UNIVERZITA KOMENSKÉHO
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
KATEDRA INFORMATIKY

Bezpečnosť data-link layer v ISO OSI referenčnom modeli

DIPLOMOVÁ PRÁCA

Michal Ulacký

Odbor: Programové a Počítačové Systémy
Vedúci dipl. práce: RNDr. Andrej Bebják

Bratislava, 2006

Čestne prehlasujem, že túto diplomovú prácu som vypracoval samostatne a použil som iba literatúru a elektronické dokumenty uvedené v zozname na konci práce.

V Bratislave, Máj 2006

.....

Ďakujem svojmu diplomovému vedúcemu RNDr. Andrejovi Bebjákovi za cenné rady, a pomoc pri písaní diplomovej práce. Taktiež chcem poďakovať svojim blízkym za podporu a rady.

Abstrakt

Diplomová práca sa zaoberá bezpečnosťou počítačových sietí na úrovni data-link layer. Podáva prehľad súčasného stavu problematiky (štandardy, používané technológie, konkrétne príklady). Bol navrhnutý a popísaný vlastný model secure data-link layer a odhadnuté jeho vlastnosti.

Kľúčové slová. data-link, bezpečnosť, počítačové siete, model, šifrovanie

Obsah

Úvod ... 11
1 Bezpečnosť data-link layer v súčasnosti ... 12
1.1 IEEE 802.10 Standard for Interoperable LAN/MAN Security (SILS) ... 13
1.1.1 Clear header ... 14
1.1.1.1 SDE Designator ... 14
1.1.1.2 SAID ... 14
1.1.1.3 MDF ... 15
1.1.2 Protected header ... 15
1.1.2.1 Station ID ... 15
1.1.2.2 Flags ... 15
1.1.2.3 Fragment Identifier ... 15
1.1.2.4 Security Label ... 15
1.1.3 SDE SDU ... 15
1.1.4 PAD ... 15
1.1.4.1 Padding ... 16
1.1.4.2 Pad Length ... 16
1.1.5 ICV ... 16
1.2 IEEE 802.1X Port-Based Network Access Control ... 17
1.2.1 Enkapsulácia EAP v sieťach LAN ... 18
1.2.2 Port Access Control ... 21
1.3 Zabezpečenie bezdrôtových sietí ... 22
1.3.1 WEP ... 22
1.3.2 TKIP ... 23
1.3.3 CCMP ... 24
1.3.4 Autentifikácia ... 25
1.3.4.1 Open System ... 25
1.3.4.2 Shared Key ... 25
1.3.4.3 Service Set Identifier ... 25
1.3.4.4 MAC Filtering ... 25
1.3.4.5 UAM ... 26
1.3.4.6 WPA-PSK ... 26
1.3.4.7 EAP ... 26
1.3.4.7.1 LEAP ... 26
1.3.4.7.2 EAP-TLS ... 27
1.3.4.7.3 EAP-TTLS a PEAP ... 27
1.3.4.7.4 EAP-FAST ... 27
1.3.5 Útoky na Bezdrôtové siete ... 28
1.3.5.1 MAC podvrstva data-link layer ... 28
1.3.5.2 WEP ... 28
1.3.5.3 WPA-PSK ... 28
1.3.5.4 EAP ... 28
1.4 Konkrétny príklad neverejnej siete so zabezpečenou data-link vrstvou ... 29
Polytechnic University of Catalonia. (U.P.C.), Barcelona ... 29
2 Model ... 33
2.1 Zhrnutie ... 35
3 Štruktúra framu ... 37
3.1 Podrobný popis jednotlivých polí ... 38

3.1.1 Typ šifrovacieho algoritmu ... 38
3.1.2 Dĺžka adresy ... 39
3.1.3 Adresa príjemcu a Adresa odosielateľa ... 40
3.1.4 Session ID ... 40
3.1.5 Frame length ... 41
3.1.6 Typ framu ... 41
3.1.7 Sekvenčné číslo framu ... 42
3.1.8 Session ID ... 42
3.1.9 Timestamp ... 42
3.1.10 Data Length ... 42
3.1.11 Flags/Reserved ... 42
3.1.12 Data ... 43
3.1.13 Padding ... 43
3.1.14 Podpis ... 43
4 Čo šifrovať ... 44
4.1 Hlavička ... 44
4.1.1 Nešifrovaná hlavička ... 44
4.1.1.1 Adresa odosielateľa a príjemcu ... 44
4.1.1.2 Typ šifrovacíeho algoritmu ... 44
4.1.1.3 Session ID ... 45
4.1.1.4 Frame Length ... 45
4.1.2 Šifrovaná hlavička ... 45
4.1.2.1 Typ framu ... 45
4.1.2.2 Sekvenčné číslo ... 45
4.1.2.3 Polia udávajúce dĺžky ... 45
4.1.2.4 Flags/reserved ... 45
4.1.2.5 Kópia Session ID, timestamp ... 45
4.2 Dáta a výplň ... 45
4.3 Digitálny podpis ... 46
5 Vysielanie viacerým adresátom ... 48
5.1 Broadcast ... 48
5.2 Multicast ... 48
6 Práca uzla ... 50
6.1 Centrálny uzol ... 50
6.2 Činnosť bežného uzla ... 52
6.2.1 Odosielanie framov ... 52
6.2.2 Prijatie framu ... 53
6.2.2.0 Dátový frame ... 56
6.2.2.1 Začiatok novej komunikácie ... 57
6.2.2.2 Handshake challenge ... 57
6.2.2.3 Handshake response ... 58
6.2.2.4 Handshake accept ... 59
6.2.2.5 Handshake reject ... 59
6.2.2.6 Žiadosť o nový Session Key ... 59
6.2.2.7 Odpoveď na žiadosť o nový Session Key ... 60
6.2.2.8 Zamietnutie žiadosti o nový Session Key ... 61
6.2.2.9 Potvrdenie nového Session Key ... 61
6.2.2.10 Žiadosť o zrušenie spojenia ... 62
6.2.2.11 Odpoveď na žiadosť o zrušenie spojenia ... 62
6.2.2.12 Keepalive ... 63

6.2.2.13 Resend request ... 63
7 Efektivita ... 64
7.1 Dátová efektivita ... 64
7.2 Výpočtová náročnosť ... 65
8 Záver ... 70
9 Zoznam použitej literatúry ... 71
Príloha A OSI model ... 73
A.1 Physical Layer ... 74
A.2 Data-link Layer ... 74
A.2.1 Media Access Control Sublayer ... 75
A.2.2 Logical Link Control Sublayer ... 75
A.3 Network Layer ... 75
A.4 Transport Layer ... 75
A.5 Session Layer ... 75
A.6 Presentation Layer ... 76
A.7 Application Layer ... 76
Príloha B Šifrovanie ... 77
B.1 Šifrovanie ... 77
B.1.1 Úvod ... 77
B.1.2 Symetrické šifry ... 77
B.1.2.1 Blokové ... 77
B.1.2.2 Prúdové ... 79
B.1.3 Asymetrické šifry ... 80
B.1.4 Zhrnutie ... 81
B.2 Elektronické podpisy ... 81
B.2.1 Úvod ... 81
B.2.2 Hašovanie ... 82
B.2.2.1 Kryptografické hašovacie funkcie ... 82
B.2.2.2 Message Authentication Code ... 82
B.2.3 Digitálne podpisy ... 83
B.2.4 Digitálne certifikáty ... 83

Zoznam obrázkov

Obr. 1 PTK 4-way handshake ... 23
Obr. 2 Štruktúra zabezpečeného framu ... 31
Obr. 3 Štruktúra administračného framu ... 32
Obr. 4 challenge - response ... 33
Obr. 5 Preklad adries pomocou TTP ... 34
Obr. 6 2 adresné módy ... 35
Obr. 7 Typ algoritmu a dĺžka adresy ... 37
Obr. 8 Štruktúra framu ... 38
Obr. 9a Typ algoritmu - oddelené tabuľky ... 39
Obr. 9b Typ algoritmu – spojené tabuľky ... 39
Obr. 10a Dĺžka adresy - tabuľka ... 39
Obr. 10b Dĺžka adresy - hodnota ... 40
Obr. 11 Session ID unikátne vzhľadom na oba uzly ... 41
Obr. 12a TTP – nepriame nadviazanie komunikácie ... 51
Obr. 12b TTP – priame nadviazanie komunikácie ... 51

Obr. 13 Nekonečná slučka ... 52
Obr. 14 Loop ... 52
Obr. 15 Odoslať ... 53
Obr. 16 Prijať ... 55
Obr. 17 Overiť timestamp ... 55
Obr. 18 Spracovať podľa typu framu ... 56
Obr. 19 Dátový ... 56
Obr. 20 Nová komunikácia ... 57
Obr. 21 Handshake challenge ... 58
Obr. 22 Handshake response ... 58
Obr. 23 Handshake accept ... 59
Obr. 24 Handshake reject ... 59
Obr. 25 Nový Session Key ... 60
Obr. 26 Odpoveď na nový Session Key ... 60
Obr. 27 Zamietnutie nového Session Key ... 61
Obr. 28 Potvrdenie nového Session Key ... 61
Obr. 29 Zrušenie spojenia ... 62
Obr. 30 Odpoveď na zrušenie spojenia ... 62
Obr. 31 Keepalive ... 63
Obr. 32 Resend request ... 63
Obr. 33 Efektivita dátového prenosu ... 65
Obr. 34 n-tá vrstva vrstvového modelu ... 73
Obr. 35 OSI model - komunikácia ... 74

Zoznam tabuliek

Tab. 1 SDE PDU ... 13
Tab. 2 Formát skupinovej SAID ... 14
Tab. 3a EAPOL – 802.3/Ethernet ... 18
Tab. 3b EAPOL – Token Ring/FDDI ... 18
Tab. 4 RC4 Key Descriptor ... 20
Tab. 5 EAP Packet ... 21
Tab. 6 veľkosti MTU pre rôzne protokoly ... 34
Tab. 7 Typy framov ... 41
Tab. 8 Timestamp ... 42
Tab. 9 rýchlosť hašovacích funkcií ... 66
Tab. 10 rýchlosť symetrických prúdových šifier ... 66
Tab. 11 rýchlosť symetrických blokových šifier ... 67
Tab. 12 rýchlosť dešifrovania asymetrických šifier ... 67
Tab. 13 rýchlosť šifrovania asymetrických šifier ... 68
Tab. 14 rýchlosť digitálneho podpisovania ... 68
Tab. 15 rýchlosť overovania digitálneho podpisu ... 69
Tab. 16 OSI vs. TCP/IP ... 73

Zoznam skratiek

AAA	Authentication, Authorization and Accounting
AES	Advanced Encryption Standard
AP	Access Point
ARP	Address Resolution Protocol
CBC	Cipher-Block Chaining
CBC-MAC	CipherBlock Chaining with Message Authentication Code
CCMP	CTR with CBC-MAC Protocol
CFB	Cipher FeedBack
CRL	Certificate Revocation List
CTR	CounTeR mode
DHCP	Dynamic Host Configuration Protocol
DoS	Denial of Service
DSAP	Destination Service Access Port
EAP	Extensible Authentication Protocol
EAP-FAST	EAP Flexible Authentication via Secure Tunneling
EAPOL	EAP Over LAN
EAP-TTLS	EAP Tunneled TLS
ECB	Electronic CodeBook
ESSID	Extended Service Set Identifier
FCS	Frame Check Sequence
ICV	Integrity Check Value
IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
ISO	International Standardization Organization
IV	Initialization Vector
LAN	Local Area Network
LEAP	Lightweight EAP
LLC	Logical Link Control
MAC	Media Access Control
MAC	Message Authentication Code
MAN	Metropolitan Area Network
MDF	Management-Defined Field
MIC	Message Integrity Code
MTU	Maximum Transmission Unit
NTP	Network Time Protocol
OFB	Output FeedBack
OSI	Open Systems Interconnection
PAE	Port Access Entity
PDU	Protocol Data Unit
PEAP	Protected EAP
PKI	Public Key Infrastructure
PMK	Pair-wise Masterk Key
PTK	Pair-wise Transient Key
SAID	Security Association ID
SDE	Secure Data Exchange
SDU	Service Data Unit
SMIB	Security Management Information Base

SNAP	SubNetwork Access Protocol
SSAP	Source Service Access Port
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
TLS	Transport Layer Security
TTP	Trusted Third Party
UAM	Universal Access Method
WEP	Wired Equivalent Privacy
Wi-Fi	Bezdrôtové siete založené na špecifikácii IEEE 802.11
WPA	Wi-Fi Protected Access
WPA-PSK	Wi-Fi Protected Access - Pre-Shared Key
XOR	eXclusive OR (súčet modulo 2)

Úvod

S nástupom bezdrôtových sietí začala byť v centre pozornosti potreba bezpečnej data-link vrstvy. V prípade sietí založených na kábloch bol pre útočníka potrebný fyzický prístup buď ku kabeláži, alebo k sieťovým zariadeniam a teda väčšinou dochádzalo k útokom z vonka a na vyšších vrstvách (najmä network layer). Preto sa bezpečnosti data-link layer nevenovala takmer žiadna pozornosť. S výrazným rozšírením bezdrôtových sietí sa to ale zmenilo, keďže prenosovým médiom je vzduch a teda na prístup do siete stačí byť v oblasti vysielania. Sieťoví administrátori neboli zvyknutí na takéto podmienky a preto bola (a stále je) veľká väčšina bezdrôtových sietí chránená veľmi slabo, resp. vôbec. Najmä v oblasti bezdrôtových sietí sa začali objavovať bezpečnostné prvky, ale stále sú nepovinné a preto problémy pretrvávajú. Navyše šifrovanie prvých štandardov sa už podarilo prekonať.

Táto práca podáva prehľad súčasného stavu problematiky a navrhuje všeobecný model. Model je určený pre ľubovoľný typ siete (kabeláž/wireless, point-to-point/bus, s centrálnym uzlom/bez neho) a bezpečnosť je v ňom povinná. Táto práca neurčuje použitie konkrétnych kryptografických algoritmov, tie sa určujú pri implementácii. Navrhovaný model je preto škálovateľný a rozšíriteľný v budúcnosti.

V prvej kapitole sa pozrieme na súčasný stav problematiky. Rozoberáme tu dva v súčasnosti používané štandardy, IEEE 802.1X (riadenie prístupu) a IEEE 802.10 (ochrana dát), zvlášť sa venujeme bezdrôtovým sieťam a na koniec uvádzame konkrétny príklad secure data-link layer použitý v praxi.

Vlastnosti modelu načrtneme v druhej kapitole a v ďalších kapitolách ich budeme spresňovať. V tretej kapitole rozoberieme štruktúru framu a v štvrtej zhodnotíme, ktoré údaje treba šifrovať. Piata kapitola rozoberá detaily vysielania viacerým príjemcom a šiesta sa pozerá na postup, ktorým bude pracovať uzol. V siedmej kapitole odhadneme efektivitu prenosu dát a náročnosť nášho modelu z hľadiska hardware.

Na správne pochopenie tohoto textu je potrebné mať aspoň základné znalosti zo šifrovania a počítačových sietí. V prílohe A sa nachádza krátky úvod do OSI modelu a v prílohe B je úvod do šifrovania a digitálnych podpisov. Čitateľom, ktorí nie sú celkom zbehlí v problematike, ich odporúčame prečítať.

