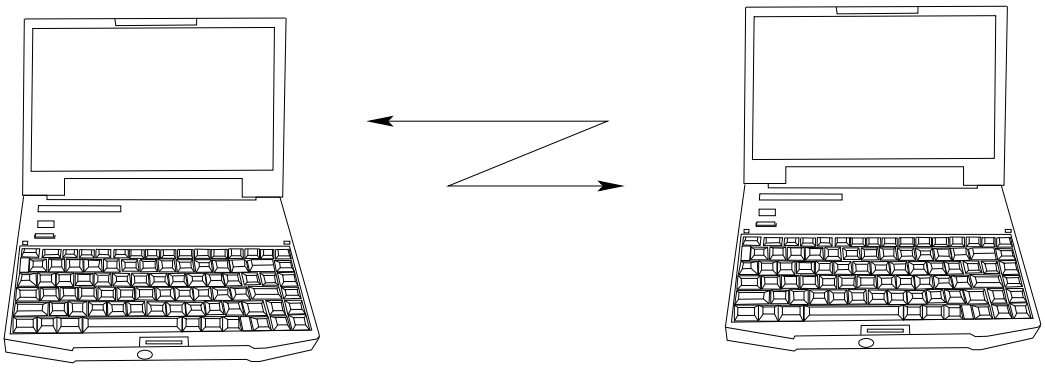
4 Langattomat lähiverkot (WLAN)

Langattomat lähiverkot ovat yksi tapa muodostaa lähiverkkoja. Perinteisen väylätyyppisen lähiverkon kanssa toiminnallisuus on lähes saman tapainen, ainoastaan siirtomedia ja siinä käytetyt metodit, kuten törmäysten havaitseminen, eroavat. Nykyisiin tähtikytkentäisiin parikaapelilla tai kuidulla toteutettuun lähiverkkoon verrattaessa toiminta eroaa enemmän, etenkin tietoturvan osalta. Kytkentäisessä verkossa muiden liikenteen kaappaamiseen vaaditaan pääsy fyysiseen siirtotiehen, eli laitteen ja kytkimen väliselle kaapelille. Yhteyttä voidaan siten pitää pisteestä pisteeseen yhteytenä. Langattomassa verkossa liikenne kulkee radioaalloilla, joten kuka tahansa pystyy kaappaamaan liikennettä. Lisäksi laitteen kytkeminen verkkoon ei vaadi fyysisen yhteyden kytkemistä, jolloin pääsynvalvonnalle tulee entistä suurempi rooli.

4.1 Perustoiminnallisuus

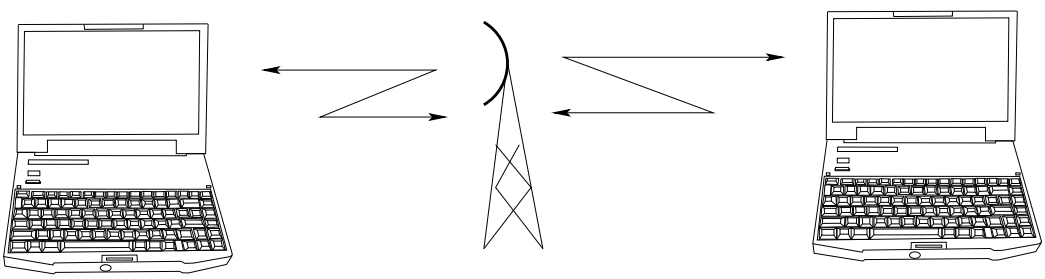
Langattomat verkot voivat toimia kahdessa eri moodissa. Toiminta Ad-Hoc moodissa on esitetty kuvassa 4.1. Ad-Hoc -moodissa toimivassa WLAN:ssa laitteet liikennöivät suoraan toisilleen. Tämä moodi on helpoin toteuttaa, koska se ei vaadi erillisiä komponentteja verkkoon. Kuitenkin Ad-Hoc-moodissa olevassa verkossa on vaatimuksena, että laitteet pystyvät muodostamaan radioyhteyden. Periaatteessa Ad-Hoc verkot rajoittavat liikennöinnin vain samalla alueella olevien laitteiden välille, koska kaikki laitteet ovat yhtäläisiä. Tämän vuoksi esimerkiksi muunnos lankaverkon ja langattoman välillä ei onnistu. Käytännössä tämä rajoitus riippuu laitteiden konfiguroinnista. Joissakin tapauksissa riittää, että laite saa

langattoman yhteyden toiseen laitteeseen. Tällöin Ad-Hoc on varteen otettava vaihtoehto helppoutensa vuoksi.



Kuva 4.1: Ad-Hoc -moodissa toimiva WLAN

Toinen toimintamoodi on Infrastructure. Infrasctructure-moodissa olevan WLAN:in toiminsta on esitetty kuvassa 4.2. Infrastructure-moodi vaatii erillisen tukiaseman, jonka kautta yhteydet muodostuvat. Toiminta tässä moodissa on hieman monimutkaisempi kuin Ad-Hoc -moodissa. Kuitenkin monipuolisuutensa ansiosta se on yleisimmässä käytössä. Infrastructure-moodissa jokainen tukiasema muodostaa palvelualueen, jonka alueella se välittää liikennettä. Näitä palvelualueita voidaan yhdistää siten, että useammalla tukiasemalla on sama laajennettu palvelualue. Laajennettu palvelualue tarkoittaa sitä, että useampi tukiasema muodostaa loogisen suuremman palvelualueen, joka näkyy käyttäjällä yhtenä verkkona. Näin käyttäjän saama palvelualue suurenee, sekä liikkuminen helpottuu. Koska Infrastructure-moodissa kaikki liikenne kulkee tukiaseman kautta, sen avulla voi yhdistää WLAN:eja toisiinsa tai yhdistää WLAN:in ja perinteisen langallisen LAN:in. Tukiasemat voivat keskustella toistensa kanssa myös langattomasti. Langattomassa tapauksessa tosin nopeus putoaa jokaisella hypyllä, koska liikenne välitetään normaalisti kuuluvuusalueelle, josta tukiasema kuulee liikenteen ja välittää sen omalle kuuluvuusalueelleen. Yleisin tapaus on liittää langaton tukiasema lähiverkkoon ja laajentaa näin lähiverkkoa.



Kuva 4.2: Infrastructure -moodissa toimiva WLAN

Vaikka langattomat lähiverkot toimivat lähes samaan tapaan kuin langallisetkin, aiheuttaa radiosiirtotie joitain ongelmia. Siirtotie on merkittävästä epävarmempi kuin langallisissa lähiverkoissa. Tämän vuoksi ei voida olettaa, että kaikki siirretty data menee varmasti läpi. Toisaalta sellaista oletusta ei voi tehdä langallisissakaan verkoissa, mutta niissä virhesuhteet ovat huomattavasti pienempiä. Lisäksi mahdolliset häiriösignaalit voivat vaikuttaa liikenteen kulkuun. Langattomissa lähiverkoissa ei ole keinoa estää häiriösignaaleja. Häiriösignaalit voivat tulla muista langattomiin lähiverkkoihin liittymättömistä laitteista, tai sitten tukiasemat voivat häiritä toisiaan. Suurin ero langallisiin verkkoihin verrattuna on se, että langatonta verkkoa ei voi pitää täysin kytkettynä. Kaikki laitteet eivät välttämättä kuule toisiaan, ja se saattaa aiheuttaa törmäyksiä siirtotiellä. Tämän vuoksi laitteen, jolle useampi laite liikennöi täytyy pystyä kertomaan, että joku toinen on jo siihen tai sen kautta liikennöimässä. Liikennöinnin loputtua laite antaa luvan seuraavalle laitteelle liikennöidä.

4.2 Autentikointi

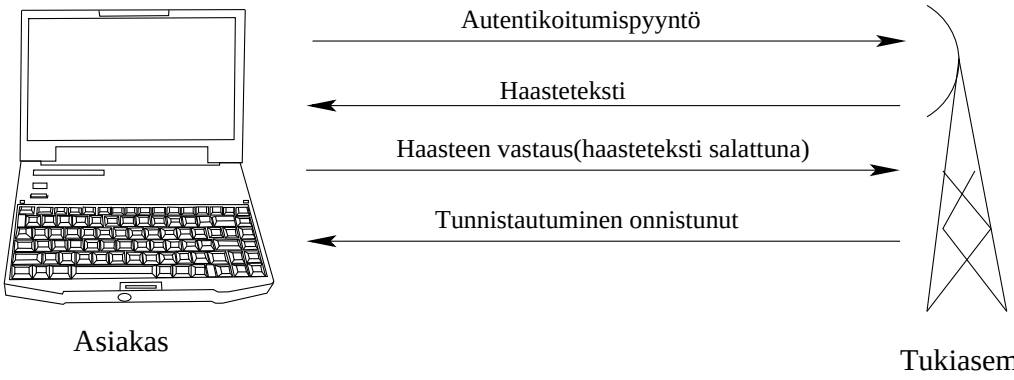
Autentikointi on merkittävässä osassa langattomissa lähiverkoissa. Alkuperäinen langattomien lähiverkkojen määrittely esitteli autentikointiprotokollan WEP. Kuitenkin nykyisin se on osoittautunut heikoksi, sekä salasanojen jakaminen muodostui ongelmaksi. Useissa paikoissa WEP:iä käytetään jaetulla avaimelle, eli kaikilla käyttäjillä on sama salasana, jolloin sen salaisuus on kyseenalainen. IEEE on määritellyt 802.1x-protokollan 802-lähiverkoissa käytettäväksi. 802.1x:n suurin etu siinä käytettävä EAP-protokolla, joka tukee RADIUS:ta. Tällöin WLAN -palvelu ei tarvitse erillisiä käyttäjätunnuksia ja niiden hallinta helpottuu. 802.1x:ää voidaan käyttä myös langallisten lähiverkkojen kanssa, jolloin niidenkin autentikointi voidaan suorittaa samalla protokollalla. Kuitenkin 802.1x on varsin uusi määrittely, joten sille ei ole vielä kaikissa laitteissa tukea. 802.1x:n määrittelyn uutuudesta johtuen nykyisissä WLAN-laitteissa on paljon valmistajakohtaisia ratkaisuja, jotka eivät ole keskenään yhteensopivia.

4.2.1 Wired Equivalency Privacy (WEP) ja autentikointi

IEEE:n langattomien lähiverkkojen määrittely sisältää kaksi autentikointimenetelmää. Avoimen autentikoinnin ja jaetun avaimen autentikoinnin. Avoin autentikointi ei itse asiassa

autentikoi käyttäjää lainkaan, mutta se on mainittu parametriksi, jotta käyttäjän liittyminen WLAN-palveluun tapahtuisi aina samalla kaavalla. Jaetun avaimen autentikoinnissa käytetään hyväksi WEP-mekanismia. WEP käyttää hyväkseen RSA:n RC4 algoritmia ja lisää siihen keinot, jolla molemmat osapuolet saavat saman syötteen algoritmille, jotta salauksen purkaminen olisi mahdollista. WEP:n avulla on myös mahdollista salata koko liikenne, jonka erikoistapauksena autentikointia voidaan pitää. WEP-algoritmin toiminta on esitetty tarkemmin liitteessä B. Alkuperäinen idea WEP-mekanismin takana oli taata turvattomalla radiotielle langallisen verkon tasoinen tietoturva.