1 Bezpečnosť data-link layer v súčasnosti

V súčasnosti sa nekladie dôraz na zabezpečenie data-link layer. S výnimkou konkrétnych neverejných sietí a sietí typu Wi-Fi sa prakticky nepoužíva žiadne zabezpečenie. Bezpečnostnými otázkami sa zaoberá skôr na Network alebo Transport layer (IPSec, SSL, TLS, ...).

Toto má niekoľko príčin. Podľa [4] sa neodporúča do budúcich aplikácií šifrovanie na druhej vrstve (data-link). Podľa tohoto dokumentu, šifrovanie na úrovni druhej vrstvy neprináša žiadne výhody oproti šifrovaniu na prvej (physical), alebo tretej (network) vrstve.

Ďalej väčšina útokov sa neviedla využitím chýb na druhej vrstve, ale na vyšších vrstvách, takže z toho nevyplývala potreba chrániť druhú vrstvu. A takisto s výnimkou bezdrôtovej komunikácie je odchytávanie komunikácie dosť zložité, takže až do masovejšieho rozšírenia bezdrôtových sietí nebola veľká potreba zabezpečovať komunikáciu na data-link vrstve.

Zabezpečením data-link vrstvy sa v súčasnosti zaoberajú štandardy ISO 7498-2, IEEE 802.1X a IEEE 802.10

1.1 IEEE 802.10 Standard for Interoperable LAN/MAN Security (SILS)

Tento štandard sa zaoberá zabezpečením dát prenášaných data-link vrstvou. Je založený na Secure Data Exchange (SDE), ktorý je súčasť LLC podvrstvy. Zabezpečuje:

- utajenosť dát

  pomocou zašifrovania údajov
- integritu dát

  pomocou vypočítania Integrity Check Value (ICV), ktorý sa odošle spolu s dátami. je to vlastne forma digitálneho podpisu
- autentickosť odosielateľa

  pomocou umiestnenia adresy odosielateľa k dátam ktoré sa budú šifrovať
- kontrolu prístupu (access control)

  pomocou správy kľúčov, systému, alebo označovania údajov. Je závislá od integrity a autentickosti

Akékoľvek údaje, ktoré sa zašifrujú spolu s dátami zaručujú, že ich zapísal subjekt poznajúci šifrovací kľúč. Toto sa využíva pri ICV, určení autentickosti odosielateľa a kontrole prístupu.

Všeobecne tento štandard závisí od externej správy kľúčov a voľby šifrovacieho algoritmu. Protokol je transparentný a umožňuje koexistenciu zabezpečenej aj nezabezpečenej komunikácie. To znamená, že uzly, komunikujúce prostredníctvom 802.10 vysielajú špeciálny typ framov. Uzly nekomunikujúce zabezpečene tieto framy ignorujú, pretože nespĺňajú štruktúru bežného framu. Ak uzol prijme bežný frame, spracuje ho ako bežný, ak prijme frame 802.10, spracuje ho podľa tohoto štandardu. Správa kľúčov je externá, takže tento štandard sa nezaoberá dohadovaním kľúčov medzi uzlami, ale predpokladá existenciu SMIB, z ktorej čerpá potrebné dáta.

SDE PDU													
Clear header			Protected header				Data (SDE SDU)				PAD		ICV
SDE Designator	SAID	MDF	Station ID	Flags	Fragment Identifier	Security Label	DSAP	SSAP	Control	Data	Padding	Pad Length	
3	4	<=20	8	1	4	<=247	>=1				<=255	1	>=1

Tab. 1 SDE PDU

Zašifrované sú polia v Protected header, Data, Pad a ICV. ICV sa počíta z polí Protected header, Data a Pad.

1.1.1 Clear header

Identifikuje SDE PDU a pomáha pri spracovaní informácií v nich obsiahnutých. Obsah clear header je určený raz pri nadviazaní spojenia a ďalej sa nemení. Clear header nie je povinná, ale pri niektorých špecifických zariadeniach môžu vzniknúť problémy ak sa nepoužije.

1.1.1.1 SDE Designator

Zabezpečuje, aby SDE PDU nebolo spracované zariadením nepodporujúcim SDE. Podľa špecifikácie $(0A\ 0A\ 03)_{hex}$

1.1.1.2 SAID

Security Association ID, identifikuje použitú bezpečnostnú asociáciu prímateľa. Pokiaľ je prímateľ skupina, majú všetky uzly v tejto skupine rovnakú SAID a tá je dohodnutá vopred pomocou managementu kľúčov alebo systému.

G-bit	S1-bit	S2-bit	ID bity	
0	Individuálna SAID			
1	1	10 bitov	20 bitov	
1	0	1	17 bitov	12 bitov
1	0	0	21 bitov	8 bitov

Tab. 2 Formát skupinovej SAID

Prvý bit (G-bit) určuje, či sa jedná o individuálnu SAID (0), alebo skupinovú (1). Na inicializáciu (tzv. bootstrap SAID) sa používajú 4 vyhradené hodnoty. Pre management kľúčov sú všetky bity za G-bitom nastavené na 0. Pre management systému sú nastavené na 1. Tento spôsob managementu nie je povinný, dá sa nahradiť komunikáciou so správnou adresou.

Ak je adresát skupina, existujú 3 rôzne triedy SAID, v závislosti od maximálneho počtu prideliteľných SAID. Prvá trieda umožňuje 1 024 managementov a každý spravuje 1 048 576 SAID, druhá trieda 131 072 managementov po 4 096 SAID a tretia trieda 2 097 152 managementov po 256 SAID (podobný princíp ako triedy IP adries). Jeden identifikátor z tretej triedy, $(800001)_{hex}$, je rezervovaný pre izolované siete a testovanie. Registračná autorita pre prideľovanie týchto identifikátorov je IEEE.

1.1.1.3 MDF

Management-Defined Field, umožňuje prenášať doplnkové informácie. Nepovinné pole s variabilnou dĺžkou (0-20 octetov)

1.1.2 Protected header

Chránená hlavička, nepovinná.

1.1.2.1 Station ID

Jednoznačná identifikácia odosielateľa. Dĺžka je 8 octetov, od začiatku obsahujú MAC adresu odosielateľa, zvyšné octety sú nedefinované.

1.1.2.2 Flags

Používa sa najmä pri fragmentácii. Posledné dva bity tohoto poľa sú More segments a Fragmented. Ak fragmented je true, tak SDE PDU je len časť SDE SDU a že pole Fragment Identifier je prítomné. More segments sa používa na číslovanie fragmentov, ak je true, tak SDE PDU je prvý fragment SDE SDU, inak je to druhý fragment.

1.1.2.3 Fragment Identifier

Používa sa pri fragmentácii. Určuje z ktorého SDE SDU bol odvodený tento SDE PDU.

1.1.2.4 Security Label

Obsahuje množinu tzv. Security Tag, ktoré obsahujú rôzne dáta používané napr. na vytvorenie rôznych hierarchií bezpečnosti.

1.1.3 SDE SDU

Dátová časť, väčšinou priamo prevzatá z vyššej vrstvy/podvrstvy. Ako príklad sa uvádza LLC PDU:

DSAP	Destination Service Access Port, adresa adresáta.
SSAP	Source Service Access Port, adresa odosielateľa.
Control	Riadiace údaje.
Data	Samotné dáta.

1.1.4 PAD

Výplň, používa sa v niektorých algoritmoch na zvýšenie bezpečnosti, alebo integrity. Nepovinné pole.

1.1.4.1 Padding

Samotná výplň. Maximálna dĺžka je 255 octetov. Dĺžka je celé číslo, takže nie je možné padding použiť na kompenzáciu problémov so zarovnaním octetov (spôsobenými niektorými algoritmami na integritu alebo šifrovanie)

1.1.4.2 Pad Length

Dĺžka Padding časti, pokiaľ sa padding používa, táto položka je povinná. Do dĺžky sa nezapočítava jeden octet potrebný pre Pad Length. Pokiaľ sa nepoužíva ICV, je to posledný octet SDE PDU, v opačnom prípade je to octet hneď pred ICV.

1.1.5 ICV

Mechanizmus na zabezpečenie integrity dát. ICV sa počíta z Protected Header, Data a Pad. Dĺžka je závislá od konkrétnej asociácie. Nepovinné pole.

1.2 IEEE 802.1X Port-Based Network Access Control

Tento štandard sa zaoberá zabezpečením komunikácie v sieťach LAN, konkrétne riadi prístup k jednotlivým portom (rozhranie medzi systémom a sieťou). Predpokladajú sa spojenia typu point-to-point. Každý port môže mať jednu z nasledujúcich rolí

1. **Authenticator**
   port, ktorý vyžaduje overenie druhej strany pred tým, než jej sprístupní svoje služby
2. **Supplicant**
   port, ktorý chce prístup k službám authenticatora
3. **Authentication server**
   poskytuje služby overenia totožnosti a autorizácie

Jeden systém (reprezentovaný v sieti svojim portom) môže nadobudnúť aj viac z týchto rolí. Napríklad môže byť zároveň authenticator aj authentication server (bez nutnosti komunikovať s externým servrom), prípadne môže pre jednu službu byť authenticator a pre inú supplicant.

Časť systému zaoberajúca sa protokolom tohoto štandardu sa nazýva PAE - Port Access Entity. Supplicant PAE je zodpovedná za reagovanie na požiadavky Authenticatora pre informácie overujúce totožnosť. Authenticator PAE je zodpovedná za komunikáciu so supplicantom, za odoslanie obdržaných informácií authentication serveru a za prijatie výsledku autorizácie. Podľa výsledku potom nastavuje stav svojho portu na autorizovaný/neautorizovaný.

Každý port je logicky rozdelený na kontrolovaný port a nekontrolovaný port. Kontrolovaný port poskytuje svoje služby len autorizovaným systémom, nekontrolovaný port aj neautorizovaným systémom. Každé PDU ktoré dorazí na port je sprístupnené aj kontrolovanému, aj nekontrolovanému logickému portu.

Podľa nastavenia portu môže byť kontrolovaný port v jednom z týchto stavov: vždy neautorizovaný, vždy autorizovaný a auto. Auto znamená, že na autorizáciu je potrebné získať súhlas od authorization servera.

Systém nemá prístup k sieti, pokiaľ je fyzické pripojenie odpojené (či už fyzicky, alebo administratívne). V takomto prípade sa automaticky nastaví kontrolovaný port na neautorizovaný.

Počiatočná komunikácia medzi authenticator-om a supplicantom prebieha cez nekontrolované porty, komunikácia medzi authenticatorom a authentication servrom môže prebiehať cez jeden alebo viac kontrolovaných alebo nekontrolovaných portov (táto komunikácia je mimo štandardu 802.1X a väčšinou prebieha prostredníctvom EAP na vyšších vrstvách).

Kontrolovaný port môže pracovať v jednom z dvoch módov. Jednosmerný mód kontroluje len prichádzajúcu komunikáciu, obojsmerný mód kontroluje aj prichádzajúcu, aj odchádzajúcu komunikáciu. (príklad využitia jednosmerného módu je Wake-on-LAN)

1.2.1 Enkapsulácia EAP v sieťach LAN

802.3/Ethernet				
PAE Ethernet Type	**Protocol Version**	**Packet Type**	**Packet Body Length**	**Packet Body**
1-2	3	4	5-6	7-N

Tab. 3a EAPOL – 802.3/Ethernet

Token Ring/FDDI				
SNAP-encoded Ethernet Type	**Protocol Version**	**Packet Type**	**Packet Body Length**	**Packet Body**
1-8	9	10	11-12	13-N

Tab. 3b EAPOL – Token Ring/FDDI

PAE Ethernet Type

Obsahuje pridelenú hodnotu pre Ethernet Type. Podľa tohoto štandardu. je to hodnota z rozsahu $(88\text{-}8E)_{hex}$.

SNAP-encoded Ethernet Type

Obsahuje pridelenú hodnotu pre Ethernet Type zakódovanú SNAP.

Prvé tri octety obsahujú štandardnú SNAP hlavičku $(AA\ AA\ 03)_{hex}$, ďalšie tri octety obsahujú SNAP PID, hodnota $(00\ 00\ 00)_{hex}$ a zvyšné dva octety obsahujú PAE Ethernet Type.

Protocol Version

Obsahuje verziu EAPOL protokolu podporovanú odosielateľom tohoto framu. podľa tejto špecifikácie bude obsahovať hodnotu $(0000\ 0001)_2$.

Packet Type

a. EAP-Packet - $(0000\ 0000)_2$

b. EAPOL-Start - $(0000\ 0001)_2$

c. EAPOL-Logoff - $(0000\ 0010)_2$

d. EAPOL-Key - $(0000\ 0011)_2$

e. EAPOL-Encapsulated-ASF-Alert - $(0000\ 0100)_2$

EAPOL-Key sa používa iba ak je podporovaná možnosť posielania kľúčov medzi autenthicatorom a supplicantom.

EAPOL-Encapsulated-ASF-Alert sa používa na posielanie ASF správ (konkrétne sa jedná o "alert-y" používané napr. pri SNMP) cez neautorizované porty. Všetky framy označené takto budú postúpené entite protokolu zodpovednej za spracovanie ASF. Táto časť je už mimo tejto špecifikácie.

Packet Body Length

Dĺžka tela paketu; neznamienkové celé číslo, ak sa rovná nule, paket nemá telo (EAPOL-Start, EAPOL-Logoff, prípadne nedefinované hodnoty packet type).

Packet Body

Samotné telo paketu (ak packet type je EAP-Packet, EAPOL-Key alebo EAPOL-Encapsulated-ASF-Alert. v opačnom prípade táto časť neexistuje). Enkapsulovaný je práve jeden paket/key descriptor/ASF alert.

Každý prijatý frame musí byť pred spracovaním najprv validovaný. Overuje sa cieľová MAC adresa (musí byť PAE group address, 01-80-C2-00-00-03 (je to rezervovaná MAC adresa, ktorú MAC bridge nepreposielajú ďalej), v sieťach s nezdieľaným médiom, alebo špecifická MAC adresa PAE v sieľach so zdieľaným médiom), pole Ethernet Type obsahuje správnu hodnotu $(88\ -\ 8E)_{hex}$ a pole Packet Type obsahuje jednu z hodnôt EAP-Packet, EAPOL-Start, EAPOL-Logoff alebo EAPOL-Key. Pokiaľ aspoň jedna z týchto troch podmienok neplatí, frame nebude spracovaný. Pokiaľ packet type je EAPOL-Start, alebo EAPOL-Logoff, akékoľvek octety nasledujúce po packet type v tomto frame sú ignorované. Ak je packet type EAP-Packet, alebo EAPOL-Key, akékoľvek octety nasledujúce po packet body v tomto frame sú ignorované.

Aby bola zabezpečená spätná kompatibilita medzi verziami tohoto protokolu, v prípade prijatia framu s vyššou verziou protokolu sa tento frame bude spracúvať ako by mal rovnakú verziu protokolu ako príjemca. Všetky nedefinované parametre sú

ignorované, tak isto ako aj všetky octety nasledujúce za posledným definovaným octetom v spracúvanej verzii. V prípade prijatia framu s nižšou verziou protokolu sa bude frame spracúvať touto nižšou verziou.