Kuvassa 4.3 on esitetty onnistunut jaetun avaimen autentikoituminen. Aluksi päätelaite ilmoittaa halustaan tunnistautua. Jos tukiasema tukee jaetun avaimen tunnistautumista, lähettää se haastetekstin asiakaslaitteelle. Haasteteksti generoidaan yleensä WEP:n satunnaislukugeneraattorilla, mutta se ei ole välttämättömyys. Kuitenkin haasteteksti tulisi olla aina ennustamaton. Tästä johtuen ei ole suositeltavaa käyttää aina samaa haastetta tai käyttää uuden haasteen generointiin vanhaa haastetta. Haasteen saatuaan asiakaslaite kopioi saaduun haasteen viestiinsä ja suorittaa sille WEP:n avulla salauksen. Tämä salattu viesti lähetetään tukiasemalle. Tukiasema tarkistaa ensin, että viestin tarkistussumma täsmää ja sen jälkeen se purkaa saadun viestin. Puretusta viestistä se tarkistaa haastetekstikentän, ja jos sen arvo on sama kuin alkuperäisesti lähetetty, katsotaan autentikoituminen onnistuneeksi.



Kuva 4.3: Onnistunut WEP autentikoituminen

Jaetun avaimen salaus ja autentikointi ei ole hyvä, koska saman salaisuuden jakaa useampi kuin kaksi osapuolta. Kaikki jotka tietävät salasanan pystyvät purkamaan liikenteen selkokieliseen muotoon. Lisäksi kun salasana on usealla henkilöllä tiedossa, suurenee riski sen vuotamiseen. Myös WEP:iä käytetään määrittelyssä siten, että salasana on mahdollista selvittää, ja siten purkaa liikenteen salaus. Tämä tosin vaatii, että hyökkääjä pystyy kuun-

telemaan liikennettä pidemmän aikaa kuin yhden paketillisen. Ongelma WEP:n käytössä on pieni alustusvektorin pituus. Alustusvektori on vain 24 bittiä pitkä, joten se pyörähtää usein ympäri. Tällöin hyökkääjä saa useamman samalla avaimella salatun viestin, jolloin salasanan selville saaminen helpottuu. Mitä enemmän samalla avaimella salattuja viestejä saa, sitä helpompi on saada salasana selville. Lisäksi koska kyseessä on jaettu salaisuus kaikki käyttävät samaa salasanaa. Tällöin tulee tilanne, jolloin useamman kuin yhden käyttäjän liikenne salataan samalla alustusvektorilla ja salasanalla. Tällöin mahdollisen liikenteen kaappaajan työ helpottuu, kun ei tarvitse vain odottaa yhden päätelaitteen käyttävän samaa alustusvektoria, vaan voi etsiä kaikkien päätelaitteiden liikenteestä niitä. Näiden syiden vuoksi WEP:iä ei pidetä turvallisena, ja niinpä onkin lähdetty kehittämään parempia tapoja.

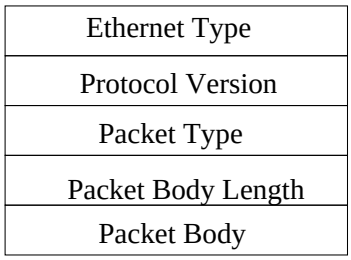
4.2.2 IEEE 802.1x

Lähiverkkojen yleistyessä autentikoinnin tarve kasvaa huomattavasti. Koska ihmiset ovat tottuneet käyttämään verkkoyhteyttä niin kodeissaan kuin töissään, halutaan sitä käyttää myös julkisissa paikoissa, kuten lentokentillä. Lisäksi yrityksissä on julkisissakin tiloissa usein laitteita, jotka ovat kytkettynä yrityksen verkkoon. Tällöin riski yrityksen verkkoon tunkeutumiselle kasvaa. Lisäksi käytettäessä radiotaajuuksia, ei voida niin selkeästi rajata verkon käyttöoikeutta. 802.1x-konsepti mahdollistaa eri lähiverkkotekniikoilla käyttäjän porttikohtaisen autentikoinnin. Tämä malli istuu suoraan nykyisiin kytkettyihin Ethernet-lähiverkkoihin, mutta myös muun tyyppisiin lähiverkkoratkaisuihin. Esimerkiksi langattomissa lähiverkoissa tukiasema muodostaa jokaista käyttäjää kohden virtuaaliportin, jonka avulla käyttäjän liikennöinti voidaan sallia tai estää.

Porttipohjaista autentikointia käytettäessä laitteet on kytketty verkkolaitteeseen, joka ei päästä liikennettä käyttäjälle ennenkuin käyttäjä on tunnistautunut laitteelle. Tälläinen verkkolaite voi olla kytkin, silta tai langattoman verkon tukiasema. Koska nämä verkkolaitteet ovat IP-kerroksen alapuolella, eli ne eivät ohjaa liikennettä IP-osoitteen perusteella, täytyy autentikoituminen tehdä alemman tason protokollalla. Nykyisin yleisimmin käytössä oleva protokolla on Ethernet. Vasta kun autentikointi on suoritettu onnistuneesti, laite voi saada esimerkiksi DHCP:llä IP-osoitteen. 802.1x käyttää jo aikaisemmin esiteltyä EAP-protokollaa autentikointiprotokollanaan ja oikeastaan se vain määrittelee keinot, joilla EAP

saadaan kuljetettua lähiverkkoprotokollien päällä. Tämä paketointitekniikka on nimeltään EAPOL, joka on lyhenne englanninkielisistä sanoista EAP over LANs.

EAPOL määrittelee uuden ethernetin tyyppikentän, sekä sitä seuraavat kentät. Näissä kentissä kuljetetaan EAP:n tarvitsemat parametrit ja niiden arvot. EAPOL:ia käytettäessä Ethernet-kehyksen tyyppikentästä alkaen olevat kentät on esitetty kuvassa 4.4. EAPOL:n ethernetin tyyppikentän arvo on 88-8E. Protocol Version-kenttä määrittelee käytettävän version, tällä hetkellä on ainoastaan määritelty arvo 1. Packet type -kenttä määrittelee viestin tyyppin. Mahdollisia viestityyppejä ovat EAP-packet, EAPOL-Start, EAPOL-Logoff, EAPOL-Key ja EAPOL-Encapsulated-ASF-Alert. Viestityyppi EAP-packet määrittelee, että viesti sisältää EAP-viestin. EAPOL-Start käynnistää tunnistautumisprosessin ja EAPOL-Logoff lopettaa istunnon, jonka jälkeen portti lopettaa liikenteen välittämisen. EAPOL-Key:n avulla pystytään välittämään käytettäviä salaus- ja allekirjoitusavaimia. Viimeksi mainitun viestityypin avulla pystytään tarkkailemaan epäonnistuneita autentikointiyrityksiä. Packet Body Length- kertoo hyötykuorman pituuden. Varsinainen hyötykuorma on Packet Body -kentässä.



Kuva 4.4: EAPOL viestin rakenne

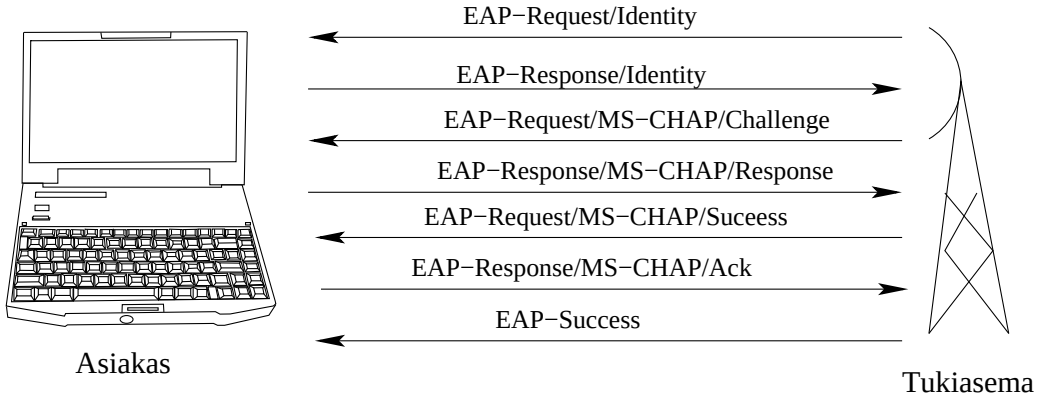
Varsinainen autentikointiprosessi menee siten, että käyttäjän laitteen liittyessä verkkoon, vastapäässä oleva verkkolaite lähettää EAP-Request/Identity viestin. Myös käyttäjän laite voi aloittaa autentikointiprosessin, jolloin se lähettäisi EAPOL-Start viestin, joka jälkeen vastapään verkkolaite lähettää EAP-Request/Identity viestin. EAP-Request/Identity viesti ei ole pakollinen, jos käyttäjä pystytään muuten tunnistamaan esimerkiksi MAC-osoiteella. Muuten autentikoituminen menee samaan tapaan kuin luvussa 2 kuvassa 2.4. Onnistautuneen autentikoitumisen jälkeen laite laittaa portin välittävään tilaan ja liikennöinti voi alkaa. Liikennöinnin lopettaminen suoritetaan EAPOL-Logoff viestin avulla. Kuitenkaan ei voida aina olettaa että käyttäjä poistuisi palvelusta kirjautumalla ulos. Tämän vuoksi on käytössä ajastin, joka säännöllisin väliajoin suorittaa uuden rekisteröinnin. Lisäksi jos fyysinen yh-

teys katkeaa, täytyy käyttäjän tunnistautua uudestaan. Näin vältytään siltä, että joku ottaisi heti saman portin käyttöön ja pystyisi liikennöimään ilman autentikoitumista.

Käytettävät MAC-osoitteet ovat hieman erilaisia riippuen siitä käytetäänkö tekniikkaa jossa MAC-osoite täytyy olla vastapään tiedossa vai ei. Esimerkiksi langattomissa lähiverkoissa virtuaaliportti muodostetaan MAC-osoitteen perusteella, jolloin EAPOL-viestin lähdeosoitteena on laitteen oma osoite ja kohdeosoitteena vastapuolen MAC-osoite. Jos taas käytössä on tekniikka, joka ei välitä vastapuolen MAC-osoitteesta, lähdeosoitteena on edelleen lähettäjän MAC-osoite, mutta kohdeosoitteena on ryhmäosoite 01-08-C2-00-00-03. Esimerkiksi perinteiset kytketyt lähiverkot toimivat viimeksimainitulla tavalla.

4.2.3 MS-CHAP EAP -autentikointi

Myös Microsoftin MS-CHAP -protokollan version 2 käyttö on mahdollista 802.1x -autentikoinnissa. Kyseisen toiminnallisuuden määrittely on vielä työn alla ja nykyinen määrittely [19] on IETF:n Internet-draftina. Käytännössä toiminnallisuus on lähes sama kuin normaalisti 802.1x:ssä, mutta EAP-protokollassa autentikoimiseen käytetään Microsoftin MS-CHAP protokollaa. MS-CHAP -protokollan etuna on salasanojen talletusmuoto. MS-CHAP:ia käytettäessä ei kummallakaan osapuolella tarvitse olla salasanoja selkokielisessä muodossa. Lisäksi MS-CHAP -protokolla tukee myös salasanojen hallintaa, eli sen avulla on esimerkiksi mahdollista vaihtaa salasana. Lisäksi käytettäessä EAP:issa MS-CHAP:ia, autentikoituminen on aina kaksisuuntainen.



Kuva 4.5: Onnistunut EAP/MS-CHAP autentikoituminen

Kuvassa 4.5 on esitetty onnistunut EAP/MS-CHAP autentikoituminen. Autentikointiprosessin alku on samanlainen kuin normaalisti EAP-protokollassa. Tukiasema lähettää auten-

tikoitumispyynnön asiakkaalle, ellei asiakasta voida tunnistaa jollain muulla menetelmällä. Tukiaseman saatua käyttäjätunnuksen, se lähettää asiakkaalle MS-CHAP -haasteen. Käyttäjä laskee salasanastaan tarkistussumman, ja lähettää sen tukiasemalle. Jos tukiasemalla tarkistussummat täsmäävät, lähettää se asiakkaalle MS-CHAP Success viestin. Tämän viestin tarkoitus on kertoa käyttäjlle, että autentikoituminen onnistui asiakkaan osalta. Asiakas tunnistaa tukiaseman Success-viestin avulla, joten autentikoituminen on siis aina kaksisuuntainen. Jos tukiasemankin autentikoituminen onnistuu, lähettää asiakas Ack-viestin. Tukiaseman saatua Ack-viestin lähettää se EAP-Success -viestin, ja autentikoituminen katsotaan onnistuneeksi.

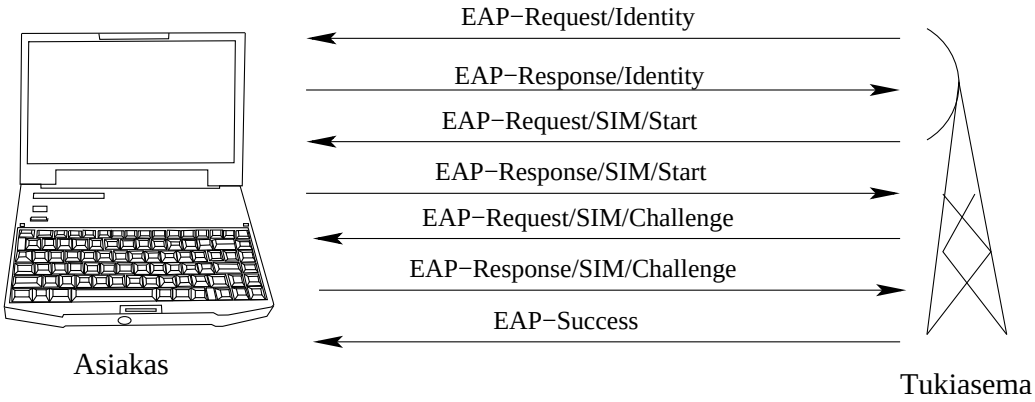
EAP/MS-CHAP tukee samaan tapaan RADIUS:ta kuin normaali EAP:kin. Tällöin toiminnallisuus on samanlainen kuin yllä olevassa esimerkissä, mutta autentikoinnin suorittaa RADIUS-palvelin, ja tukiasema toimii pelkästään liityntäpisteenä. Käytännössä MS-CHAP laajennus tuo EAP-protokollaan lisää kaksisuuntaisen tunnistamisen ja mahdollisuuden käyttää päätelaitteissa salattuja salasanoja.

Cisco käyttää omissa WLAN-ratkaisuissaan LEAP (Lightweigh EAP) -autentikointia. LEAP käyttää autentikoinnissa MS-CHAP -protokollaa. MS-CHAP:in avulla autentikaatiosta tulee kaksisuuntainen. Kuitenkin LEAP on paljon muutakin kuin autentikointia, minkä vuoksi MS-CHAP -autentikointia ei suoraan voida rinnastaa LEAP:iin. LEAP:issa on kiinnitetty suurempaa huomiota WEP:n heikkouteen ja ongelmiin. LEAP:issa WEP avaimet saadaan kyseisellä protokollalla, eikä niitä tarvitse etukäteen jakaa. Kuitenkaan Cisco ei ole julkaissut määrittelyä protokollastaan. Ainoa luotettava informaatio asiasta on Cisco:n myyntimainoksissa, jotka eivät kerro paljoakaan toiminnasta. Tämän vuoksi tässä ei laajemmin käsitellä LEAP-protokollaa .

4.2.4 EAP SIM-autentikointi

Nokia on tehnyt omiin WLAN-tuotteisiinsa SIM-kortin avulla tapahtuvan EAP-autentikoinnin, josta on julkaistu myös IETF:n Internet-draft [12]. Nokian WLAN -verkkokorteissa on SIM-kortin lukija, johon käyttäjä voi laitaa GSM SIM-korttinsa. Autentikointiprosessissa käytetään GSM pohjaista autentikoitumista. Tukiasema käy hakemassa käyttäjän kotioperaattorin autentikointipalvelimelta autentikaatiovektorin. Tässä vektoris-

sa on satunnaisluku, siihen odotettu vastaus ja verkon tunnistautumisparametri. Käyttäjä pystyy viimeksimainitulla parametrilla varmistamaan, että autentikaatiovektori on varmasti haettu oman kotiverkon palvelimelta. Autentikointipalvelimelta saatu satunnaisluku sekä verkon tunnisteparametri lähetetään käyttäjälle EAP-viestillä, ja käyttäjä myös vastaa pyyntöön EAP-viestillä. EAP SIM-autentikoinnin avulla vältytään erillisiltä käyttäjätunnuskannoilla, mutta se vaatii pääsyn GSM-verkon autentikointipalvelimelle. Käytännössä tämä rajoittaa palvelun käytön ainoastaan GSM-palveluita tarjoavalle palveluntarjoajalle, koska tietoturvasyistä autentikointipalvelimet eivät voi olla kaikkien saatavilla.



Kuva 4.6: Onnistunut EAP SIM autentikoituminen

Onnistunut EAP SIM-autentikoituminen on esitetty kuvassa 4.6. Aluksi tukiasema lähettää EAP Identity-kyselyn, johon käyttäjä vastaa omalla GSM IMSI-tunnuksellaan. Tämän jälkeen tukiasema pyytää EAP SIM-autentikoinnin aloittamista. Jos asiakaslaite tukee sitä, vastaa se EAP-Response-viestillä, jossa on SIM-parametrit mukana. Jos tukiasemalla ei ole valmiina kyseiselle käyttäjälle autentikaatiovektoria, se hakee autentikaatiovektorin käyttäjän kotiverkon GSM-autentikointipalvelimelta. Kun tukiasemalla on käyttäjälle autentikaatiovektori, se lähettää EAP-viestin, jossa on haasteena vektorista otettu satunnaisluku ja verkon tunnistusluku. Viestin saatuaan asiakaslaite laskee ensin, että vektori on saatu oikealta palvelimelta ja sen jälkee laskee satunnaisluvulla ja omalla salaisella avaimellaan tarkistussumman. Laskettuaan tarkistussumman, lähettää se vastauksensa tukiasemalle. Jos tarkistussummat täsmäävät katsotaan autentikoituminen onnistuneeksi ja tukiasema lähettää EAP-Success -viestin.

Oletusarvoisesti kaikki EAP SIM -viestit sekä niiden parametrikentät kulkevat salaamattomina. Koska ilmatie kuitenkin on turvaton, voidaan EAP SIM autentikoinnissa käyttää myös kenttien salausta. Tällöin esimerkiksi Identity-kenttä salataan jollain ennaltasovitulla

avaimella, esimerkiksi edellisessä istunnossa käytössä olleella avainparilla. Jos tukiasemalla ei ole avainta tiedossaan, lähettää se uuden pyynnön, jossa se pyytää Identity:ä selkokielisessä muodossa. Toinen vaihtoehto ongelman ratkaisuksi voisi olla EAP-TLS, mutta sitä ei ole käsitelty nykyisessä määrittelyssä.

5 3G-autentikointi

Tässä luvussa käsitellään vertailun vuoksi 3G-verkoissa tapahtuva käyttäjän tunnistaminen. Tunnistamisprosessi jakautuu kahteen pääkategoriaan: radiotien tunnistautumiseen ja IP-palveluihin tunnistautumiseen. Radiotien tunnistamista voidaan verrata edellisessä luvussa käsiteltyyn lähiverkon tunnistamiseen. IP-multimediapalveluiden tunnistus yhdistää EAP-protokollan SIP:n tunnistamismenetelmäksi ja lopullinen tunnistamisprosessi suoritetaan Diameter-protokollalla. IP-palveluihin tunnistautumiseksi voidaan laskea myös mobile-ip-palveluun tai vastaavan palveluun tunnistautuminen, jolloin SIP-palvelun tunnistamisen kanssa prosessi voi olla kolmitasoinen. Viimeksi mainittua palvelua ei kuitenkaan käsitellä tässä työssä.

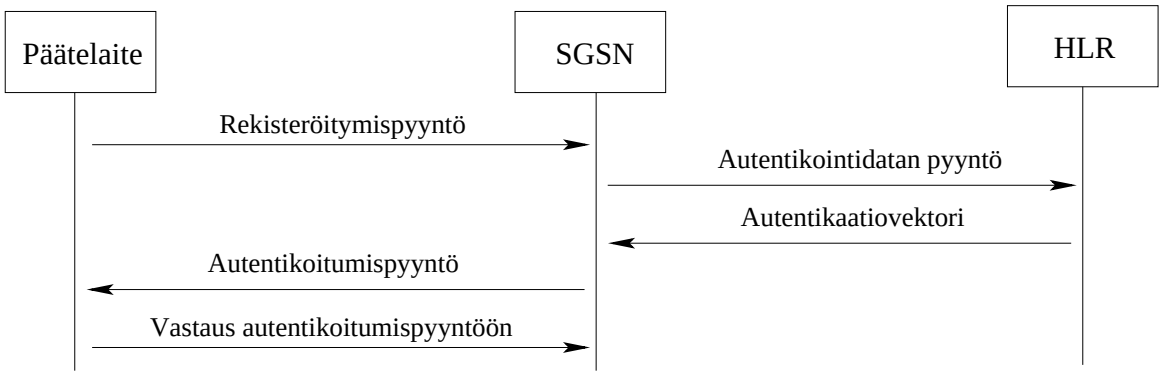
5.1 Radiotielle autentikoituminen

Radiotien autentikoitumisprosessi on melko samanlainen kuin GSM-verkossa. Yksi merkittävin syy tähän on halu mahdollisimman helposti mahdollistaa 3G- ja GSM-verkkojen välinen yhteistoiminta. 3G-verkoissa on määritelty sekä piirikytkentäinen että pakettikytkentäinen verkko. Tunnistautuminen molemmissa verkoissa on samanlainen, joten käsittelemme ainoastaan pakettikytkentäiseen verkkoon autentikoitumista.

Tärkeimmät komponentit pakettikytkentäisessä verkossa ovat: itse päätelaite, SGSN (Serving GPRS Support Node) ja HLR (Home Location Register). SGSN on laite, joka yhdistää pakettikytkentäisen verkon matkapuhelinverkkoon. Se on samalla myös laite, johon

käyttäjän täytyy kirjautua halutessaan pakettikytkentäistä palvelua. HLR on palvelu johon käyttäjän SIM-kortin salainen avain liittymän avauksen yhteydessä tallennetaan. Tätä salaista avainta käytetään autentikointiprosessissa, mutta sitä ei koskaan siirretä laitteiden välillä. Koska HLR on tavallaan vain yksi palvelin se ei suorita itse tunnistamista vaan antaa SGSN:lle autentikaatiovektorin, jonka avulla SGSN suorittaa autentikoinnin.

SGSN voi olla joko oman palveluntarjoajan tai jonkun toisen palveluntarjoajan laite, kuitenkin autentikaatiovektori haetaan aina oman palveluntarjoajan HLR:stä AKA (Authentication and key agreement) -protokollalla, joka on sama kuin nykyisin GSM:ssä käytössä oleva. Autentikaatiovektorissa on myös parametri, joka SGSN:n täytyy välittää päätelaitteelle autentikointipyynnön yhteydessä. Tällä parametrilla päätelaite voi varmistua siitä, että autentikaatiovektori on saatu omalta HLR:ltä. Näin molemmat osapuolet voivat varmistua, että toinen on se joka väittää olevansa. Autentikointiprosessi on siis kaksisuuntainen, verkko autentikoi käyttäjän ja käyttäjä verkon. Tämä toiminnallisuus on erittäin tärkeä, jos SGSN on toisen palveluntarjoajan laite.