Key Descriptor format		
Descriptor Type	1	Typ deskriptora, určuje interpretáciu ostatných polí. Hodnota 1 – RC4 Key Descriptor
Key Length	2-3	Dĺžka kľúča v octetoch
Replay Counter	4-11	Počítadlo, používané na odhalenie a zamedzenie replay útokov
Key IV	12-27	Inicializačná hodnota kľúča, 128 bitov náhodných dát
Key Index	28	7 bitové neznamienkové celé číslo a 1 bit flag. Flag 0 je broadcastovanie kľúča, flag 1 unicast. Číslo znamená číslo kľúča v prípade množiny kľúčov, je určované authenticatorom
Key Signature	29-44	Podpis všetkých polí EAPOL paketu od Protocol Version po Encrypted Key vrátane, pričom pri podpisovaní sa pole key signature berie ako nulové
Key	45-Packet body length	Nepovinné pole, ak chýba, supplicant použije peer key vygenerovaný počas EAP autentifikačného procesu ako kľúč pre túto správu. Ak kľúč je dlhší ako špecifikovaná dĺžka kľúča, zvyšok sa ignoruje

Tab. 4 RC4 Key Descriptor

Replay counter obsahuje NTP čas, Key IV obsahuje náhodnú hodnotu použitú na vygenerovanie enkrypčného kľúča RC4, Signature obsahuje podpis typu HMAC-MD5, použitý kľúč je server key vygenerovaný pri EAP autentifikácie. Na enkrypciu poľa Key sa použije RC4. Kľúč sa vytvorí zreťazením Key IV a session key vygenerovaného EAP autentifikáciou.

EAP Packet		
Code	1	Typ EAP paketu. 1- request, 2- response, 3- success, 4- failure
Identifier	2	Identifikátor používaný na párovanie requestov a responsov. Spolu s Port-om tvorí jednoznačnú identifikáciu autentifikácie
Length	3-4	Dĺžka celého paketu
Data	5-N	Dáta, ich formát je určený poľom Code

Tab. 5 EAP Packet

1.2.2 Port Access Control

Proces využíva EAPOL ako spôsob výmeny autentifikačných informácií medzi supplicantom a authentication servrom. EAP umožňuje výber autentifikačného mechanizmu na základe získaných informácií - Request paket obsahuje typové pole. Typický postup je, že authenticator vyšle Identity Request nasledovaný jedným alebo viac Requestmi pre ďalšie informácie (aj keď Identity Request môže byť vynechaný, v prípade, že identita supplicanta je vopred predpokladaná). Supplicant odpovie na každý Request Response paketom, ktorý taktiež obsahuje typové pole, ktoré korešponduje s typovým poľom Requestu. Výmena je ukončená Acceptom alebo Rejectom od Authentication Servra (môže obsahovať enkapsulovaný EAP paket).

Autentifikácia môže byť iniciovaná buď Authenticatorom, alebo Supplicantom. Typicky, ak Authenticator zistí zmenu stavu MAC daného portu z disabled na enabled, zaháji autentifikáciu. Pokiaľ neobdrží odpoveď v určenom čase, nastáva retransmisia. Pokiaľ nie je známa identita Supplicanta, vyšle sa EAP-Request/Identity frame. Authenticator môže podporovať opakovanú reautentifikáciu a môže ju požadovať v ľubovoľnom čase (napríklad po reinicializácii Authenticatora). Pokiaľ reautentifikácia zlyhá, port sa prepne do neautorizovaného stavu. Iniciáciu môže vykonať taktiež aj Supplicant - vyslaním EAPOL-Start paketu.

Keď Supplicant chce, aby Authenticator prepol stav portu na neautorizovaný, vyšle EAPOL-Logoff (používa sa napr. pri odhlásení užívateľa, rekonfigurácii atď.).

Authenticator PAE sa stará o preposielanie EAP framov medzi Supplicantom a Authentication Servrom. Framy EAPOL-Start, EAPOL-Logoff, EAPOL-Key a EAP-Request/Identity nebudú preposielané, keďže je to zbytočné. Všetky ostatné EAP framy prijaté od Supplicanta budú preposlané Authentication Servru a naopak, všetky prijaté EAP framy od Authentication Servra budú preposlané Supplicantovi.

1.3 Zabezpečenie bezdrôtových sietí

Najväčší dôraz na zabezpečenie druhej vrstvy sa kladie pri bezdrôtových sieťach. Dôvod je jednoduchý: zatiaľ čo pri klasických sieťach je potrebné byť fyzicky pripojený na sieť (či už na kábel - médium, alebo do nejakého hardwarového prvku siete - hub, switch, router), pri bezdrôtových sieťach stačí byť v dosahu vysielania, keďže médium je vzduch. Takže fyzický prístup sa omnoho horšie kontroluje ako pri bežných sieťach (niekedy je kontrola nemožná, napr. na verejných priestranstvách). Po masovejšom rozšírení Wi-Fi sietí začali byť problémy s bezpečnosťou veľmi naliehavé a postupne začali vznikať rôzne bezpečnostné prvky. Prvý bezpečnostný protokol bol WEP (Wired Equivalent Privacy), ktorý sa ale ukázal ako nedostatočný. Neskôr bol nahradený protokolom WPA, ktorý vychádza z TKIP (Temporal Key Integrity Protocol) a CCMP (CTR (CounTeR mode) with CBC-MAC (CipherBlock Chaining with Message Authentication Code) Protocol).

1.3.1 WEP

WEP je bezpečnostný protokol uvedený v štandarde ANSI/IEEE 802.11. Vznikol v časoch, kedy bol obmedzený vývoz kryptografických technológií s viac ako 40 bitovým kľúčom z USA, takže používa iba 40 bitový kľúč. (v súčasnosti existuje modifikácia používajúca 104 bitový kľúč) WEP používa algoritmus RC4 na šifrovanie framu, pričom sa šifruje iba dátová časť a kontrolný súčet. WEP podporuje nastavenie 4 nezávislých kľúčov, ktoré je možné striedať pri šifrovaní framov. Použitý kľúč sa potom určí podľa časti IV framu. Postup: najprv sa vygeneruje IV, zreťazí sa s kľúčom a použije sa ako vstup pre RC4 (24 bitov IV + 40 bitov kľúča je 64 bitový vstup pre RC4). Výstupom RC4 je reťazec, ktorým sa zašifrujú dáta a kontrolný súčet framu použitím operácie XOR:

```
C = IV.(RC4(k.IV) XOR (P.crc(P)))
```

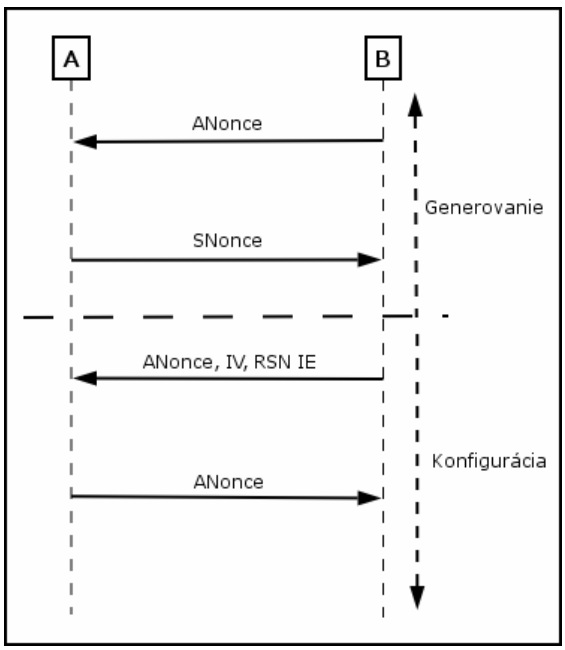
pričom

- **C** sú získané zašifrované dáta
- **P** je dátová časť framu
- **crc()** je funkcia vracajúca kontrolný súčet
- **k** je použitý šifrovací kľúč
- **IV** je inicializačný vektor
- **.** je operácia zreťazenia

Hlavné nedostatky spočívajú v tom, že 24 bitov zo vstupu pre RC4 sa posielajú nezašifrované, že sa šifrujú iba dáta a ich kontrolný súčet a vôbec sa nešifrujú control a management framy. Ďalej kľúče sú rovnaké pre celý segment siete, takže každý uzol dokáže dešifrovať všetku komunikáciu v rámci segmentu (ktorú zachytí). Štandard nedefinuje žiaden management kľúčov a generáciu IV, takže tieto závisia od konkrétnej implementácie. Bežné bezpečnostné praktiky pri používaní RC4 je používať dlhé kľúče a nepoužívať prvých 512 bajtov generovaného výstupu. WEP toto ignoruje, takže je veľmi zraniteľný. Chýbajúci management IV zas spôsobuje zraniteľnost pred replay útokmi a útokmi využívajúcimi znalosť plaintextu. Vo všeobecnosti stačí zachytiť dostatočný počet framov (rádovo 100000, navyše sa väčšinou dajú jednoducho generovať, napríklad opakovanými ARP requestami) a s využitím známych nedostatkov RC4 sa dá získať kľúč.

1.3.2 TKIP

Tento protokol bol navrhnutý na zlepšenie bezpečnosti na už existujúcom hardware. Keďže používa taktiež šifrovanie RC4, môže fungovať na rovnakom hardwari ako WEP, pričom postačuje upgrade software (resp. firmware). Hlavné zlepšenie je, že miesto statického kľúča generuje nový kľúč pre každú session. Tento kľúč sa nazýva PTK (Pair-wise Transient Key) a vypočíta sa pomocou 4-fázového handshake.



Obr. 1 PTK 4-way handshake

PTK je funkciou MAC adries oboch uzlov, PMK (Pair-wise Master Key, jedná sa o jediné zdieľané tajomstvo medzi oboma uzlami) a kvôli dynamickosti sa na oboch

stranách vygenerujú náhodné hodnoty. Ďalej na ochranu pred replay útokom používa TKIP sekvenčné počítadlo framov. Každý frame, ktorý príde mimo poradia je ignorovaný. Toto počítadlo sa ukladá do IV a extended IV. Na zabezpečenie integrity sa používa MIC (Message Integrity Code, taktiež volané Michael), ktoré chráni polia data, source a destination address a priority. MIC ochraňuje tieto polia pred týmito útokmi:

1. zámena bitov
2. úprava dát (orezanie, spájanie, ...)
3. fragmentačné útoky
4. iteratívne hádanie kľúča
5. presmerovanie (úpravou destination address)
6. impersonácia (úpravou source address)

TKIP používa hierarchiu kľúčov; základné rozdelenie je na kľúče pre unicast komunikáciu (Pair-Wise Keys) a kľúče pre skupinovú komunikáciu (multicast, broadcast). Oddelením unicast od multicast a broadcast zabraňuje falšovaniu adries (napr. úprava adresy broadcastovaného framu na unicastovú); keďže sa používa úplne iný kľúč, upravený frame bude odhalený a ignorovaný.

1.3.3 CCMP

Tento protokol bol predstavený ako súčasť štandardu IEEE 802.11i. Protokol zabezpečuje autentifikáciu a enkrypciu pomocou ľubovoľnej blokovej šifry (teda každá odoslaná správa je zašifrovaná blokovou šifrou a obsahuje autentifikačné polia). IEEE 802.11i používa blokovú šifru AES (Advanced Encryption System). CCMP používa rovnakú hierarchiu kľúčov ako TKIP a taktiež sa kľúče vytvárajú rovnako. Vstup pre šifru je kľúč, premenná hodnota (nonce), autentifikačné dáta a samotná správa. Hlavná požiadavka pre nonce je, aby bolo pre každú správu (pri rovnakom kľúči) unikátne; z toho dôvodu sa často používa sekvenčné číslovanie správ ako nonce (a ďalej tento prístup pomáha detekovat replay útoky a nesprávne poradie správ). Dĺžka nonce nie je pevne stanovená. Kvôli bezpečnosti je vhodná veľká dĺžka (vačšia odolnosť voči replay útokom), ale v prípade častého výskytu krátkych správ je výhodnejší kratší nonce, kvôli menším nárokom na objem prenesených dát.

1.3.4 Autentifikácia

1.3.4.1 Open System

Táto schéma nepoužíva žiadne overovanie totožnosti. Zdrojová stanica vyšle management frame požadujúci Open System autentifikáciu. Autentifikačná stanica odpovie, či bola autentifikácia úspešná, alebo nie.

1.3.4.2 Shared Key

Definovaná v IEEE 802.11, táto schéma vyžaduje WEP, pretože používa WEP (RC4) na šifrovanie autentifikačných framov. Predpokladá sa, že všetky participujúce stanice majú vopred k dispozícii zdieľaný tajný kľúč (jeho distribúciou sa štandardy nezaoberajú). Zdrojová stanica vyšle autentifikačnej stanici žiadosť o autentifikáciu. V prípade, že autentifikácia je možná odpovie autentifikačná stanica tzv. challenge textom. Zdrojová stanica tento text okopíruje a pošle naspäť, pričom frame bude šifrovaný pomocou WEP a zdieľaného kľúča. Autentifikačná stanica sa pokúsi dešifrovať frame a ak sa to podarí (skontroluje WEP ICV), porovná text s originálom. Ak súhlasia, autentifikácia je úspešná (a pošle sa príslušná odpoveď), vo všetkých ostatných prípadoch autentifikácia zlyháva (buď nesedí ICV, alebo sa texty nezhodujú). Táto schéma je nedostačujúca proti replay útokom a taktiež umožňuje vidieť ako čistý text, tak aj zašifrovaný, čo zjednodušuje hľadanie kľúča.

1.3.4.3 Service Set Identifier

Prístup do siete je povolený len staniciam, ktoré poznajú ESSID tejto siete. ESSID je verejný a je broadcastovaný v Beacon framoch. Tieto broadcasty sa dajú vypnúť, prípadne z nich odstrániť ESSID. Táto metóda nie je veľmi spoľahlivá, kedže ESSID sa dá zistiť pasívnym sledovaním siete (pretože ESSID sa nachádza v niektorých management framoch).

1.3.4.4 MAC Filtering

Táto schéma je centrálne riadena access pointom, ktorý povolí autentifikáciu podľa zoznamu MAC adries. Môže sa používať súčasne s ďalšími formami autentifikácie na ďalšie ohraničenie staníc. Bezpečnosť nie je veľmi vysoká. Jednak MAC adresy sú šírené nezašifrované (a teda môžu byť ľahko vystopované a sfalšované), ďalej zoznamy MAC adries sa ťažko spravujú, takže schéma nie je vhodná pre väčšie siete. Navyše neexistuje žiaden štandardný protokol na zdieľanie takýchto zoznamov medzi viacerými access pointami, čo sťažuje použitie pri väčšom počte access pointov (a najmä od rôznych výrobcov).

1.3.4.5 UAM

Táto schéma blokuje neznáme (nové MAC adresy), pričom im povolí jedine základné služby (ARP, DNS, DHCP a SSL na konkrétnu adresu). Stanica sa musí autentifikovať pomocou SSL spojenia na konkrétnu adresu (web server), dovtedy sa používa DNS hijacking, ktorý odpovie na každý DNS request adresou autentifikačného servra. Po úspešnej autentifikácii sa povolí normálna prevádzka pre danú MAC / IP adresu. Zabezpečenie komunikácie je prenechané v autentifikačnej fáze na TLS/SSL, neskôr nie je vyžadované žiadne.