Kuva 5.1: Onnistunut radiotien tunnistautuminen

Kuvassa 5.1 on kuvattu onnistunut radiotien autentikointi. Päätelaitteen halutessa rekisteröityä SGSN:lle, pyytää SGSN HLR:ltä autentikaatiovektoria, ellei SGSN:llä sitä ole jo valmiina. Pyynnön jälkeen HLR laskee käyttäjälle autentikaatiovektorin, jossa on yksi tai useampia satunnaislukuja, sekä niihin odotettuja vastauksia. Lisäksi autentikaatioverktorissa on kyseiselle satunnaisluvulle laskettu salaukseen ja viestien aitouden toteamiseen käytetyt avaimet, ja AUTN-parametri, jota päätelaite käyttää verkon autentikointiin. Kun SGSN on saanut vektorin, se ottaa siitä ensimmäisen käyttämättömän satunnaisluvun ja liittää sen päätelaitteelle lähetettävään autentikointipyyntöön. Päätelaitteen saadessa autentikointipyynnön, se laskee ensin salaisen avaimensa, satunnaisluvun ja AUTN-parametrin

avulla tarkistussumman. Tällä tarkistussummalla se pystyy varmistamaan että SGSN on saanut autentikaatiovektorinsa käyttäjän kotiverkon HLR:ltä eikä mahdolliselta hyökkääjältä. Jos tarkistussuma on sama kuin AUTN-parametrissa oleva, laskee päätelaite satunnaisluvun ja salaisen avaimensa avulla vastauksen SGSN:n autentikointipyyntöön. Päätelaite liittää tämän tarkistussumman autentikointipyynnön vastaukseen, ja lähettää sen SGSN:lle. Tämän jälkeen päätelaite laskee vielä itselleen käytettävät salaus- ja eheysavaimet. SGSN:n saadessa vastauksen, se vertaa saatua tarkistussummaa autentikaatiovektorissa olevaan. Jos tarkistussummat täsmäävät, SGSN tulkitsee autentikoinnin onnistuneeksi ja liikennöinti voi alkaa.

Jos tarkistussummat eivät vastaa toisiaan jossain edellä mainituissa tarkistuksissa, täytyy yhteyden muodostus keskeyttää ja lähettää virheilmoitus. SGSN voi virhetilanteessa pyytää uuden autentikaatiovektorin HLR:ltä tai aloittaa uuden yhteyden muodostuksen seuraavalla autentikaatiotusvektorissa olevalla satunnaisluvulla. Turvallisuussyistä se ei voi käyttää samaa lukua uudestaan.

5.2 IP Multimedia-palveluun tunnistautuminen

IP Multimedia-palvelun signalointiprotokollaksi on valittu SIP ja autentikointiprotokollaksi EAP. Jotta ratkaisut olisivat mahdollisimman skaalautuvia, varsinaiset autentikointitiedon kyselyt suoritetaan Diameterin avulla. Kaikki nämä protokollat on esitelty aikaisemmissa luvuissa ja tämä luku keskittyy käsittelemään niiden yhteistoimintaa.

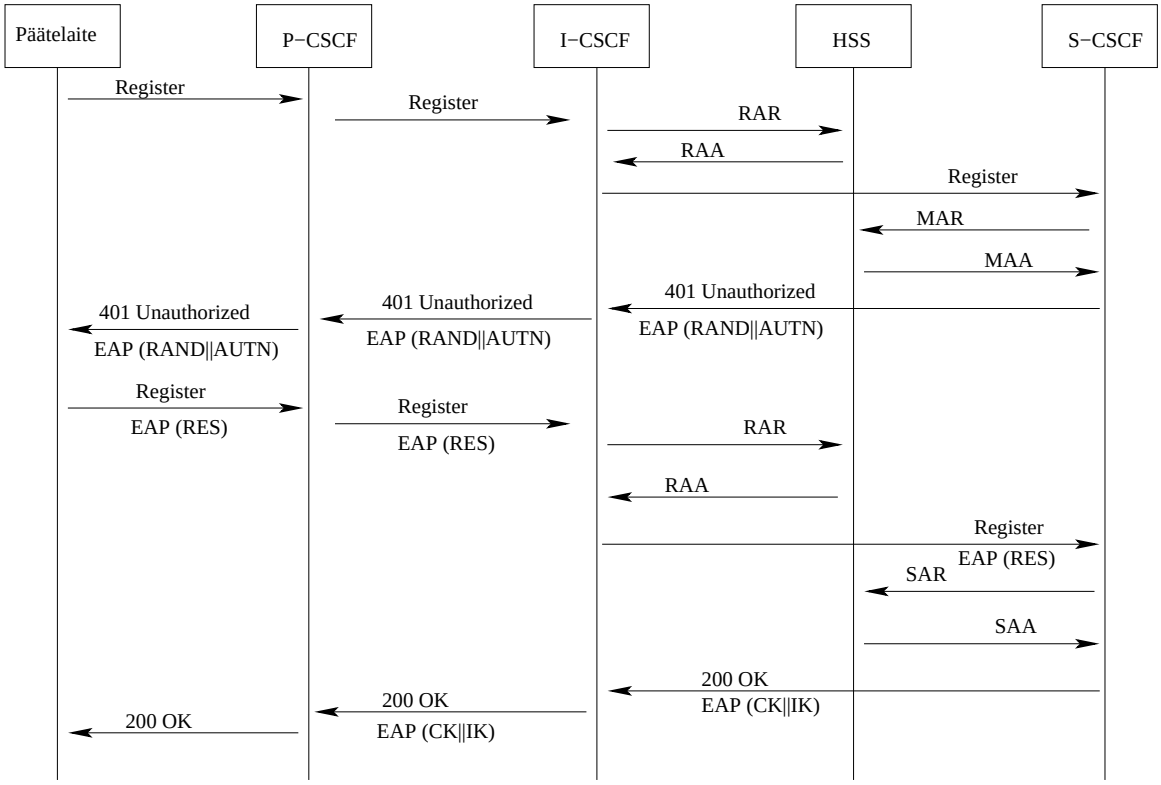
IP Multimedia-palvelun autentikointiprosessi käyttää hyväkseen edellisessä luvussa mainittua autentikaatiovektoria. EAP-protokollaan joudutaan tästä syystä lisäämään hieman toiminnallisuutta muutaman uuden viestityypin muodossa. Autentikointi tapahtuu muuten samaan tapaan kuin kuvassa 2.4, mutta nyt MD5-Challengen asemasta lähetetään AKA-Challenge. AKA-Challenge -viesti sisältää autentikaatiovektorissa olleen satunnaisluvun ja AUTN-parametrin. Vastaavalla tavalla asiakkaan lähettämässä vastauksessa MD5-Challengen tilalla on AKA-Challenge ja parametrina laskettu tarkistussumma saadulla satunnaisluvulla. Lisäksi on määritelty muutama muukin uusi viestityyppi, mutta ne eivät ole merkityksellisiä tämän luvun kannalta.

Koska SIP-protokolla ei tue suoraan EAP-protokollaa autentikointiprotokollanaan, siihen joudutaan tekemään pieniä muutoksia. Protokollaan täytyy määritellä uusi autentikointikeino HTTP EAP-tunnistus. Toiminnallisuus on on muuten samanlainen kuin muissakin autentikointimenetelmissä, mutta nyt Unauthorized -viestissä pyydetään EAP:iin liittyviä parametrejä. Ensimmäisessä Unauthorized-viestissä palvelin pyytää asiakkaalta EAP ID:tä. Asiakas lähettää EAP ID:nsä, jonka jälkeen palvelin antaa EAP-haasteen. Käyttäjä laskee nyt haasteellaan salasanastaan tarkistussuman ja lähettää sen vastauksena palvelimelle. Jos tarkistussummat täsmäävät autentikointi on mennyt oikein, ja asiakas katsotaan rekisteröityneeksi.

IP Multimedia-palvelussa on samaan tapaan kuin radiotiellä yksi asiakkaan omassa kotiverkossa oleva palvelin, jossa pidetään käyttäjään liittyvät autentikointitiedot. Tässä palvelussa kyseisen laitteen nimi on HSS(Home Subscriber Server). Kyseisellä palvelimella on myös tieto SIP-palvelimesta, johon käyttäjä on kirjautuneena, sekä kaikkien oman verkon SIP-palvelinten osoitteet sekä SIP-palvelinten tukemat parametrit. Rajapinta, jolla HSS:n kanssa vaihdetaan tietoa, on Diameter perustainen. Käytännössä tämä vaatii kuitenkin kokonaan uuden Diameter-sovelluksen määrittelyn, koska käytettävät viestityypit ja attributti/arvo-parit eroavat merkittävästä tällä hetkellä määriteltynä olevista sovelluksista.

Kuvassa 5.2 on esitetty tärkeimmät autentikointiin liittyvät komponentit ja onnistunut autentikointitapahtuma. P-CSCF (Proxy- Call Session Control Function) on laite, joka välittää käyttäjän pyynnöt kohti omaa palvelevaa SIP-palvelinta. P-CSCF voi olla joko oman operaattorin tai vieraan operaattorin laite. I-CSCF (Interrogating-CSCF) on käyttäjän kotiverkon laidalla oleva laite, joka ottaa vastaan kaikki SIP-palvelua pyytävät pyynnöt ja HSS:n avustuksella ohjaa ne oikealle SIP-palvelimelle. S-CSCF (Serving-CSCF) on laite, johon käyttäjät ovat rekisteröityneet. S-CSCF on siten varsinainen SIP-palvelun toteuttava laite. Näitä laitteita voi olla palveluntarjoajan verkossa useampia ja I-CSCF ja HSS valitsevat niistä sopivimman palvelua suorittamaan. Kaikki nämä CSCF:t ovat SIP- proxy -palvelimia, mutta niillä on vain erilainen toimenkuva.

Onnistuneessa autentikointitapahtumassa on yhdistetty kaikki alussa kuvatut protokollat ja niiden vaatimat muutokset toimivaksi kokonaisuudeksi. Alussa käyttäjän päätelaite lähettää rekisteröitymispyynnön P-CSCF:lle, joka ohjaa sen oikealle I-CSCF:lle. I-CSCF kysyy Diameterin RAR (Registration-Authorization-Request) -viestillä tietoa käyttäjän rekiste-



Kuva 5.2: Onnistunut IP Multimedia-tunnistautuminen

röitymisestä. Jos käyttäjä on jo rekisteröitynyt, palautuu Diameterin RAA (Registration-Authorization-Answer) -viestissä S-CSCF:n osoite, johon käyttäjä on rekisteröitynyt. Ellei käyttäjä ole rekisteröitynyt, RAA-viestissä tulee kaikkien S-CSCF:ien osoitteet sekä tuetut parametrit, joista I-CSCF valitsee parhaimman. Saatuaan S-CSCF:n osoitteen, I-CSCF lähettää rekisteröintipyynnön valitulle S-CSCF:lle. Saatuaan pyynnön S-CSCF pyytää Diameterin MAR (Multimedia-Authentication-Request) -viestillä HSS:ltä autentikaatiovektorin rekisteröintipyynnössä olevalle EAP ID:lle. HSS laskee tunnistusvektorin ja palauttaa sen Diameterin MAA (Multimedia-Authentication-Answer) -viestillä. Tämän jälkeen S-CSCF lähettää Unauthorized-viestin, jossa pyytää käyttäjää autentikoitumaan EAP:illa, ja laskemaan tarkistussumman RAND-parametrillä. Päätelaitteen saatua viestin, se tarkistaa ensin AUTN -parametristä, että S-CSCF on oman kotiverkon laite. Jos tarkistussummat täsmäävät, laskee päätelaite omalla salaisella avaimella saadusta satunnaisluvusta tarkistussumman ja lähettää sen uudessa rekisteröitymispyynnössä P-CSCF:lle. Rekisteröitymispyyntö ohjautuu oikealle S-CSCF:lle aikaisemmin kuvatulla tavalla. Kun S-CSCF saa pyynnön, se tarkistaa että vastaus on sama kuin tunnistusvektorissa. Jos summat täsmäävät, lähettää S-CSCF HSS:lle Diameter-viestin SAR (Server-Assignment-Request), jossa

se kertoo HSS:lle käyttäjän rekisteröityneen siihen, ja pyytää HSS:ää päivittämään tietonsa tämän käyttäjän osalta. HSS vastaa tähän Diameter-viestillä SAA (Server-Assignment-Answer), jossa se kuittaa saadun tiedon, sekä kertoo onnistuiko tietojen päivitys. Lopuksi S-CSCF lähettää OK-viestin merkkinä onnistuneesta rekisteröitymisestä, sekä kertoo salaus- ja eheysavaimet I-CSCF:lle ja P-CSCF:lle. Päätelaitteelle niitä ei kerrota, vaan sen täytyy laskea ne itse saamastaan AUTN-parametrista.

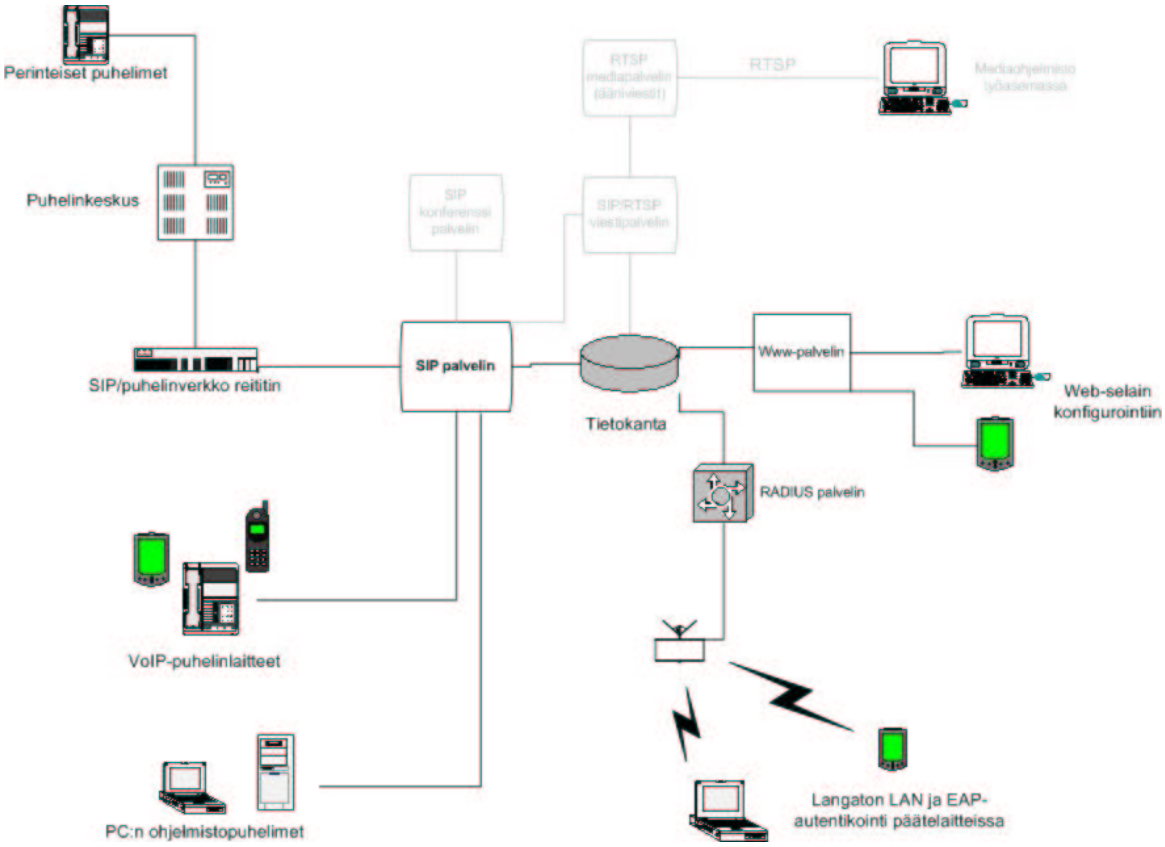
6 WirLabin palveluiden toteutus

WirLab-projektin yhtenä tutkimuskohteena on langattomien yhteyksien, ja IP-pohjaisten kommunikointipalveluiden tarjoaminen sekä paikallisesti että suuremmalla maantieteellisellä alueella. WirLabin malli näiden palvelujen toteuttamiseksi on esitetty kuvassa 6.1. Langattomissa verkoissa käyttäjän autentikointiin käytetään ensisijaisesti EAP-pohjaista 802.1x -autentikointia. Kommunikointipalveluiden toteuttamiseen käytetään SIP-protokollaa. Tämän työn käytännön osuudessa on keskitytty näiden palveluiden käyttäjätietokantojen yhdistämiseen. Kuvassa harmaalla näkyvät palvelut eivät ole tällä hetkellä vielä kehityksen alla, mutta tulevaisuudessa niitäkin tullaan kehittämään.

6.1 Yhteisen käyttäjätietokannan toteutus

Kuten aikaisemmista luvuista käy ilmi, ei EAP:illa ja SIP:llä ole tällä hetkellä yhteistä autentikoinnissa käytettävää protokollaa. SIP-protokollassa ei ole kunnollista RADIUS-tukea, eikä EAP-protokollassa ole SIP:n käyttämiä HTTP-autentikointimenetelmiä.

Koska RADIUS-palvelimen ja SIP-palvelimen käyttämien SQL-taulujen välillä ei ole ristiriitaisuuksia, ratkaistiin ongelma WirLab-projektissa sijoittamalla kaikki taulut samaan MySQL-kantaan. Kuitenkin molemmilla sovelluksilla on kyseisessä tietokannassa omat taulut, koska SIP-palvelin ei käytä selkokielisessä muodossa olevia salasanoja, vaan niistä laskettuja tarkistussummia, joita käytetään syötteenä MD5-algoritmille. Kumpikin sovellus käyttää siten omia autentikointitapojaan ja käyttäjätunnustaulujaan. Käyttäjätunnus-



Kuva 6.1: WirLabin testi- ja kehitysympäristö

ten hallinta toteutettiin WWW-pohjaisella käyttöliittymällä. Käyttöliittymän avulla tietokannan rakenne saadaan peitettyä, joten käyttäjätietoja voidaan hallita keskitetysti yhdestä paikasta. Käyttäjälle sallittavat palvelut määritellään käyttöliittymästä, jonka jälkeen sovellus käy lisäämässä tai muuttamassa käyttäjän tietoja osasovelluskohtaisissa tauluissa. Myös loppukäyttäjälle on oma WWW-pohjainen käyttöliittymä, joka on tarkoitettu ensisijaisesti käyttäjän omien tietojen hallintaan ja tarkasteluun. Samassa tietokannassa on myös WWW-sovelluksiin liittyvät taulut, jotta käyttäjän tunnistus WWW-palveluun voidaan suorittaa muista palveluista riippumattomasti.

6.2 Käyttäjätietokannan hallinta

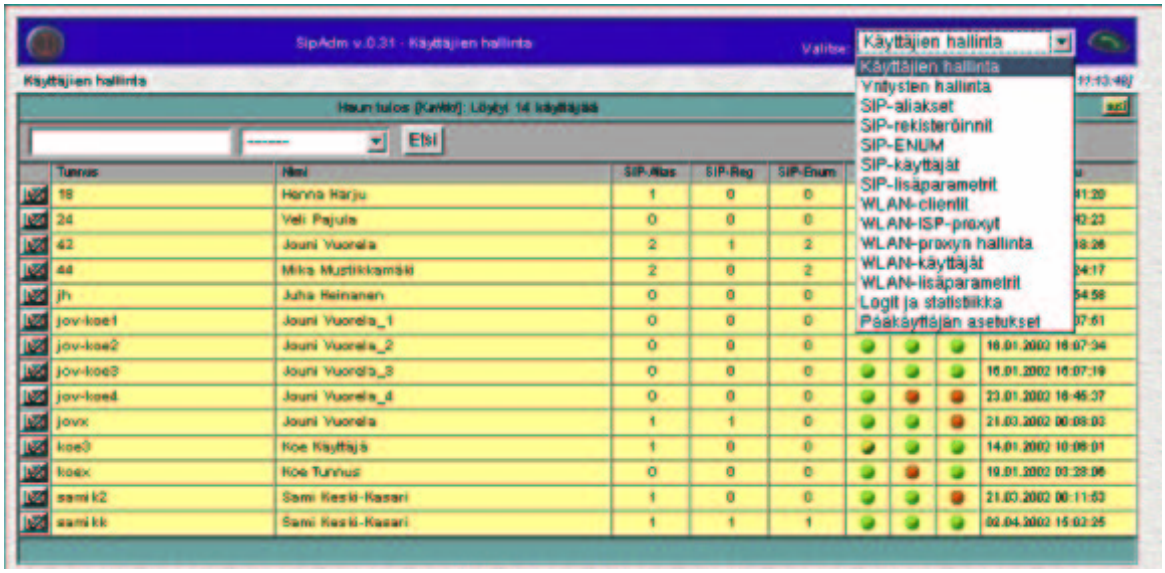
Pääkäyttäjän hallintalomake on esitelty tässä luvussa. Hallintalomake on toteutettu perl-pohjaisella CGI-skriptillä, jonka generoimat sivut ovat HTML:ää (Hypertext Markup Language). Tämä mahdollistaa sovelluksen käytön mahdollisimman monilla laitteilla ja selai-

milla. Käyttäjän autentikoinnissa käytetään WWW-palvelimen omaa autentikointimekanismia. Sovellusta on mahdollista käyttää sekä SSL-salattuna että selkokielisenä. Sovellus on tehty mahdollisimman modulaariseksi, joten sen avulla hallittavia palveluita voidaan lisätä tai poistaa helposti.

Kuva 6.2: Hallintaikkunan päänäkymä

Kuvassa 6.2 on kirjautumisen jälkeen avautuva näkymä. Sivulla on listattuna kaikki järjestelmässä määriteltynä olevat käyttäjät. Yhdelle sivulle tulostetaan oletusarvoisesti 20:n käyttäjän tiedot. Jos käyttäjiä on enemmän, muodostetaan useampia sivuja, joita voi selata yläkulmassa olevien nuolien avulla. Tietyn käyttäjän löytämiseksi sivulle on toteutettu myös etsintätoiminne. Käyttäjää voidaan etsiä sekä tunnuksen että nimen avulla.