1.3.4.6 WPA-PSK

Táto schéma používa TKIP na autentifikáciu a šifrovanie komunikácie. Celá sieť používa ako kľúč jedno zdieľané tajomstvo. PMK sa vypočíta zreťazením tohoto hesla, ESSID, dĺžky ESSID a celé je to 4096 krát hašované do 256 bitového digestu. Zraniteľné miesto je heslo, ktoré býva väčšinou alfanumerické a v takom prípade je to ekvivalentné približne 2.5 bitovému šifrovaniu na jeden znak. Táto schéma je využiteľná hlavne pre menšie siete, keďže používa rovnakú správu kľúčov ako WEP; vo väčších je z bezpečnostných dôvodov vhodnejšie používať samostatné heslá pre každého užívateľa.

1.3.4.7 EAP

Tento protokol sa používa v IEEE 802.1x. Protokol enkapsuluje layer 2 framy. Môže byť použitý na autenifikáciu, autorizáciu a účtovanie (AAA). Využíva služby externého autentifikačného servra na vydávanie kľúčov (per-session, per-user), napríklad RADIUS. Zdrojová stanica sa najprv asociuje s access pointom (použitím open system autentifikácie) a potom sa zacne samotná EAP autentifikácia. Ak sa úspešne dokončí, AP aj zdrojová stanica majú množinu potrebných kľúčov na bezpečnú komunikáciu medzi sebou. Možné problémy predstavuje počiatočná fáza, ktorá keďže nie je zabezpečená, je zraniteľná voči DoS útokom. Na samotnú autentifikáciu je možné použiť niekoľko protokolov:

1.3.4.7.1 LEAP

Jedná sa o proprietárny protokol firmy Cisco. Používa MS-CHAPv2 ako autentifikačnú metódu. Táto metóda je zraniteľná voči slovníkovým útokom, keďže nepoužíva salt (znáhodňovanie dát), username nie je šifrované a používa len 16 bitový DES algoritmus. Táto metóda totiž nebola navrhnutá na siete s voľným prístupom k médiu.

1.3.4.7.2 EAP-TLS

Tento protokol je v súčasnosti najbezpečnejší. Obidve strany sa navzájom autentifikujú pomocou certifikátov: najprv autentifikačný server pošle svoj certifikát klientovi, ten po jeho schválení odošle svoj certifikát. Hlavná nevýhoda tohoto protokolu je, že niektoré údaje (ako napríklad meno užívateľa či prihlasovacie meno) sa dajú získať z certifikátu, takže je možné zhromažďovať informácie o zvyklostiach užívateľov. Ďalej sú tu značné nároky na PKI (management certifikátov), takže väčšie uplatnenie má vo veľkých sieťach.

1.3.4.7.3 EAP-TTLS a PEAP

Tieto protokoly riešia vysoké nároky EAP-TLS na PKI. Obidva protokoly sú dvojfázové: v prvej fáze založia bezpečné spojenie medzi oboma stranami (TLS tunel) a autentifikačný server sa preukáže svojim certifikátom. V druhej fáze sa vykoná autentifikácia klienta. EAP-TTLS posiela cez TLS páry atribút-hodnota, čo umožňuje používať veľké množstvo metód (napr. všetky metódy definované v EAP ako aj staršie metódy ako PAP, CHAP, MS-CHAP, MS-CHAPv2) a pomocou zadefinovania nových atribútov sa dá jednoducho rozšíriť na použitie nových protokolov. PEAP je podobný EAP-TTLS, ale podporuje menšie množstvo protokolov.

1.3.4.7.4 EAP-FAST

Tento protokol bol vytvorený ako náhrada za LEAP. Podobne ako PEAP používa na zabezpečenie komunikácie TLS tunel, ale nepoužíva sa tu autentifikácia servra pomocou certifikátu. V druhej fáze sa využíva MS-CHAPv2. Na založenie TLS sa používa PAC, ktoré je unikátne pre každého užívateľa a je generované na strane servra. Distribúcia PAC k užívateľom je nultá fáza a od nej závisí celková bezpečnosť protokolu. Aby bola bezpečnosť rovnaká ako PEAP, musela by sa v nultej fáze použiť autentifikácia servra pomocou certifikátu (čo sú vlastne rovnaké požiadavky ako má PEAP). Pokiaľ sa použije metóda bez certifikácie, hrozí riziko man-in-the-middle útoku (keďže nie je zaručená totožnosť strán). Na rozdiel od LEAP sa ale nultá fáza vykonáva iba raz, takže ak prebehne úspešne, bezpečnosť je rovnaká ako u PEAP.

1.3.5 Útoky na Bezdrôtové siete

Bežné typy útokov na bezdrôtové siete môžeme rozdeliť na niekoľko kategórií podľa toho, na ktorú časť siete sa zameriavajú.

1.3.5.1 MAC podvrstva data-link layer

Väčšinou sa jedná o DoS útoky, napríklad broadcastom deautentifikačných framov so sfalšovanou MAC adresou, čo spôsobuje sústavné odpájanie uzla od AP. Útok je to aktívny a je možné ho odhaliť podľa enormne zvýšeného počtu deautentifikačných framov

1.3.5.2 WEP

Bežný spôsob útoku na WEP spočíva v pasívnom zhromažďovaní framov, ich analýze a následnému odvodeniu kľúča. Tento typ útoku je prakticky nezistiteľný. Po odvodení kľúča získa útočník prístup k celej sieti.

1.3.5.3 WPA-PSK

Bežné útoky sú taktiež pasívne, zhromažďujú a analyzujú framy, kľúč sa snažia odvodiť pomocou slovníkovej metódy.

1.3.5.4 EAP

Útoky závisia od typu použitého autentifikačného protokolu. Pokiaľ chýba certifikátová autentifikácia servra, používajú sa man-in-the-middle útoky (keď sa útočník vydáva za autentifikačný server). Ak sa používa LEAP, útoky sú pasívne a využívajú slovníkovú metódu.

1.4 Konkrétny príklad neverejnej siete so zabezpečenou data-link vrstvou

Polytechnic University of Catalonia. (U.P.C.), Barcelona

V tejto sieti vytvorili špecializované šifrovacie zariadenia (nazvané CryptoNet), ktoré nahradili funkciu ethernet switchov.

Okrem toho vytvorili aj Supervision and Administration Center (SAC), ktoré sa staralo o správu systému a prácu so šifrovacími kľúčmi.

Samotný princíp je nasledovný:

Celá sieť pozostáva z jednotlivých segmentov pripojených na backbone. Predpokladá sa, že komunikácia vnútri segmentov je už zabezpečená (napr. riadením prístupu, fyzickým obmedzením prístupu k zariadeniam atď.). Tieto segmenty sa potom pripájajú na backbone pomocou CryptoNetov, ktoré zabezpečujú komunikáciu.

CryptoNety majú nasledovné vlastnosti:

- poskytujú bezpečnostné služby nezávislé na protokoloch vyšších vrstiev
- nie sú závislé na dizajne jednotlivých zariadení, poskytujú služby pre celý segment, ktorý je na nich pripojený
- sú transparentné pre užívateľov aj zariadenia
- umožňujú koexistenciu zabezpečenej aj nezabezpečenej komunikácie
- nelimitujú možnú komunikáciu medzi rôznymi zariadeniami

CryptoNet pozostáva z troch základných častí:

Komunikačný submodul

> Zachytáva všetku komunikáciu pomocou komunikačných kontolérov v promiskuitnom móde. Po analýze adresy odosielateľa a adresáta sa aplikuje filtrovanie a kontrola citlivosti informácií

Kryptografický submodul

> Šifruje/dešifruje citlivé informácie pomocou DES, taktiež obsahuje procedúry na detekciu chýb a útokov a v takom prípade informuje SAC

Administračný submodul

> Zabezpečuje komunikáciu so SAC a stará sa o inicializáciu, konfiguráciu a správu samotného CryptoNet-u. Všetka komunikácia medzi zariadením a SAC je šifrovaná pomocou RSA

SAC zabezpečuje vzdialenú konfiguráciu systému (napr. nastavovať uzly ktoré komunikujú zabezpečene) a celkový chod systému, vrátane správy kľúčov.
Používa sa hybridný systém distribúcie kľúčov; hlavné šifrovacie kľúče sú distribuované SAC a session kľúče sú vymieňané priamo zariadeniami.
Systém kryptografie je takisto hybridný. Na digitálne podpisy a distribúciu kľúčov sa používa asymetrické šifrovanie s verejným kľúčom (konkrétne RSA), na samotné šifrovanie dát sa používa symetrické šifrovanie s tajným kľúčom (konkrétne DES).
Bezpečnostný protokol tvorí trojvrstvovú hierarchiu

i. Session protokol

   Pre každý pár CryptoNet-ov existuje spoločný kľúč (session key), ktorý sa používa na šifrovanie všetkej zabezpečenej komunikácie medzi touto dvojicou CryptoNet-ov (nezávislý od koncových zariadení). Tento je vyberaný z určenej množiny kľúčov a postupne sa mení

   - Na šifrovanie dát sa používa Triple Plaintext Block Cipher DES
   - Kvôli autentifikácii framov a detekcii neautorizovaných zmien sa pridáva MAC - Message Authentication Code, dlhý 4 bajty
   - Keďže sa týmto predĺži data field, môže byť potrebné rozdeliť frame na dva, preto je potrebné na túto skutočnosť upozorniť (2 bity, na určenie poradového čísla framu)
   - DES čipy zo svojej povahy pracujú najlepšie s dátami dĺžky násobku 8, takže pridáme výplň - padding. Informácia o dĺžke výplne sa pridá k flagom (3 bity)
   - Na zvýšenie bezpečnosti CryptoNet mení adresu prijímateľa na adresu prijímacieho CryptoNet-u, takže adresu pôvodného príjemcu (aj odosielateľa) taktiež musí pridať do dátovej časti
   - Navyše sa ešte pridáva sekvenčné číslo, kvôli prevencii útokov typu replay/reflection a znáhodneniu správ (tzv. whitening, zabráni porovnať dve šifrované správy a zistiť, či boli pred šifrovaním rovnaké) . Bežne sa používajú časové známky, ale kvôli jednoduchosti sa tu používajú iba sekvenčné čísla (nevyžadujú časovú synchronizáciu). Dĺžka je 4 bajty.

Header			Data							
				Encrypted data						
Destination Bridge Address	Source Bridge Address	Type	Flags + Padding length	Sequence no.	Source address	Destination address	Flags + Padding length	Data + Padding	MAC	CRC
# bytes			1	4	6	6	1	26-1474	4	

Obr. 2 Štruktúra zabezpečeného framu

ii. Master protokol

Zabezpečuje ďalší stupeň ochrany. Používa sa na zmenu session key medzi dvojicou CryptoNet-ov

- Keď je spoločný kľúč použitý istý počet krát (preddefinovaná hodnota), je potrebné ho vymeniť
- Využíva 2 bity z flagov na notifikáciu
- Vyslanie žiadosti o zmenu kľúča - prvý bit je nastavený na 1, druhý na 0
- Ak druhá strana dostane príslušný frame v poriadku, odpovie naňho prvým bitom 1 a druhým bitom 1
- Po prijatí odpovede vyšle iniciátor frame obsahujúci nový kľúč (vybratý náhodne z množiny a zašifrovaný master kľúčom) a bity nastaví na 0 a 1
- Ak je nový kľúč prijatý správne, príjemca odpovie vynulovaním oboch bitov (0 a 0)

iii. Administračný protokol

Tento protokol sa používa na zmenu master key pre dvojicu CryptoNet-ov. Táto činnosť je vykonávaná SAC. Použité šifrovanie je RSA

- Vyžaduje sa potvrdenie každého vyslaného framu, preto sa odoslané framy udržiavajú v bufferi, až kým nepríde odpoveď (prípadne su znovu odoslané ak odpoveď nepríde do predom stanoveného času)
- Protokol je založený na IEEE/802.2 LLC type II, pretože je spoľahlivý, štandardizovaný, connection-oriented a obsahuje flow control
- Dáta sú šifrované RSA algoritmom, identifikácia použitého kľúča je uložená do troch bitov flagov. Tie sú nezašifrované (inak by nebolo možné odšifrovanie)
- Adresa odosielateľa a prijímateľa je zakódovaná a navyše vo forme jednobajtových identifikátorov (keďže je predpoklad, že komunikujúcich strán nieje veľa, iba SAC a CryptoNet-y). Tabuľku identifikátorov obsahuje SAC aj každý CryptoNet

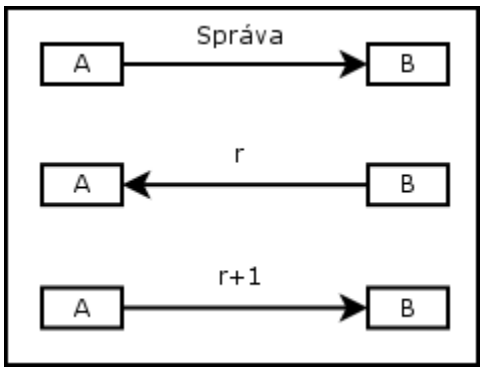
- Ďalej sa do framu pridáva typ služby, 1 bajt a v šifrovanej sekcii
- Keďže implementácia RSA pracuje s 512 bitovými blokmi (=64 bajtov), prípadne sa pridá výplň
- Kvôli ochrane sa ešte pridáva sekvenčné číslo

DSAP	SSAP	Control	Information							
			Flags				D.I./S.I	Service class	Padding length	Data	Padding	Sequence no.
			F1	F2	F3	XXXX						
# bytes			1				2	1	1	0-1464	0-63	4

Obr. 3 Štruktúra administračného framu

2 Model

Základná myšlienka modelu je využitie verejných kľúčov ako adresácie. Ak verejný kľúč uzla tvorí jeho adresu, tak je zaručené, že správa odoslaná na príslušnú adresu môže byť prečítaná len so znalosťou príslušného súkromného kľúča. Takže na overenie pravosti adresy odosielateľa stačí jednoduchý challenge-response.



Obr. 4 challenge - response

Takto odpadá nutnosť získavať autentické verejné kľúče inou cestou. Taktiež v prípade digitálneho podpisu umožňuje okamžité overenie, keďže príslušný verejný kľúč je súčasťou prijatého framu. Model preto predpokladá digitálne podpisovanie odosielaných framov.

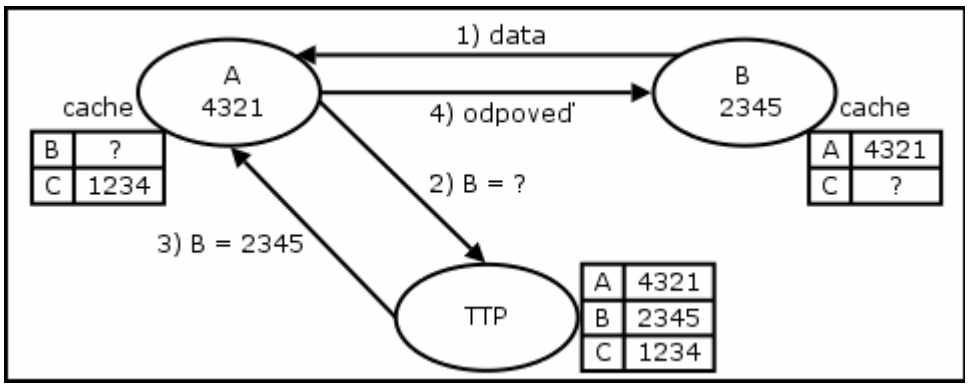
Nanešťastie, digitálne podpisovanie dát rovnakým kľúčom, aký sa používa na šifrovanie (konkrétne súkromným kľúčom) prináša bezpečnostné riziko. Riešením je teda použiť iný pár kľúčov na podpisovanie a iný na šifrovanie, prípadne úplne odlišný algoritmus na podpisovanie a na šifrovanie. Z hľadiska abstraktného modelu veľmi nezáleží ktorý z týchto prístupov použijeme. Pre jednoduchosť teda budeme uvažovať o rozdielnych kľúčoch na šifrovanie a podpisovanie.

Ak chceme dostatočnú bezpečnosť šifrovania, v prípade asymetrických šifier treba použiť relatívne veľké kľúče (rádovo 1024-2048 bitov). Ak uvažujeme adresu zloženú z dvoch verejných kľúčov (šifrovací kľúč a kľúč na overenie podpisu), jedna adresa zaberá zhruba 2048-4096 bitov, teda 256-512 octetov. Typický frame obsahuje adresy dve, odosielateľa a adresáta, okrem toho ešte nejaké riadiace údaje (napr. FCS, rôzne flagy, ...). Takto sa dopracujeme k hodnotám 512 - 1024 octetov len na adresáciu, čo pri bežnej MTU okolo 1500 octetov (Ethernet) vedie k veľmi nízkej efektivite.

Network	MTU [B]	Mód 0	Mód 1
16 Mbps Token Ring	17914	3,63%	0,39%
4 Mbps Token Ring	4464	14,56%	1,57%
FDDI	4352	14,94%	1,61%
Ethernet	1500	43,33%	4,67%
IEEE 802.3/802.2	1492	43,57%	4,69%
PPPoE (WAN Miniport)	1480	43,92%	4,73%
X.25	576	112,85%	12,15%
Veľkosť hlavičky: RSA 2048 a ECDSA 320 / MAC		**650 B**	**70 B**

Tab. 6 veľkosti MTU pre rôzne protokoly

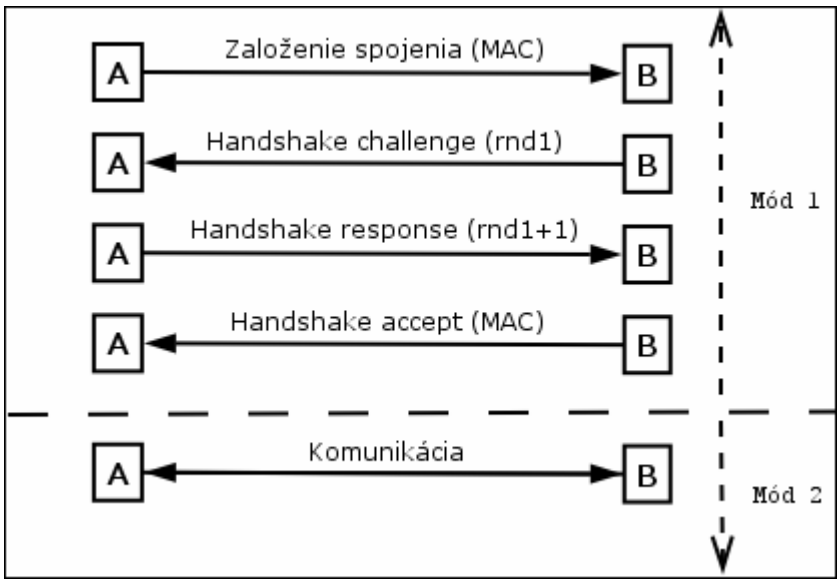
Keďže zväčšiť MTU nie je vždy možné, treba zmenšiť veľkosť adresy. Jednou z možností je použiť TTP na preklad jednej adresnej schémy na druhú (napr. klasické MAC adresy na adresy zložené z dvoch verejných kľúčov), pričom vo framoch by sa používala menej náročná schéma (MAC), ktorá by sa mapovala na náročnejšiu.



Obr. 5 Preklad adries pomocou TTP

Tento prístup má niekoľko nevýhod. Jednak je to závislosť od TTP: veľká záťaž na TTP (pravdepodobne by sa muselo jednať o dedikovaný server), nutnosť bezpečnej komunikácie medzi uzlami a TTP (autentickosť dát aj odosielateľa) a jediný bod zlyhania (ak zlyhá TTP, tak komunikácia je vážne ochromená).

Ďalšia možnosť je použiť súbežne dve adresné schémy. Každý uzol by mal jednu adresu vo forme dvoch verejných kľúčov a zároveň druhú adresu (napr. MAC). Obidve adresy by boli v rámci svojej adresnej schémy unikátne. Komunikácia by potom prebiehala nasledovne: najprv sa použije prvá adresná schéma (verejné kľúče), v ktorej sa dohodnú detaily komunikácie a vymenia sa adresy druhej adresnej schémy. Samotný prenos údajov už potom prebieha za použitia menej náročnej druhej adresnej schémy. Pokiaľ budú dáta podpisované príslušným kľúčom, dá sa ľahko overiť autentickosť dát aj odosielateľa. Efektivita prenosu dát bude porovnateľná s bežnými systémami (rovnaká adresná schéma, dáta navyše obsahujú digitálny podpis). Model bude využívať túto adresnú schému.



Obr. 6 2 adresné módy

Ako vieme, asymetrické šifrovanie je výpočtovo náročné, preto sa vačšinou používa spolu so symetrickými šiframi. Tento prístup využije aj náš model. V prvej fáze komunikácie (za použitia prvej adresnej schémy) sa uzly dohodnú na session key a (symetrickom) algoritme na šifrovanie, v druhej fáze budú dáta šifrované symetricky použitím dohodnutého session key. Ak sa použije dostatočne silná symetrická šifra, bezpečnosť bude rovnaká ako s použitím asymetrickej šifry, pričom výpočtová zložitosť by mala byť nižšia. Autentickosť dát aj odosielateľa zaručuje úvodný handshake a digitálny podpis.

2.1 Zhrnutie

Model využíva dva adresné módy. V prvom slúži ako adresa kombinácia verejného šifrovacieho kľúča a verejného kľúča určeného na overenie digitálneho podpisu. Tento mód sa použije pri nadviazaní komunikácie medzi dvoma uzlami na dohodnutí parametrov komunikácie (adresácia, použité algoritmy, stupeň zabezpečenia, session-key). Šifrovanie je asymetrické, šifruje sa verejným šifrovacím kľúčom adresáta, dáta sa podpisujú súkromným podpisovacím kľúčom odosielateľa. V druhom móde sa ako adresa používa unikátny identifikátor uzla (napríklad klasické MAC adresy v ethernete). Tento mód slúži na samotný prenos dát a vopred musia byť dohodnuté jeho parametre. Šifrovanie dát prebieha symetrickým algoritmom pomocou dohodnutého session-key, dáta sa podpisujú súkromným podpisovacím kľúčom odosielateľa. Dodnuté parametre si uzly držia v pamäti až do skončenia príslušnej session.

Každý odoslaný frame je podpísaný odosielateľom, bez ohľadu na adresný mód. Asymetrické algoritmy na šifrovanie a podpisovanie závisia od konkrétnej adresnej schémy, zatiaľ čo symetrický algoritmus na šifrovanie dát sa môže dynamicky meniť, podrobnosti si uzly dohodnú počas handshaku.

3 Štruktúra framu

V prípade, že v jednej sieti koexistujú viaceré protokoly, je treba ich medzi sebou rozlíšiť. V takom prípade by na začiatku framu mal byť identifikátor protokolu. Ďalej je potrebné rozlíšiť adresný mód (verejné kľúče vs. ID) a určiť dĺžku adresy. Buď môžeme použiť dve polia, jedno na mód adresácie a druhé na vyjadrenie dĺžky adresy, alebo iba jedno pole obsahujúce identifikátor, pričom identifikátor by určoval ako adresný mód, tak aj dĺžku adresy.

#	Mód	Šifrovanie	Podpis	Dĺžka adresy
0	1	RSA 1024	RSA 1024	256
1	1	RSA 2048	RSA 1024	384
2	1	RSA 2048	ECDSA 320	296
3	1	ElGamal 2048	DSA 1024	384
4	1	ElGamal 2048	ECDSA 320	296
5	1	ElGamal 1024	ElGamal 1024	256
...	...	...	...	...
128	2	AES	ECDSA 320	6
129	2	AES	RSA 2048	6
130	2	Twofish	ECDSA 320	6
...	...	...	...	...

Typ algoritmu a dĺžka adresy v jednom poli

#	Mód	Šifrovanie	Podpis
0	1	RSA	RSA
1	1	RSA	DSA
2	1	RSA	ECDSA
3	1	ElGamal	DSA
4	1	ElGamal	ECDSA
5	1	ElGamal	ElGamal
...	...	...	...
128	2	AES	ECDSA
129	2	AES	RSA
130	2	Twofish	ECDSA
...	...	...	...

Typ algoritmu

1. mód

Dĺžka adresy	
Šifrovanie	Podpis
2048	320

Dĺžka adresy	
Šifrovanie	Podpis
2048	1024

2. mód

Dĺžka adresy	
Šifrovanie	Podpis
0	6

Dĺžka adresy určená hodnotou

Obr. 7 Typ algoritmu a dĺžka adresy

Nutná súčasť hlavičky framu je adresa prijímateľa. Adresa odosielateľa nie je potrebná v prípade, že prijímateľ je schopný určiť odosielateľa z iných údajov (Session ID). Toto je ale nie je možné vždy, ale len po úspešnom nadviazaní spojenia, tj. v druhom adresnom móde. V tomto prípade ale adresy nemajú veľkú dĺžku a teda nemá veľký efekt vynechať ich (mierne sa zníži veľkosť hlavičky, ale vyžiada si to viac výpočtov na strane prijímateľa, keďže ten musí vyhľadať informácie o odosielateľovi). Z dôvodu jednoduchosti a flexibility bude teda náš model obsahovať aj adresu odosielateľa.

V prípade už nadviazaného spojenia je taktiež potrebné pole identifikujúce toto spojenie, Session ID. V prvom adresnom móde nemá veľký význam, ale môže sa použiť napríklad na dohodnutie Session ID pre spojenie.

Taktiež ak je možné používať rôzne šifrovacie algoritmy, je potrebné pole určujúce použitý algoritmus.

Ak je možné používať viac rôznych metód na podpisovanie, taktiež treba určiť ktorá z nich sa používa. Podľa identifikátora typu je potom možné zistiť veľkosť podpisu. Typy podpisov sú dopredu zadefinované, vrátane rôznych dĺžok v octetoch.

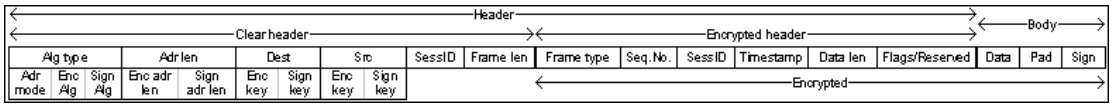
Ďalej môže byť potrebné rozlíšiť typ framu (request / reply / riadiace framy), prípadne nejaké parametre framu (flags).

Pre správne spracovanie framu je dobré poznať dĺžku celého framu. V prípade, že hlavička nemá pevnú dĺžku, tak aj dĺžku hlavičky.

Niektoré šifrovacie algoritmy (hlavne blokové symetrické šifry) vyžadujú vstup pevnej dĺžky (alebo násobku pevnej dĺžky). Preto môže byť frame doplnený na potrebnú dĺžku paddingom. Potrebujeme teda pole udávajúce dĺžku výplne, aby ju bolo možné odlíšiť od dát. Prípadne môžeme mať pole udávajúce dĺžku dát a dĺžku výplne vypočítame.

Výplň môže byť pred dátami, alebo za nimi. Keďže protokol vyžaduje celý frame neporušený, žiadna z možností nedáva výhody (ak by bolo možné spracovať neúplný frame, výplň by mala nasledovať za dátami. V prípade poškodenia konca framu je vyššia šanca zachovania dát nepoškodených). Model bude používať výplň za dátami, kvôli jednoduchšiemu spracovaniu.

Na zaručenie autentickosti dát (aj odosielateľa) bude celý frame podpísaný a podpis pripojený na koniec. Tento podpis zároveň bude slúžiť aj ako kontrolný súčet, keďže akékoľvek chyby pri prenose spôsobia, že podpis nebude súhlasiť s dátami.



Obr. 8 Štruktúra framu

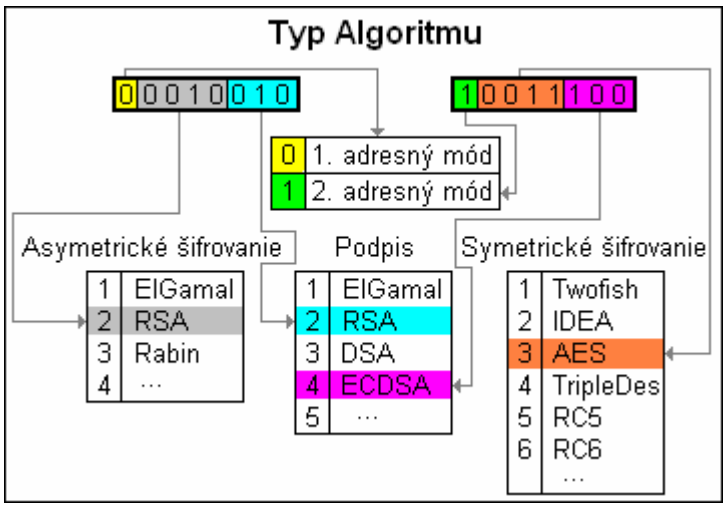
3.1 Podrobný popis jednotlivých polí

3.1.1 Typ šifrovacieho algoritmu

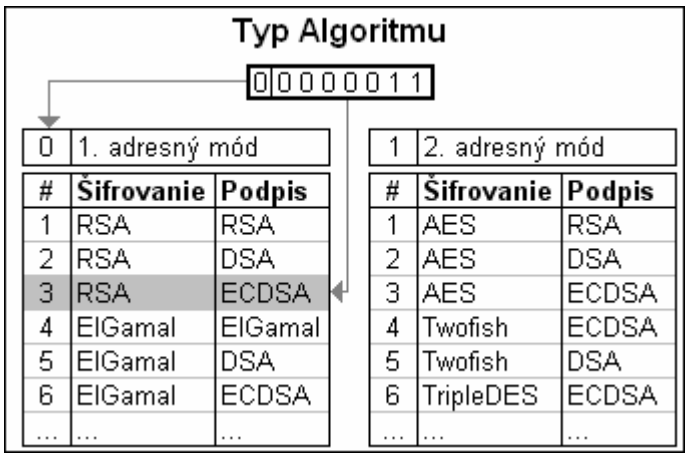
Toto pole určuje typ adresácie. Rozlišuje medzi prvým a druhým adresným módom. V prípade, že sa súbežne používa viac rôznych asymetrických šifrovacích algoritmov, alebo existuje viac algoritmov na digitálne podpisovanie, toto pole medzi nimi rozlišuje. Pre druhý adresný mód taktiež určuje použitý šifrovací algoritmus (ak je možné použiť viac algoritmov).