Kuvassa 6.3 on päänäkymän yläkulmassa oleva valikko avattuna. Valikon SIP-alkuiset vaihtoehdot ovat SIP-palvelimeen liittyviä lisätoiminteita. Käyttäjien hallinta ei tapahdu niiden kautta, vaan toiminteet on tarkoitettu etupäässä SIP-palvelimen parametrien ja käyttäjän tietojen yhdistämiseen. Esimerkiksi sillä voidaan tarkistaa kaikki tällä hetkellä rekisteröityneet käyttäjät. Valikon WLAN-alkuiset vaihtoehdot mahdollistavat samanlaisen toiminnalisuuden, mutta tietojen yhdistäminen tapahtuu RADIUS-palvelimen suhteen. Valikossa on myös käyttäjään liittymättömiä WLAN-parametrejä, kuten WLAN-clientit, WLAN-ISP-Proxyt ja WLAN-proxyn hallinta. Näiden vaihtoehtojen avulla määritellään RADIUS-



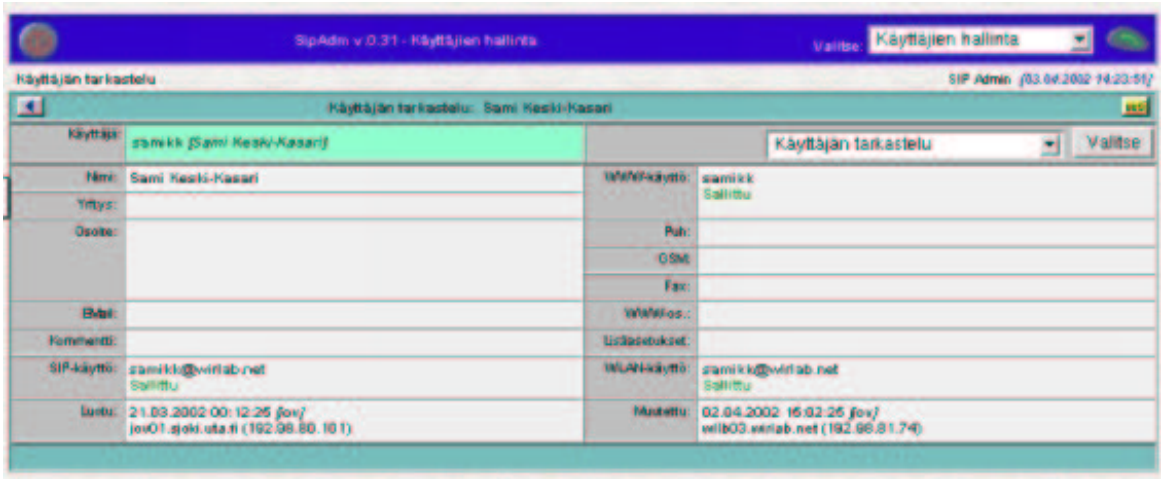
Kuva 6.3: Päänäkymän valikko

palvelua käyttäviä asiakkaita RADIUS-palvelimelle.

Valittaessa Yrityksen hallinta-vaihtoehto, avautuu uusi käyttäjänhallintaa muistuttava sivu. Sivulla on listattuna kaikki sillä hetkellä määritellyt yritykset, sekä niihin liittyviä parametrejä. Yrityksen hallinnassa määritellään yritykseen liittyviä tietoja, kuten nimi ja puhelinnumero. Yritystietojen kautta voidaan käyttäjälle periyttää oikeuksia sekä muuta informaatiota.

Jokaisella rivillä on käyttäjän tunnus ja nimi. Lisäksi rivillä näkyvät SIP-aliaksien lukumäärä, sekä SIP-rekisteröintien lukumäärä. Väripallot kuvastavat käyttöoikeuksia. Mahdollisia palveluita ovat WWW, SIP ja WLAN-palvelu. Näitä kuvaavat lyhenteen ovat W3,S ja WL. Mahdollisia värikoodeja on kolme, jotka kuvaavat käyttäjän oikeuksia palveluun. Vihreä tarkoittaa palvelun olevan mahdollinen, keltainen tilapäisesti estetty ja punainen kielletty. Tilapäisesti estetyssä palvelussa käyttäjän tietoja ei poisteta palvelun tietokantataulusta. Lisäksi rivillä on päivämäärä, jolloin käyttäjän asetuksia on viimeksi muutettu. Klikkaamalla nimeä tai tunnusta päästään kuvassa 6.4 esitetylle sivulle.

Kuvassa 6.4 on esitetty käyttäjän tarkasteluun liittyvä ikkuna. Sivulla näkyy kyseisellä hetkellä määriteltynä olevat tiedot, kuten nimi, osoite ja puhelinnumero. Lisäksi sivulta näkee käyttäjän tunnukset ja oikeudet eri palveluihin. Kuvassa olevalla käyttäjällä on kaikki palvelut käytössä, mutta ne voivat olla toisistaan riippumatta estettynä. Kuvassa 6.5 on



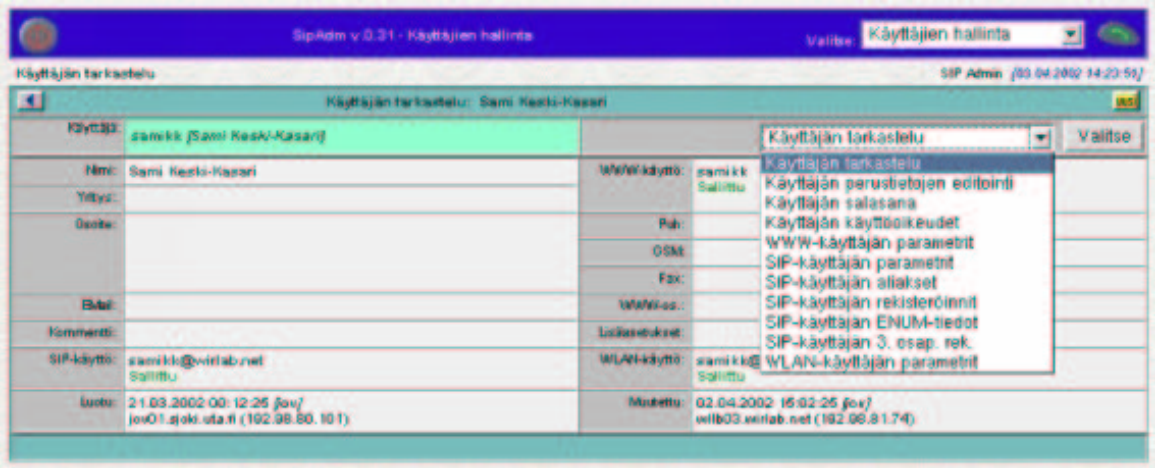
Kuva 6.4: Käyttäjän tietojen tarkastelu

käyttäjän tarkasteluikkunan valikko avattuna. Valikon vaihtoehtojen avulla on mahdollista muuttaa käyttäjän tietoja, kuten perustietoja sekä salasanoja. Toistaiseksi salasanat ovat samoja kaikissa palveluissa. Lisäksi valikossa on vaihtoehtoja, joilla käyttäjän parametrejä ja oikeuksia palveluihin voidaan muokata.

SIP-palvelussa on tällä hetkellä eniten muokattavia parametrejä ja oikeuksia. Määriteltäviä oikeuksia ovat palvelun käyttöoikeus, käyttäjälle sallitut aliakset, sekä kolmannen osapuolen rekisteröintioikeus. Sallituilla aliaksilla on mahdollista lisätä käyttäjän osoitteita määrittelemättä uutta käyttäjätunnusta. Kolmannen osapuolen rekisteröinnillä tarkoitetaan oikeutta rekisteröidä tai käyttää soitettaessa jotain vierasta käyttäjätunnusta. Esimerkiksi kaikilla myyjillä on oikeus rekisteröidä ja käyttää tunnusta myynti@domain. Lisäksi sovelluksen avulla on mahdollista tehdä rekisteröinti SIP palveluun ilman SIP-signalointia, jolloin käyttäjä voi tehdä soitonsiirron toiseen numeroon. SIP-palveluun liittyviä muokattavia parametrejä ovat käytettävä algoritmi käyttäjän tunnistuksessa, tuetut viestityypit, sekä oikeus PSTN (Public Switched Telephone Network) -signalointiyhdyskäytävän käyttöön.

6.3 SIP-toteutus

Projektissa SIP-palvelu on toteutettu mukautetulla Columbia University:n sipd-palvelimella. Sipd-palvelin valittiin, koska se oli tutkimuskäyttöön ilmainen ja siihen oli mahdollista saada myös lähdekoodi. Tarpeita vastaavaa palvelinta lähdettiin kehittämään



Kuva 6.5: Käyttäjän tietoihin kohdistuvat toiminteet

sipd:n versiosta 1.18. Palvelimessa oli jo valmiiksi toteutettuna perustoiminnallisuus, sekä käyttäjän tunnistus. Käyttäjän autentikointiin oli kaksi vaihtoehtoa, HTTP Basic ja HTTP Digest. Johtuen HTTP Digestin paremmasta tietoturvasta, se otettiin käyttöön. Käyttäjätietokannan toteutukseen oli useampia vaihtoehtoja. Tuettuina oli unixin salasanatietokanta, joka käytännössä on pelkkä paikallinen tiedosto, jossa on käyttäjätunnus ja salasana. Muista vaihtoehtoja olivat palvelinpohjaiset LDAP ja MySQL -tietokannat. Näistä valittiin MySQL-tietokanta, johtuen sen parhaasta yhteensopivuudesta muihin palveluihin.

Vaikka palvelimessa oli MySQL-tietokannalle tuki valmiina, huomasimme sen varsin huonosti testatuksi, sekä epävakaaksi. Autentikointitiedon hakeminen palvelimelta ei onnistunut johtuen muuttujien väärästä käsittelystä. Esimerkiksi käyttäjätietoja etsittiin muuttujista ennen SQL-kyselyn suorittamista. Korjaustyötä haittasi ohjelman dokumentoinnin puute, sekä itse ohjelmassa virhetarkastelujen puuttuminen. Lisäksi ohjelmassa oli suuria muistivuotoja, liittyen MySQL-kyselyihin ja niiden käsittelyyn.

Ohjelma muutenkin tuntui epävakaalta ja keskeneräiseltä. Virhetarkastelut oli jätetty osittain tai kokonaan pois, mikä aiheutti epävakautta ohjelman toiminnassa. Osittain keskeneräisyys ja epävakaus voidaan selittää sillä, että ohjelma ei ehkä ole suoraan tarkoitettu kaupalliseen käyttöön. Kohderyhmä on oletettavasti tutkimuskäyttäjät, joita ohjelman kaatuilu tai toimimattomuus ei niin suuresti haittaa. Varsinkin kun kaikki kaupallisesti myytävät tuotteet maksavat melkoisesti. Lisäksi kaupallisiin sovelluksiin ei yleensä saa lähdekoodeja, jolloin ohjelman mukauttaminen ja pienien virheiden korjaaminen on lähes mahdotonta.

Saatuamme toteutuksen kannalta suurimmat virheet korjattua, pääsimme tekemään tarvittavia toiminnallisuuden muutoksia. Sipd:ssä oli jo alustavaa toteutusta kolmannen osapuolen rekisteröintien suhteen. Tietokannassa oli jo tarvittavat taulut ja kentät, mutta itse totetus puuttui. Käytännössä työksemme jäi tietojen hakeminen tietokannassa ja niiden sopiva käsittely rekisteröinnissä ja yhteyden muodostuksessa. Kolmannen osapuolen rekisteröinnissä on kyse aliaksista tai tunnuksista, joita käyttäjän on mahdollista rekisteröidä tai käyttää yhteyden lähdeosoitteena. Toteutuksessa täytyy siis katsoa autentikoinnin yhteydessä lähdeosoite, jolla käyttäjä lähettää sanomansa. REGISTER- ja INVITE-sanomien käsittelyssä ei siis pelkästään riitä käyttäjätunnuksen ja salasanan tarkistaminen, vaan täytyy myös katsoa viestin FROM-kenttää INVITE-viestissä ja viestin TO-kenttää REGISTER-viestissä. Koska käyttäjällä voi olla oikeus rekisteröidä käyttäjä, mutta ei muodostaa yhteyttä, täytyy tietokannassa olla kaksi erillistä kenttää. Toisessa on määritelty tunnukset, jotka käyttäjä voi rekisteröidä ja toisessa tunnukset joilla voi soittaa. Tunnuksia voi olla useampia kuin yksi, jolloin käytettävän tietorakenteen täytyy olla taulukko. Koska tunnuksia voidaan olettaa olevan kuitenkin melko pieni määrä, ei toteutuksessa käytetty mitään edistyneempiä tietorakenteita. Tunnistusvaiheessa luetaan järjestyksessä taulukkoa, josta etsitään käyttäjän käyttämää tunnusta. Jos tunnus löytyy hyväksytään rekisteröinti tai yhteyden avaaminen, muussa tapauksessa toimitaan samoin kuin virheellisen käyttäjätunnuksen tapauksessa.

6.4 WLAN-toteutus

WirLabin WLAN-toteutus perustuu nykyisin 802.11b -standardin mukaisiin laitteisiin. Käyttäjän autentikointi perustuu RADIUS-pohjaisiin autentikointimenetelmiin. Ensisijaisena tunnistusprotokollana on EAP. Koska EAP-protokolla on melko uusi autentikointiprotokolla, ei monissa laitteissa ole sille vielä toteutusta. Tämän vuoksi myös muita autentikointitapoja täytyy tutkia. Tämän vuoksi WirLabissa ei laitteita ole sidottu kiinteästi EAP-protokollaan, vaan muitakin tapoja voidaan ottaa helposti käyttöön. Ainoa vaadittava asia vaihtoehtoisilla tavoilla on, että nekin tukevat RADIUS-protokollaa.

Vaihtoehtoinen menetelmä, jota WirLabissa on kokeiltu, on WWW-pohjainen autentikointi. Tämä menetelmä on käytössä useissa nykyisin rakennetuissa verkoissa, johtuen sen yhteensopivuudesta. Käytännössä menetelmä toimii siten, että käyttäjä avaa WWW-selaimensa.

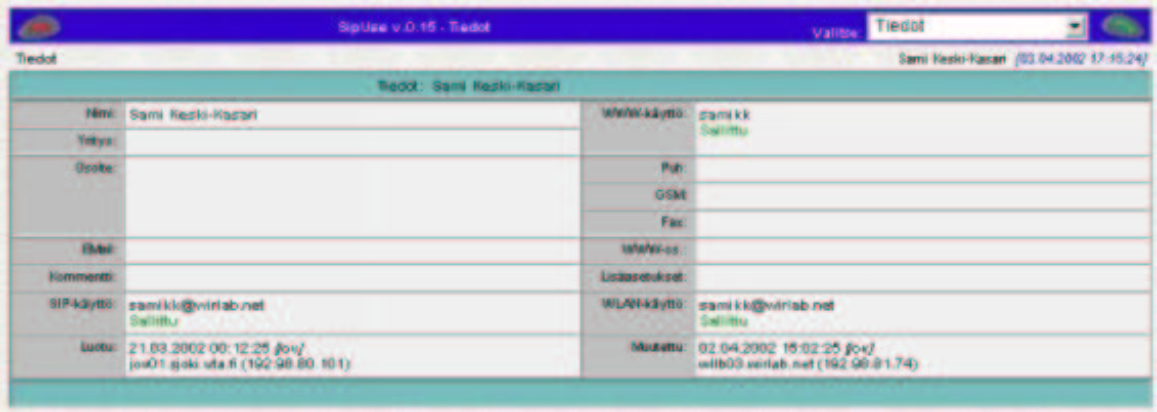
Tällöin reitittimenä toimiva laite kaappaa paketin ja ohjaa käyttäjän autentikointisivulleen. Käyttäjä syöttää avautuneeseen lomakkeeseen käyttäjätunnuksen ja salasanan, jonka jälkeen laite suorittaa RADIUS-autentikoinnin. Jos tunnukset ovat oikein, päästetään käyttäjän liikenne läpi laitteesta, muulloin se estetään. Tämä ratkaisu on varsin yksinkertainen mutta vaatii aina sen, että verkkoon liitettävässä laitteessa on WWW-selain ja joku kirjoittamassa käyttäjätunnuksen ja salasanan. Tämä menetelmä ei siis voi olla lopullinen käytössä oleva ratkaisu. Kuitenkin se täyttää tehtävän EAP-toteutuksien puuttuessa.

RADIUS-palvelimena on käytössä Australilaisen Open System Consultants-yrityksen valmistama Radiator palvelinohjelmisto. Radiator on kohtuuhintainen ja voimakkaasti kehittyvä ohjelma, minkä vuoksi se on WirLabissa käytössä. Tämän projektin aikana Radiatorista julkaistiin versio, jossa on EAP-protokolla tuettuna. Lisäksi Radiatorissa ei ole käyttäjätietokantaa sidottuna mihinkään valmistajakohtaiseen ratkaisuun, vaan se tukee yleisimpiä vaihtoehtoja. Tämän ansiosta WirLabissa Radiator käyttää MySQL:ää käyttäjätietokantana.

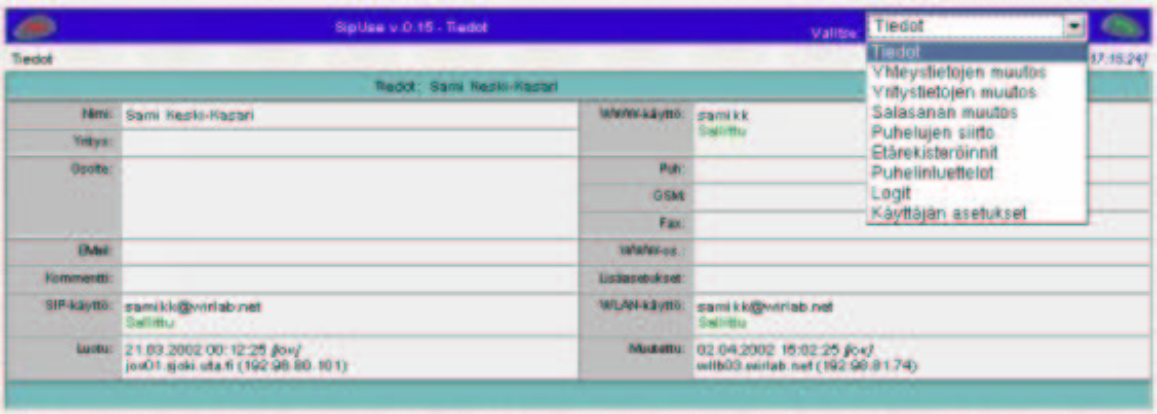
6.5 Loppukäyttäjän palvelut

WirLabin käyttäjähallintaan kuuluu myös käyttäjille tarkoitettu WWW-palvelu. Pääkäyttäjä voi tunnusta luodessaan määritellä käyttäjälle oikeuden käyttää WWW-palvelua. Palvelun avulla käyttäjän on itse mahdollista muuttaa tai tarkistaa itseensä liittyviä tietoja. Käyttöoikeuksia käyttäjällä ei ole oikeutta muuttaa. Loppukäyttäjän WWW-palvelu on samaan tapaan perlillä toteutettu kuin pääkäyttäjän palvelukin. Loppukäyttäjän palvelussa käyttäjätunnukset ja salasana haetaan SQL-kannasta, toisin kuin pääkäyttäjän palvelussa. Muuten autentikointi tapahtuu samaan tapaan kuin pääkäyttäjän palveluun, eli käytetään WWW-palvelimen omaa autentikointimekanismia. Autentikoinnin jälkeen avautuva ikkuna on esitetty kuvassa 6.6.

Avausikkunasta käyttäjä näkee omat tietonsa, sekä palveluihin olevat oikeudet. Oikeudet ovat pääkäyttäjän määrittelemät, eikä käyttäjä tällä sovelluksella pysty niitä muuttamaan. Kyseisestä ikkunasta käyttäjä näkee myös käyttäjätunnuksensa, jota hänen edellytetään käyttävän kyseiseen palveluun. Kuvassa 6.7 on yläkulmassa oleva hallintavalikko avattuna. Valikossa on operaatiot, joita käyttäjä voi valita.



Kuva 6.6: Loppukäyttäjän palveluiden pääikkuna



Kuva 6.7: Loppukäyttäjän palveluiden pääikkunan valikko

Kolmella ensimmäisellä vaihtoehdolla käyttäjä voi muuttaa itseensä liittyviä henkilötietoja, yritystietoja sekä vaihtaa salasanaa. Puhelujen siirron avulla käyttäjä voi ohjailla tulevia puheluita esimerkiksi matkapuhelimeensa. Siirtopalvelussa käyttäjän täytyy määritellä osoite johon tulevat INVITE-pyynnöt ohjataan. Siirrolle voidaan määritellä myös prioriteetti. Esimerkiksi siirto on voimassa vain silloin kuin käyttäjä ei ole rekisteröityneenä palveluun. Etärekisteröinnin ja soitonsiirron välillä on pieni ero. Etärekisteröinnissä käyttäjä kirjautuu SIP-palveluun WWW-lomakkeella, kuitenkin hänen täytyy määritellä aika, jonka rekisteröinti on voimassa. Siirto ei siten ole voimassa jatkuvasti, kuten soitonsiirrossa.

7 Yhteenveto

Autentikointi näyttelee tulevaisuudessa entistä suurempaa osaa. Internetissä käytettävät palvelut lisääntyvät, sekä palvelut muuttuvat entistä henkilökohtaisimmiksi. Esimerkiksi henkilötietojen tarkistukset ja muutokset tulevat tulevaisuudessa mahdollisiksi verkon välityksellä. Tämä asettaa erityisiä vaatimuksia suojaukselle. Suojauksen tarvetta lisää myös jatkuvasti yleistyvä langattomuus. Yhä useammin ihmiset haluavat käyttää sijainnista riippumatta palveluitaan. Kuitenkaan radiotietä ei voida koskaan pitää täysin luotettavana siirtomediana, jolloin salauksessa ja autentikoinnissa käytettävien protokollien suunnittelussa ei voida olettaa siirtotieltä mitään turvaa. Tällöin autentikointia joudutaan tekemään useammalla kerroksella, jolloin yhden palvelun käyttämiseen saatetaan tehdä useampia autentikointeja. Esimerkiksi 3G:ssä on autentikointi siirtotielle, mahdolliselle mobiilitoiminnallisuudelle sekä varsinaiselle IP-palvelulle. Tällöin on pakko pystyä yhdistämään autentikointimenetelmiä, koska muuten käyttäjätunnuskantojen hallinta muuttuu mahdottomaksi. Lisääntyneet tunnistukset näkyvät myös lähiverkoissa. Yhä useammin halutaan autentikointia ennen lähiverkkopalveluun päästämistä, ja autentikointia erikseen jokaiselle palvelulle. Esimerkiksi aluksi käyttäjä autentikoituu WLAN-palveluun. Autentikoinnin onnistuttua käyttäjä pääse lähiverkkoon, mutta sen lisäksi hänen täytyy autentikoitua erikseen SIP- tai WWW-palveluun.

Lisääntyvät palvelut aiheuttavat myös sen, että käyttäjälle halutaan sallia joitain palveluita, mutta ei kuitenkaan kaikkia palveluita. Tämän vuoksi jokaiseen palveluun täytyy olla oma autentikoituminen. Tällaiseen tilanteeseen pitää varautua myös käytettävällä autentikointiprotokollalla. Ei voida pelkästään olettaa, että käyttäjä käy aina autentikoitumassa

samalla protokollalla ja parametreilla, vaan protokollan täytyy pystyä kuljettamaan informaatio käyttäjän haluamasta palvelusta. RADIUS- ja Diameter-protokollat tukevat kyseistä toiminnallisuutta, joten se ennestään lisää niiden mahdollisuuksia yhdistävänä tekijänä.