Príklad:

1 octet. Najvýznamnejší bit rozhoduje o adresnom móde, ostatných 7 bitov určuje typ šifrovacích algoritmov. Buď formou tabuľky, alebo niekoľko bitov (4) určuje typ šifrovacieho algoritmu a zvyšné bity (3) určujú typ algoritmu na podpisovanie. Tento prípad umožňuje 128 rôznych typov adresácie v každom adresnom móde. Ak je potrebné viac, treba toto pole rozšíriť.



Obr. 9a Typ algoritmu - oddelené tabuľky



#	Šifrovanie	Podpis
1	RSA	RSA
2	RSA	DSA
3	RSA	ECDSA
4	ElGamal	ElGamal
5	ElGamal	DSA
6	ElGamal	ECDSA
...	...	...

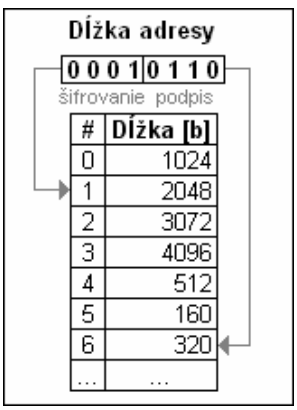
#	Šifrovanie	Podpis
1	AES	RSA
2	AES	DSA
3	AES	ECDSA
4	Twofish	ECDSA
5	Twofish	DSA
6	TripleDES	ECDSA
...	...	...

Obr. 9b Typ algoritmu – spojené tabuľky

3.1.2 Dĺžka adresy

Poskytuje informáciu o dĺžke adresy. Sú dve možnosti uloženia údajov: identifikátory do tabuľky s vopred pripravenými hodnotami, alebo priame číselné hodnoty. Prvý prístup vyžaduje menej bitov, druhý je flexibilnejší.

Príklad: Pri použití tabuľkových hodnôt stačia 1-2 octety.



#	Dĺžka [b]
0	1024
1	2048
2	3072
3	4096
4	512
5	160
6	320
...	...

Obr. 10a Dĺžka adresy - tabuľka

Pri použití číselných hodnôt sú potrebné 4 octety: 2 na dĺžku šifrovacieho kľúča v octetoch a 2 na dĺžku podpisovacieho kľúča v octetoch. To znamená, že kľúče môžu mať teoreticky 0-524280 bitov (dĺžka je udávaná v octetoch, 0-65535 octetov). V prípade jednooctetových polí by kľúč mohol mať len 0-2040 bitov, čo nemusí byť postačujúce. Riešiť by sa to dalo určením minimálnej veľkosti kľúča, napríklad 1024 bitov; v takomto prípade by jeden octet stačil pre veľkosť 1024-3064 bitov.

Celková dĺžka adresy bude teda súčtom dĺžok jednotlivých kľúčov. V prípade druhého adresného módu môže byť ako dĺžka adresy brané celé pole ako jedno číslo, alebo jedna časť bude nulová.

Dĺžka adresy			
Šifrovanie		Podpis	
00000001	00000000	00000000	00101000
256 octetov = 2048 bitov, 40 octetov = 320 bitov			

Obr. 10b Dĺžka adresy - hodnota

3.1.3 Adresa príjemcu a Adresa odosielateľa

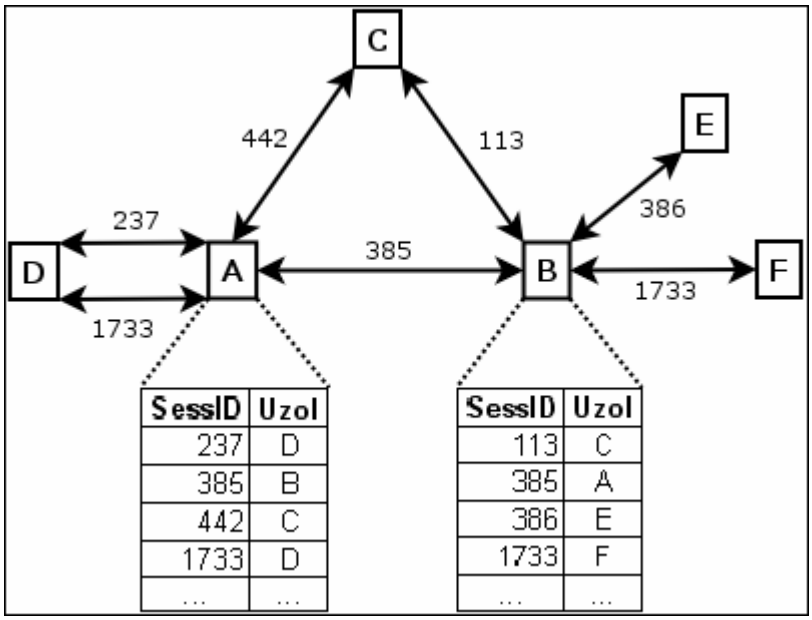
Polia udávajúce adresy príjemcu a odosielateľa. Dĺžka je závislá od poľa Dĺžka adresy.

3.1.4 Session ID

Identifikátor konkrétneho spojenia medzi dvoma uzlami. Session ID môže byť unikátne vzhľadom na celú sieť (čiže pomocou Session ID sa dá určiť konkrétne spojenie v rámci celej siete), unikátne vzhľadom na uzol (jeden uzol má nanajvýš jedno spojenie s daným Session ID), alebo unikátne vzhľadom na oba uzly (tj. Session ID určuje konkrétne spojenie medzi danými dvoma uzlami. Jeden uzol môže mať viac spojení s rovnakým Session ID, ale tieto spojenia majú rozličné druhé strany).

Taktiež Session ID môže byť obojstranne rovnaké, alebo z každej strany iné (môže to byť výhodné v prípade, ak sa uzlom nepodarí vždy zhodnúť na rovnakom Session ID. Táto možnosť nastáva v prípade Session ID unikátnych vzhľadom na 1 uzol).

Kvôli jednoduchosti a flexibilite bude náš model využívať obojstranne rovnaké Session ID unikátne vzhľadom na oba uzly.



Obr. 11 Session ID unikátne vzhľadom na oba uzly

3.1.5 Frame length

Dĺžka celého framu v octetoch. Veľkosť tohoto poľa závisí od MTU, typicky by mali byť potrebné 2 octety, pre MTU väčšie ako 64kB 3 octety.

3.1.6 Typ framu

Určuje typ framu.

Typ	Popis
0	Dátový frame
1	Začiatok novej komunikácie
2	Handshake challenge
3	Handshake response
4	Handshake accept
5	Handshake reject
6	Žiadosť o nový Session Key
7	Odpoveď na žiadosť o nový Session Key
8	Zamietnutie žiadosti o nový Session Key
9	Potvrdenie nového Session Key
10	Žiadosť o zrušenie spojenia
11	Odpoveď na žiadosť o zrušenie spojenia
12	Keepalive
13	Resend request

Tab. 7 Typy framov

3.1.7 Sekvenčné číslo framu

Číslo udávajúce poradie framu v aktuálnom spojení. Využitie má pri detekcii replay útokov, alebo riadení toku dát.

3.1.8 Session ID

Duplicitné pole, určené na ochranu pred replay útokom (keďže je šifrované). Frame v ktorom sa toto pole nezhoduje s nešifrovaným Session ID bude ignorovaný.

3.1.9 Timestamp

Časový údaj, určený na prenášanie informácie o čase. Taktiež ohraňuje pred replay útokmi.

Veľkosť poľa závisí od požadovaného rozsahu timestampu. Ak chceme napríklad určovanie času na milisekundy v rozsahu 100 rokov, potrebujeme 42 bitov. Po zaokrúhlení na celé octety je to 48 bitov, čo nám dáva rozsah necelých 9000 rokov. Pri použití 5 octetov dostávame rozsah zhruba 35 rokov. Ak nám stačí presnosť na desatiny sekundy, rozsah primerane vzrastie. V takom prípade by malo byť postačujúce používať 5 octetov (rozsah 3484 rokov).

Octety	Sekundy	Desatiny	Stotiny	Tisíciny
1	4,27 minúty	25,60 sekundy	2,56 sekundy	0,26 sekundy
2	18,20 hodiny	109,23 minúty	10,92 minúty	1,09 minúty
3	194,18 dňa	19,42 dňa	1,94 dňa	4,66 hodiny
4	136,10 roka	13,61 roka	1,36 roka	49,71 dňa
5	34 841,18 roka	3 484,12 roka	348,41 roka	34,84 roka
6	8 919 342,73 roka	891 934,27 roka	89 193,43 roka	8 919,34 roka

Tab. 8 Timestamp

3.1.10 Data Length

Dĺžka dátovej časti. Potrebné na odlíšenie dát od výplne, keďže dáta môžu mať ľubovoľný tvar aj veľkosť. Dĺžka výplne sa vypočíta ako

dĺžka výplne = dĺžka framu – dĺžka hlavičky – dĺžka podpisu – dĺžka dát

3.1.11 Flags/Reserved

Rôzne (jednobitové) nastavenia framu. Prípadné miesto na rozšírenia v budúcnosti.

(1-2 octety)

Príklad:

- frame nie je šifrovaný
- frame bol odoslaný zakladateľom spojenia

- potvrdiť prijatie framu
- frame je samostatný, alebo fragment
- priorita

3.1.12 Data

Samotné dáta.

3.1.13 Padding

Výplň k dátam, potrebná pre niektoré šifrovacie algoritmy. Dĺžka závisí od použitého šifrovacieho algoritmu, dĺžky dát a dĺžky nešifrovanej hlavičky.

3.1.14 Podpis

Digitálny podpis framu. Dĺžka závisí od použitého algoritmu na podpisovanie.

> Príklad: Pri použití SHA-1 ako hašovacej funkcie bude mať výsledný haš dĺžku 20 octetov. Ak sa tento haš zašifruje pomocou 1024 bitovej RSA, výsledok bude zaberať rovnako 1024 bitov, tj. 256 octetov. Ak sa použije ECDSA algoritmus so 160 bitovým kľúčom, výsledok bude zaberať 320 bitov, tj. 40 octetov.

4 Čo šifrovať

Určite treba šifrovať dáta. Otázne je, čo (a či vôbec) šifrovať vo zvyšku framu.

4.1 Hlavička

Hlavička obsahuje údaje, ktoré môžu byť potrebné na správnu identifikáciu a dešifrovanie framu. Z toho dôvodu nie je žiadúce aby bola celá hlavička šifrovaná. Napríklad adresy, typ použitého šifrovacieho algoritmu, či Session ID by nemali byť šifrované. Iné časti hlavičky nesúvisiace priamo s šifrovaním alebo adresáciou je možné šifrovať. Takto sa dostaneme k rozdeleniu hlavičky na 2 časti - nešifrovanú a šifrovanú.

4.1.1 Nešifrovaná hlavička

4.1.1.1 Adresa odosielateľa a príjemcu

Pokiaľ by sa mali šifrovať, každý uzol by musel dešifrovať hlavičku každého framu (prípadne v závislosti od použitého algoritmu aj celý frame) a framy neurčené pre neho zahadzovať (tie, ktoré po dešifrovaní nedajú zmysluplný výsledok). Tento prístup je zbytočne nákladný, keďže väčšinou adresy odosielateľa a prijímateľa nie sú citlivé údaje.

V prípade, že aj informácia ktoré uzly spolu komunikujú má byť dôverná, môže sa šifrovať celý frame. Vtedy treba odlíšiť od seba jednotlivé módy (šifrovaný celý frame vs. šifrované dáta). To sa dá buď globálnym dohovorom v sieti (všetky uzly sa zhodnú, že sa bude používať iba jeden mód), alebo odlíšením samotných framov. To sa môže robiť buď dohodnutým prefixom (čiže prakticky špeciálnym flagom na začiatku framu), metódou pokus-omyl (príjemca skontroluje frame, či obsahuje platné polia. ak nie, predpokladá, že je šifrovaný a pokúsi sa ho dešifrovať), alebo oznámením módu vopred (napr. broadcastom).

4.1.1.2 Typ šifrovacieho algoritmu

Táto časť hlavičky by sa nemala vôbec šifrovať. Pokiaľ sa využíva, znamená to, že je možné použiť viac rôznych šifrovacích algoritmov. Ak by bolo toto pole šifrované, prijímateľ by nevedel aký algoritmus použiť na dešifrovanie. Buď by sa to musel dozvedieť inou cestou (napr. vopred dohodnuté s odosielateľom, to samozrejme vyžaduje nešifrovanú adresu odosielateľa), alebo postupne skúšať všetky algoritmy,

kým úspešne nedešifruje frame (alebo nevyčerpá všetky možnosti). Táto metóda je nepraktická.

4.1.1.3 Session ID

Vzhľadom na to, že môže prebiehať naraz viac spojení medzi tými istými uzlami, môže byť Session ID nutné na výber správneho algoritmu a kľúča na dešifrovanie. V opačnom prípade je možné toto pole šifrovať.

V prvom adresnom móde sa toto pole využíva na dohodnutie budúceho Session ID, takže v tomto prípade má význam šifrovať ho.

4.1.1.4 Frame Length

Celková dĺžka framu by nemala byť citlivá informácia a preto sa kvôli jednoduchosti spracovania nebude šifrovať.

4.1.2 Šifrovaná hlavička

4.1.2.1 Typ framu

4.1.2.2 Sekvenčné číslo

4.1.2.3 Polia udávajúce dĺžky

4.1.2.4 Flags/reserved

Tieto polia môžu byť šifrované, keďže nie sú potrebné na korektné dešifrovanie framu. Šifrovanie spolu s podpisom ich chráni pred neoprávnenou modifikáciou.

4.1.2.5 Kópia Session ID, timestamp

Tieto polia sú v šifrovanej časti kvôli zamedzeniu replay útokov. Session ID nás ochráni pred pokusom upraviť frame poslaný v rámci iného spojenia. Timestamp zaručuje čerstvosť správy. Napríklad nie je možné použiť správu odoslanú v minulosti počas spojenia s rovnakým Session ID, ktoré ale medzitým bolo zrušené (a Session ID neskôr znovu použité).

4.2 Dáta a výplň

Tieto by mali byť šifrované. Výnimkou je broadcast, ktorý zo samotnej podstaty vyžaduje nešifrované dáta (alebo ak aj šifrované, tak takým spôsobom, aby ich mohol ľubovoľný uzol dešifrovať).

4.3 Digitálny podpis

Podpísaný bude celý frame (samozrejme s výnimkou samotného digitálneho podpisu), vtedy sa podpis môže použiť aj ako FCS, MAC a MIC.

Otázka je, v akom poradí šifrovať a podpisovať.

1. Dáta sa zašifrujú, zostaví sa frame a celý sa podpíše; podpis nebude šifrovaný

Tento prístup umožňuje sfalšovať adresu odosielateľa. Falšovateľ môže zachytit pravý frame, odstrániť z neho podpis, zmeniť adresu a podpísať vlastným podpisom.