Työn käytännöllisen osuuden tuloksena on yhdistetty verkko- ja VoIP-palvelun autentikointi WirLab-projektin testi- ja kehitysympäristössä. Käyttäjien hallinta tapahtuu WWW-pohjaisella hallintalomakkeella, jossa sekä ylläpitäjille että käyttäjille on oma hallintalomake. Tulevaisuudessa näihin lomakkeisiin voidaan yhdistää muitakin toiminnallisuuksia ja palveluita. Määrittelyjen vakiintuessa voidaan tietokannasta poistaa erillisiä tauluja yhdistämällä päällekkäisiä tauluja. Esimerkiksi SIP-autentikointi tehdään tulevaisuudessa RADIUS- tai Diameter-protokollan avulla käyttäen esimerkiksi EAP -protokollaa. Varsinkin 3G-verkot ovat tuomassa Diameteriä yleisempään käyttöön myös SIP-protokollalle. Koska EAP tarjoaa luotettavan ja laajennettavan tunnistuksen myös SIP-palvelulle, tulee se varmasti leviämään 3G-verkkojenkin ulkopuolelle, varsinkin kuin nykyisin SIP:ssä käytettävää HTTP Digest-tunnistamismenetelmää ei pidetä nykyisin kovinkaan luotettavana.

Lähdeluettelo

[1] Simpson *The Point-to-Point Protocol (PPP).* rfc1661.txt, Heinäkuu 1994

[2] Blunk, Vollbrecht, Aboba *Extensible Authentication Protocol (EAP).* draft-ietf-pppext-rfc2284bis-02.txt, Helmikuu 2002

[3] McGregor *The PPP Internet Protocol Control Protocol (IPCP).* rfc1332.txt, Toukokuu 1992

[4] Aboba, Simon *PPP EAP TLS Authentication Protocol.* rfc2716.txt, Lokakuu 1999

[5] Andersson, Josefsson, Zorn, Simon, Palekar *Protected EAP Protocol (PEAP).* draft-josefsson-pppext-eap-tls-eap-02.txt, Helmikuu 2002

[6] Funk, Blake-Wilson *EAP Tunneled TLS Authentication Protocol (EAP-TTLS).* draft-ietf-pppext-eap-ttls-01.txt, Helmikuu 2002

[7] Rigney, Willens, Rubens, Simpson *Remote Authentication Dial In User Services (RADIUS).* rfc2865.txt, Kesäkuu 2000

[8] Rigney *RADIUS Accounting.* rfc2866.txt, Maaliskuu 2002

[9] Calhoun, Arkko, Guttman, Zorn, Loughney *Diameter Base Protocol.* draft-ietf-aaa-diameter-09.txt, Maaliskuu 2002

[10] Rosenberg, Schulzrinne, Handley, Schooler *SIP: Session Initiation Protocol.* rfc2543.txt, Maaliskuu 1999

[11] Rosenberg, Schulzrinne, Camarillo, Johnston, Peterson, Sparks, Handley, Schooler *SIP: Session Initiation Protocol.* draft-ietf-sip-rfc2543bis-09.txt, Helmikuu 2002

[12] H.Haverinen. *EAP SIM Authentication.* draft-haverinen-pppext-eap-sim-03.txt, Helmikuu 2002

[13] Rigney, Willats, Calhoun *RADIUS Extensions.* rfc2869.txt, Kesäkuu 2000

[14] Blunk, Vollbrecht *PPP Extensible Authentication Protocol (EAP).* rfc2284.txt, Maaliskuu 1998

[15] Arkko, Haverinen *EAP AKA Authentication.* draft-arkko-pppext-eap-aka-03.txt, Helmikuu 2002

[16] Sterman, Sadolevsky, Schwartz, Williams *RADIUS Extensions for Digest Authentication.* draft-sterman-aaa-sip-00.txt, Marraskuu 2001

[17] Franks, Hallam-Baker, Hostetler, Lawrence, Leach, Luotonen, Stewart *HTTP Authentication: Basic and Digest Access Authentication.* rfc2617.txt, Kesäkuu 1999

[18] Arkko, Torvinen, Niemi *HTTP Authentication with EAP.* draft-torvinen-http-eap-

01.txt, Marraskuu 2001

[19] Potter, Zamick *PPP EAP MS-CHAP-V2 Authentication Protocol.* draft-potter-pppext-eap-mschap-01.txt, Tammikuu 2002

[20] James Carlson *PPP Design, Implementation, and Debugging, Second Edition.* ISBN 0-201-70053-0, Heinäkuu 2000

[21] IEEE 802.1 Working Group *Port-Based Network Access Control.* IEEE Std 802.1X-2001, Kesäkuu 2001

[22] 3GPP *3G Security, Network Domain Security, IP Network layer security (Release 5).* TS 33.210 V1.0.0, Joulukuu 2001

[23] 3GPP *3G Security, Security Architecture (Release 4).* TS 33.102 V4.3.0, Joulukuu 2001

[24] 3GPP *IP Multimedia Subsystem (IMS), Stage 2 (Release 5).* TS 23.228 V5.3.0, Tammikuu 2002

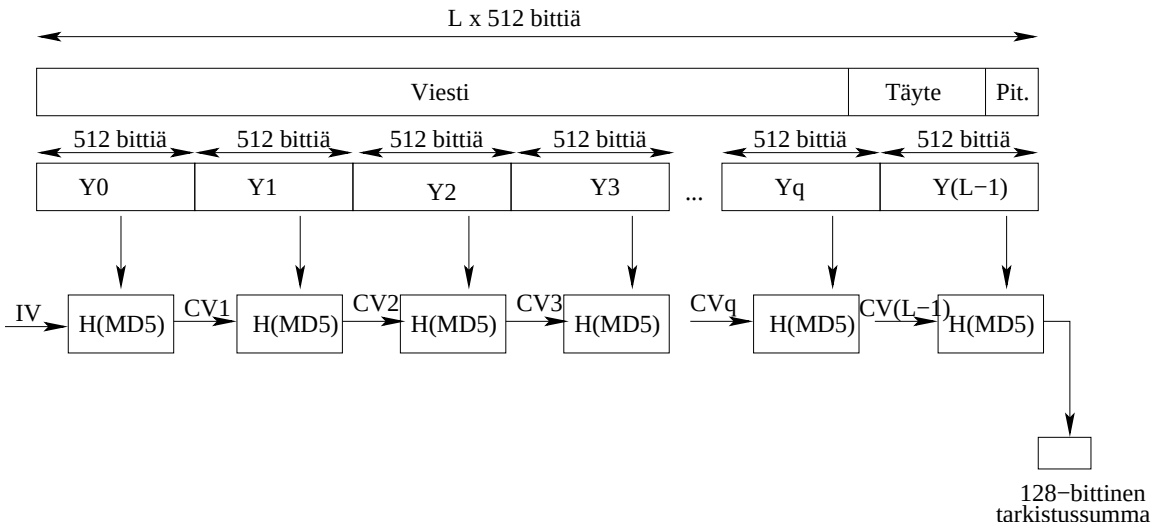
[25] 3GPP *Access security for IP-based services (Release 5).* TS 33.203 V1.0.0, Joulukuu 2001

[26] 3GPP *IP Multimedia (IM) Subsystem Cx Interface, Signalling flows and message contents (Release 5).* TS 29.228 V1.0.0, Joulukuu 2001

[27] 3GPP *Cx Interface based on Diameter protocol, Protocol details (Release 5).* TS 29.229 V1.0.0, Joulukuu 2001

[28] William Stallings *Cryptography and Network Security* 2. painos 1998

Liite A: MD5 -algoritmi

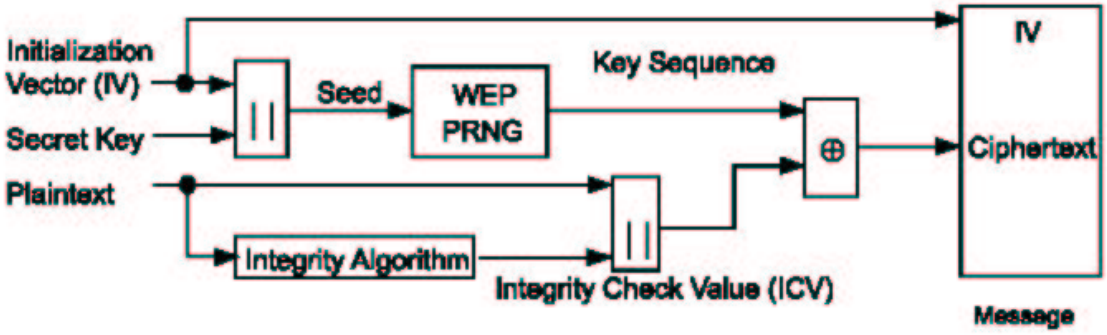


Kuva 7.1: MD5 -algoritmin lohkokaavio

Kuvassa 7.1 on esitetty MD5-algoritmin periaatteellinen lohkokaavio. Varsinaiseen MD5 -summan laskemiseen ei tässä mennä syvemmin, johtuen sen monimutkaisuudesta. Periaate MD5 -algoritmissä on kuvan kaltainen. Ensiksi viestiin laitetaan tarvittava määrä täytettä, jotta viesti on pituuskentän kanssa 512:lla jaollinen. Tämän jälkeen viesti paloitellaan 512 bitin lohkoihin, joista jokaisesta lasketaan erikseen tarkistussumma. Ensimmäinen alustusvektori IV, on aina MD5 -algoritmillä sama. Alustusvektorin ja syötteen avulla lasketaan neljässä vaiheessa 128 bittinen tarkistussumma, joka annetaan seuraavalle lohkolle alustusvektoriksi. Näin edetään kunnes viimeinen lohko on saatu käsiteltyä, jolloin siitä tuleva 128 bittinen tulos on lopullinen tarkistussumma.

H(MD5) -lohkoissa suoritetaan varsinainen tarkistussummien laskeminen. Jokaisessa lohkossa laskeminen on nelivaiheinen. Lisäksi tarkistussumman laskemiseen käytetään sinifunktiolla muodostettua 64 -elementtistä taulukkoa. Jokaisessa neljässä vaiheessa niistä käytetään 16, jolloin neljän vaiheen jälkeen jokaista elementtiä on käytetty. Kyseisen elementtitaulukon avulla syötteestä saadaan aina mahdollisimman satunnainen. Lisäksi neljässä vaiheessa syötettä pyöritetään niin paljon, että MD5 -algoritmiä voidaan pitää varsin luotettavana. Tarkemmin algoritmin toimintaa on esitetty esimerkiksi Williams Stallingsin Cryptoraphy and Network Security- kirjassa [28] sivulta 272 alkaen, sekä monissa muissa aiheeseen liittyvissä kirjoissa.

Liite B: WEP -algoritmi



Kuva 7.2: WEP -algoritmin lohkokaavio

Kuvassa 7.2 on esitetty WEP -algoritmin lohkokaavio. WEP -algoritmi käyttää RSA:n RC4 -algoritmiä, johon se muodostaa syötteen alustusvektorista ja salasanasta. RC4:lla saatu tulos summataan selkokielisen tekstin kanssa, ja näin muodostuu salattu teksti. Koska selkokielinen teksti ei kulje algoritmin läpi, on WEP -algoritmi melko heikko. Mahdollisen hyökkääjän tarvitsee ainoastaan selvittää RC4 -algoritmille mennyt syöte, jonka jälkeen liikenteen salauksen purkaminen on mahdollista. Jos hyökkääjä tietää jaetun salasanan, voi hän suoraan purkaa liikennettä, tai kaapattuaan tarpeeksi samalla alustusvektorilla muodostettuja salattuja tekstejä, hän voi arvata jaetun salasanan.