Toto je využiteľné najmä pri prvom adresnom móde, keďže sa šifruje (známym) verejným kľúčom adresáta. Teda dešifrujú sa vždy rovnakým kľúčom, nezávislým od odosielateľa.

V druhom adresnom móde by bola nutná znalosť session key na posielanie zmysluplných informácií. Keďže dáta sú šifrované session key-om a ten je závislý od odosielateľa. S pravdepodobnosťou hraničiacou s istotou je session key medzi pravým odosielateľom a adresátom rôzny od toho medzi falšovateľom a adresátom, takže po prijatí sfalšovaného framu sa ho nepodarí dešifrovať.

Možná výhoda je v tom, že ktokoľvek môže overiť autenticitu framu, bez znalosti dešifrovacieho kľúča.

2. Dáta sa zašifrujú, zostaví sa frame a celý sa podpíše, podpis sa následne zvlášť zašifruje

Podobne ako v prvom prípade hrozí falšovanie pre prvý adresný mód. Navyše v istých prípadoch môžu nastať problémy po zašifrovaní (napríklad ak sa na podpisovanie používa rovnaký algoritmus a rovnaký pár kľúčov).

Autenticitu môže overiť len príjemca (konkrétne uzol, ktorý pozná dešifrovací kľúč, či už súkromný šifrovací kľúč adresáta, alebo session key).

3. Zostaví sa frame s nezašifrovanými dátami a podpíše sa

3a. Zašifrujú sa len dáta, podpis zostane nešifrovaný

Na falšovanie podpisu je nutná znalosť buď dešifrovacieho kľúča, alebo nezašifrovaných dát. Takisto, v prípade druhého adresného módu je falšovanie zmysluplných informácií zložité.

3b. Zašifrujú sa aj dáta, aj podpis

V tomto prípade dokáže podpis sfalšovať buď príjemca, alebo odosielateľ (resp. uzly so znalosťou príslušných kľúčov). Preto model použije tento prístup.

Pozn. kvôli správnemu dešifrovaniu môže byť nutné pred zašifrovaním frame preusporiadať (v prípade, že šifrovací aj podpisovací algoritmus sú rovnaké)[1]

Z dôvodu priestorových možností sa nebude podpisovať samotný frame, ale jeho haš. V prípade použitia kvalitnej hašovacej funkcie je falšovateľnosť podpisu porovnateľná, ako pri podpisovaní samotneho framu.

[1] [2] kapitola 11, strany 435-436: 11.3.3(i) Reblocking problem

5 Vysielanie viacerým adresátom

5.1 Broadcast

Broadcast znamená vysielanie informácií všetkým uzlom na sieti. Keďže má byť dostupný každému, dáta by nemali byť šifrované. Pokiaľ chceme obmedzit rozsah príjemcov, treba použiť multicast.

V prvom adresnom móde špeciálnou adresou prijímateľa (napr. 0). Každý frame je podpísaný, takže je zaručená autenticita (obsahu, aj odosielateľa). Zo samotnej podstaty broadcastu vyplýva, že dáta sa nemôžu šifrovať.

Broadcast v druhom adresnom móde prináša problém overenia podpisu. V závislosti od nastavenia cache adries je možné prvý frame broadcastovať v prvom adresnom móde a nasledujúce framy v druhom, až kým nevyprší čas, počas ktorého sa uchováva v pamäti mapovanie prvého typu adries na druhý.

Ak je dostupný centrálny uzol, je možné od neho po obdržaní framu získať verejný podpisovací kľúč odosielateľa, takže broadcast môže plne prebiehať v druhom adresnom móde.

5.2 Multicast

Multicast slúži na vysielanie informácií viacerým (ale nie všetkým), vopred určeným uzlom v sieti. Pred samotným multicastom je teda nutné určiť okruh uzlov, ktorým sa majú informácie vysielať a zabezpečiť, aby nikto mimo tejto skupiny nemal k vysielaným dátam prístup. To znamená buď fyzický prístup, alebo nemožnosť dešifrovať dáta. Keďže nie vždy je možné fyzicky zamedziť prístup k framom (napr. bezdrôtové siete), je potrebné dáta šifrovať tak, aby ich mohli dešifrovať len určené uzly.

Model rieši multicast komunikáciu pomocou špeciálnych adries, pričom celá multicast skupina vystupuje ako jeden uzol.

V prípade, že v sieti existuje centrálny server (TTP), môže mať na starosti správu multicast skupín. Pre príslušnú skupinu vygeneruje adresy (v oboch adresných módoch), ktoré poskytne zabezpečenou komunikáciou jednotlivým uzlom. Taktiež uzly dostanú kópiu súkromného šifrovacieho kľúča, aby boli schopné dešifrovať prijaté údaje. Keďže uzly by nemali odosielať framy pod adresou multicastovej

skupiny, nepotrebujú poznať súkromný podpisovací kľúč tejto skupiny. Tým pádom adresa (v prvom adresnom móde) môže byť zložená z verejného šifrovacieho kľúča a nulového (alebo neplatného) verejného podpisovacieho kľúča. Takto je možné jednoducho rozlišovať multicast adresy a ignorovať prichádzajúce framy z multicastových adries.

Ak chce uzol posielať dáta multicast skupine (pričom sám nemusí byť jej členom), mal by najprv v prvom adresnom móde oznámiť použitý session key a potom vysielať dáta v druhom adresnom móde. Týmto sa automaticky zabezpečí to, že uzly multicast skupiny budú poznať verejné kľúče odosielateľa a môžu teda overiť podpis vo frame. V prípade komunikácie v rámci multicast skupiny môže byť periodicky dohadovaný globálny session kľúč s určitou časovou platnosťou. Túto činnosť by mal vykonávať správca multicast skupiny (centrálny uzol, ak existuje, alebo zakladateľ skupiny). Pokiaľ je komunikácia dátovo nenáročná, môže prebiehať v prvom adresnom móde.

6 Práca uzla

6.1 Centrálny uzol

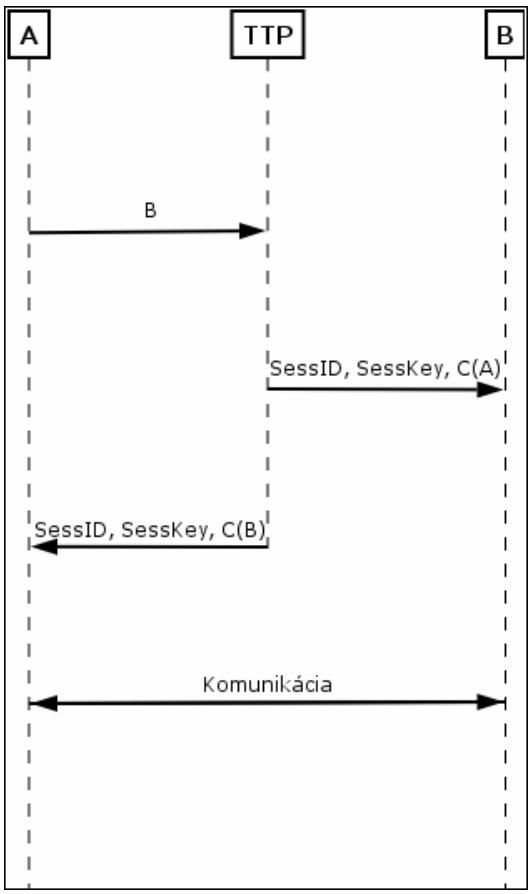
Ak je centrálny uzol k dispozícii, je možné zjednodušiť niektoré časti modelu. Centrálny uzol môže slúžiť ako certifikačná autorita, register verejných kľúčov, správca multicast skupín, ...

Takto je možné úplne vynechať prvý adresný mód pri bežnej komunikácii a používať len druhý. Problémom je určenie Session ID a Session Key, taktiež distribúcia verejných podpisovacích kľúčov obom stranám. Tieto úlohy môže splniť centrálny uzol. Vygeneruje Session ID a Session Key a spolu s certifikátom verejného podpisovacieho kľúča druhej strany ich pošle obom uzlom. Takto môže byť Session ID unikátne vzhľadom na celú sieť a teda je možné vynechať adresné polia z framu. Komunikácia medzi uzlami a centrálnym uzlom musí byť ale zabezpečená; buď bude prebiehať v prvom adresnom móde, alebo bude centrálny uzol zdieľať s každým jedným uzlom vlastný Session Key (a podľa potreby ho obmieňať).

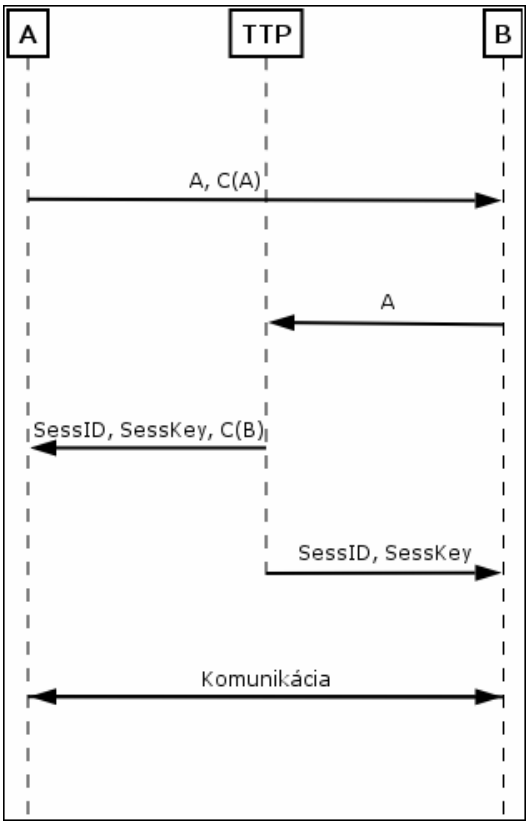
V tomto prípade môže byť žiadúce pridať do framu pole určujúce odosielateľa (alebo príjemcu) framu. Keďže na túto informáciu postačí jeden bit (hovoriaci, či odosielateľ je iniciátor spojenia, alebo nie), môže byť zahrnutý medzi flags.

Je niekoľko možností, ako nadviazať spojenie. Buď môže odosielateľ priamo kontaktovať adresáta, alebo kontakt sprostredkuje centrálny uzol. Keďže sa používa iba symetrické šifrovanie, Session Key musí byť generovaný centrálnym uzlom.

Ďalšia funkcia centrálneho uzla môže byť mapovanie adries data-link layer na adresy network layer – ARP, toto je ale už mimo nášho modelu, keďže ťažisko spočíva v network layer.



Obr. 12a TTP – nepriame nadviazanie komunikácie

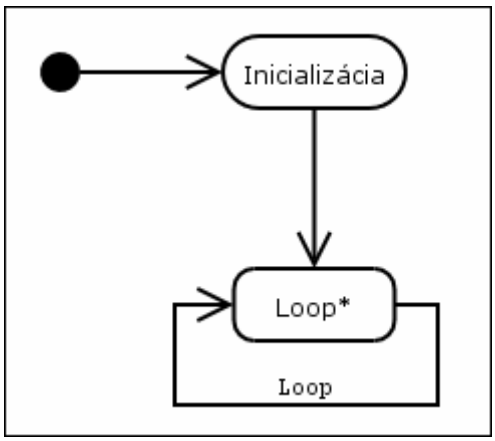


Obr. 12b TTP – priame nadviazanie komunikácie

Pri nadviazaní komunikácie pošle odosielateľ požiadavku centrálnemu uzlu. Ten vygeneruje Session ID a Session Key a pošle ich obom stranám. Prijímateľovi pošle certifikát odosielateľa a opačne.

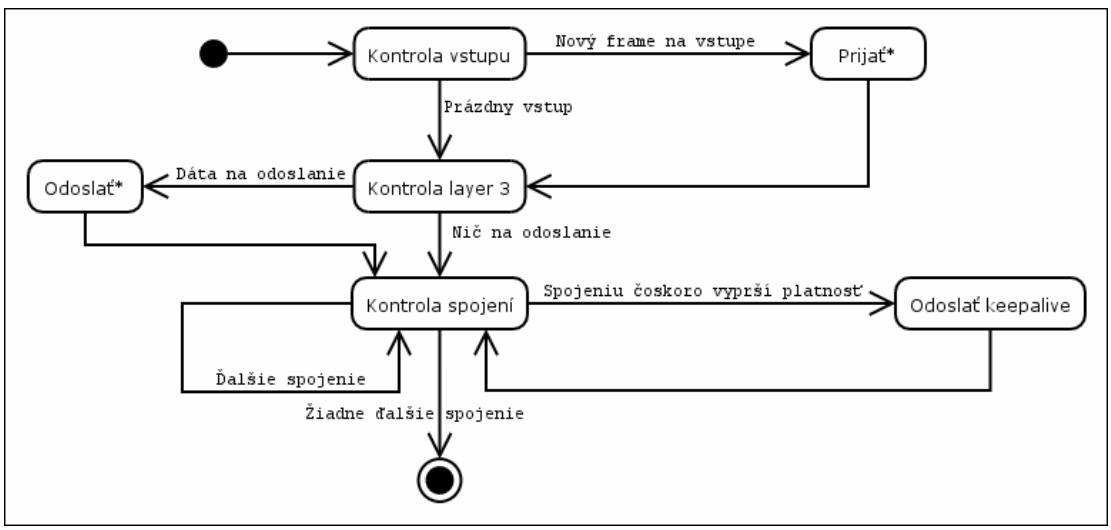
Pri nadviazaní komunikácie pošle odosielateľ svoj certifikát prijímateľovi. Ten pošle požiadavku centrálnemu uzlu, ktorý vygeneruje Session ID a Session Key a pošle ich obom stranám. Odosielateľovi taktiež pošle certifikát prijímateľa.

6.2 Činnosť bežného uzla



Obr. 13 Nekonečná slučka

Samotná činnosť bude prebiehať v nekonečnej slučke. Budeme kontrolovať, či na vstup neprišiel nejaký frame. Ak áno, prijmeme ho a spracujeme. Ďalej skontrolujeme, či netreba odoslať nejaké dáta od network layer a ak áno, odošleme ich. Nakoniec skontrolujeme stav všetkých spojení; tie, ktorým hrozí zrušenie kvôli nečinnosti, treba udržať funkčné odoslaním keepalive framu. Túto činnosť opakujeme do nekonečna (teda celý čas keď sme pripojení na sieť).



Obr. 14 Loop

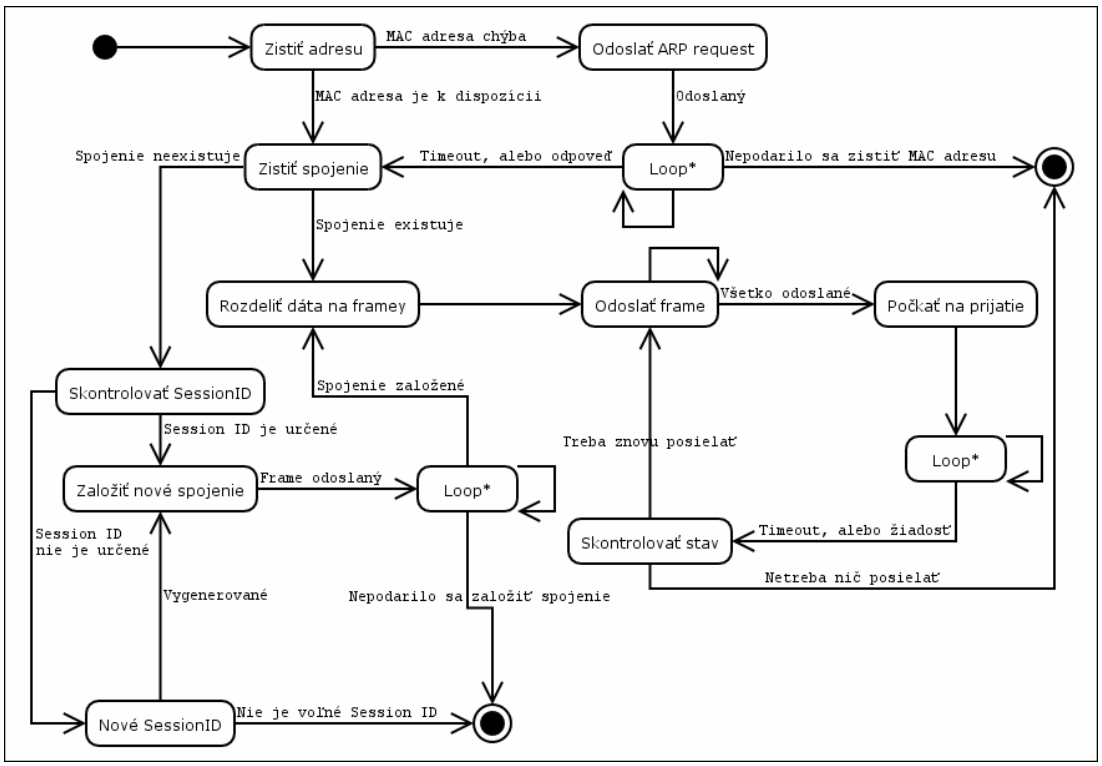
6.2.1 Odosielanie framov

Najprv treba zistiť akým spojením máme dáta odoslať. Prvá činnosť teda je zistiť MAC adresu (druhý adresný mód) príjemcu. Ak túto adresu máme v ARP pamäti, použijeme ju. Ak nie, vyšleme ARP request broadcast a počkáme na odpoveď. Ak žiadna odpoveď nepríde, dáta nemôžeme odoslať. Prípadne môžeme dáta odoslať

prvým adresným módom, ak je to možné vzhľadom na veľkosť dát a prenosovej kapacity framu.

Ďalej treba určiť konkrétne spojenie. Ak už spojenie existuje, network layer ho môže určiť (napríklad odpoveď na požiadavku na úrovni vyšších vrstiev, bežná komunikácia atď.). Ak spojenie neexistuje, odošleme požiadavku na založenie a počkáme na odpoveď. Ak sa spojenie nepodarí založiť, dáta nemôžeme odoslať. Keď sa nám podarilo určiť spojenie, rozdelíme dáta na jeden, alebo viac framov (podľa potreby) a jednotlivé framy odošleme.

Keďže model používa negatívne potvrdenie (vyžiadanie si chýbajúcich framov), počkáme určitý čas, či takáto žiadosť nepríde. Ak príde, vyžiadané framy znovu pošleme. Keď uplynie potrebný čas (alebo sa vyčerpá maximálne množstvo opätovných odoslaní), považujeme framy za doručené. Prípadná modifikácia by bola, ak by sa pomocou flagov mohlo vyžiadať potvrdenie prijatia framu. V takom prípade môžeme úplne vynechať negatívne potvrdenie.



Obr. 15 Odoslať

6.2.2 Prijatie framu

Frame čítame postupne, takže je možné spracúvať ho už počas prijímania. Najprv zistíme adresný mód z poľa Typ šifrovacieho algoritmu. Nasleduje dĺžka adresy, ktorá je potrebná na správne prečítanie adries. Potom prečítame adresu príjemcu.

Skontrolujeme či je frame určený nám, tj. adresa je naša, alebo patríme do multicast skupiny, ktorej je frame adresovaný, alebo frame bol broadcastovaný. Ak neplatí žiadna z týchto podmienok, frame ignorujeme.

Ďalej prečítame adresu odosielateľa. Keďže multicast a broadcast adresy nie sú platné adresy odosielateľa, takéto framy taktiež ignorujeme. Prečítame Session ID a skontrolujeme, či existuje takéto spojenie a či je platné (spojenie, ktoré dlho neprijímalo žiadne framy považujeme za neaktívne a zrušíme ho). Aj ak spojenie neexistuje, frame spracúvame ďalej (spojenie sa môže práve zakladať, táto možnosť bude ošetrená neskôr).

Vo frame ďalej nasleduje dĺžka celého framu. Ak je frame šifrovaný, musíme najprv prečítať zvyšok framu, dešifrovať ho a až potom môžeme pracovať s ďalšími poľami. Ak frame šifrovaný nie je, môžeme pokračovať v čítaní zvyšku framu (napr. Frame je broadcast).

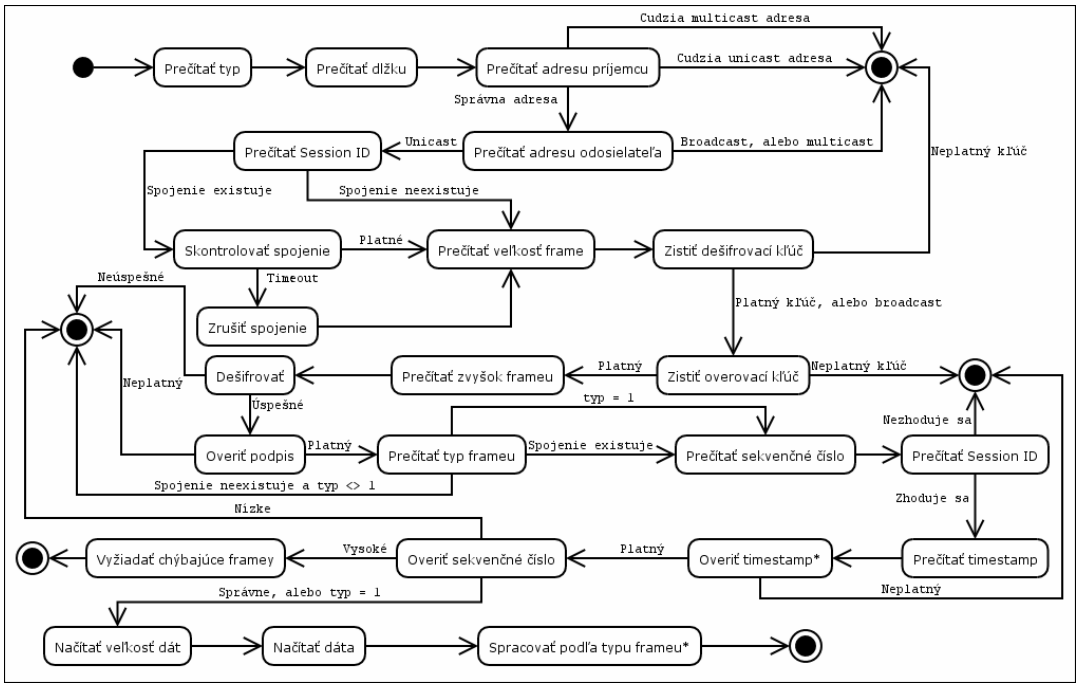
Dešifrovací kľúč určíme buď ako náš súkromný šifrovací kľúč (v prípade prvého adresného módu), alebo ho vyhľadáme v tabuľke podľa adresy odosielateľa a Session ID (druhý adresný mód). Ak sa nepodarí získať správny dešifrovací kľúč, frame ignorujeme (keďže ho nebudeme vedieť dešifrovať).

Každý frame je podpísaný svojim odosielateľom a podpis treba overovať. Kľúč na overovanie podpisu získame buď ako časť adresy odosielateľa (prvý adresný mód), alebo ho vyhľadáme v tabuľke podľa adresy odosielateľa a Session ID (druhý adresný mód). Ak sa nepodarí získať kľúč, frame ignorujeme, keďže nemáme možnosť overiť jeho pravosť.

Ak je frame šifrovaný, dešifrujeme ho. Potom overíme digitálny podpis: zrekonštruujeme pôvodný nešifrovaný frame, vypočítame jeho haš a porovnáme s digitálnym podpisom dešifrovaným pomocou kľúča na overenie podpisu. Ak sa nerovnajú, frame ignorujeme, keďže nie je autentický (buď ho niekto úmyselne sfalšoval, alebo sa poškodil počas prenosu, alebo sa frame nepodarilo správne dešifrovať - napr. zlý kľúč).

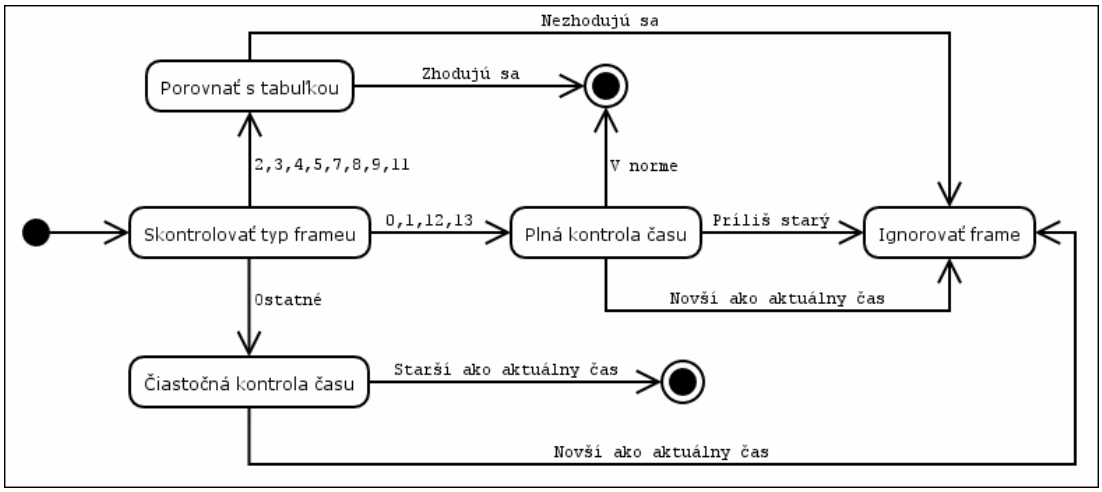
Ďalej čítame z už dešifrovaných dát typ framu. Ak typ framu nie je 1 (žiadosť o nadviazanie nového spojenia), skontrolujeme, či spojenie existuje. Ak nie, frame ignorujeme. Ďalej prečítame sekvenčné číslo a kópiu Session ID. Overíme, či sa Session ID zhoduje s tým v nešifrovanej časti hlavičky, ak nie, frame ignorujeme. Ďalej prečítame timestamp a vykonáme prvú kontrolu [Obr. 17]. Ak typ framu nie je 1, skontrolujeme či sekvenčné číslo súhlasí s tým, ktoré očakávame. Ak je sekvenčné

číslo staršie, frame ignorujeme (buď sa jedná o zablúdený frame, alebo pokus o replay útok). Ak je sekvenčné číslo vyššie ako očakávame, frame ignorujeme, ale od odosielateľa si vyžiadame poslať znovu chýbajúce framy. Nakoniec prečítame dĺžku dát a pomocou nej oddelíme dáta od výplne.



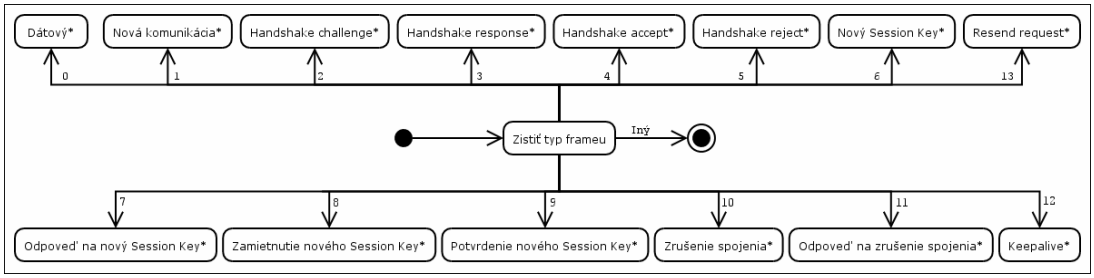
Obr. 16 Prijať

V závislosti od typu framu buď porovnáme timestamp s uloženou hodnotou v tabuľke spojení (typy 2-5,7-9,11), overíme vek timestampu (nesmie byť novší než aktuálny čas, ale ani starší než určená hodnota; typy 0,1,12,13), alebo overíme iba či timestamp nie je novší než aktuálny čas (ostatné typy). Ak timestamp nevyhovuje týmto kritériám, frame ignorujeme (pravdepodobne sa jedná o replay útok).



Obr. 17 Overiť timestamp

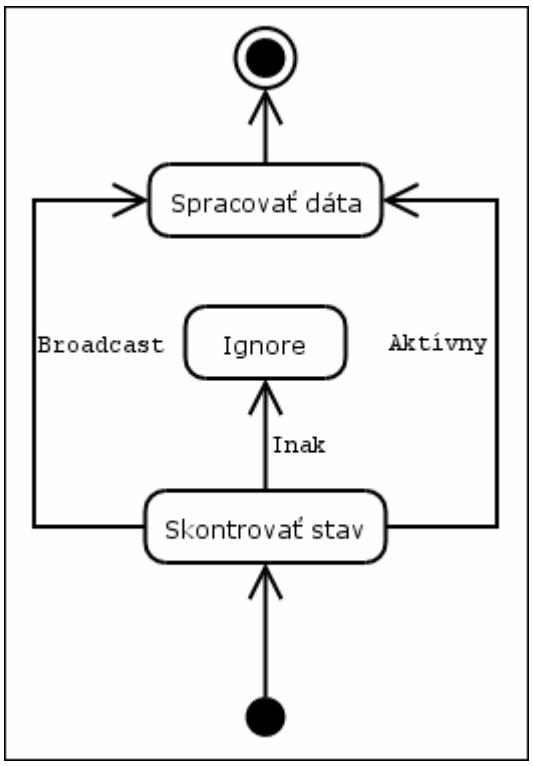
Frame ďalej spracujeme podľa typu framu. Ak sieť nepoužíva TTP na komunikáciu v druhom adresnom móde, treba ignorovať framy v druhom adresnom móde, ktoré nie sú dátové, keepalive, alebo resend request, prípadne framy na zmenu Session Key alebo na zrušenie spojenia. Ak sa používa TTP na komunikáciu v druhom adresnom móde, tak z riadiacich framov zostanú iba niektoré a tie sa používajú pri komunikácii s TTP a nie s bežným uzlom.



Obr. 18 Spracovať podľa typu framu

6.2.2.0 Dátový frame

Skontrolujeme, či frame patrí do existujúceho aktívneho spojenia, alebo bol broadcastovaný. Ak nie, frame ignorujeme. Ak áno, zaznamenáme čas prijatia framu a dáta posunieme vyššej vrstve na spracovanie.



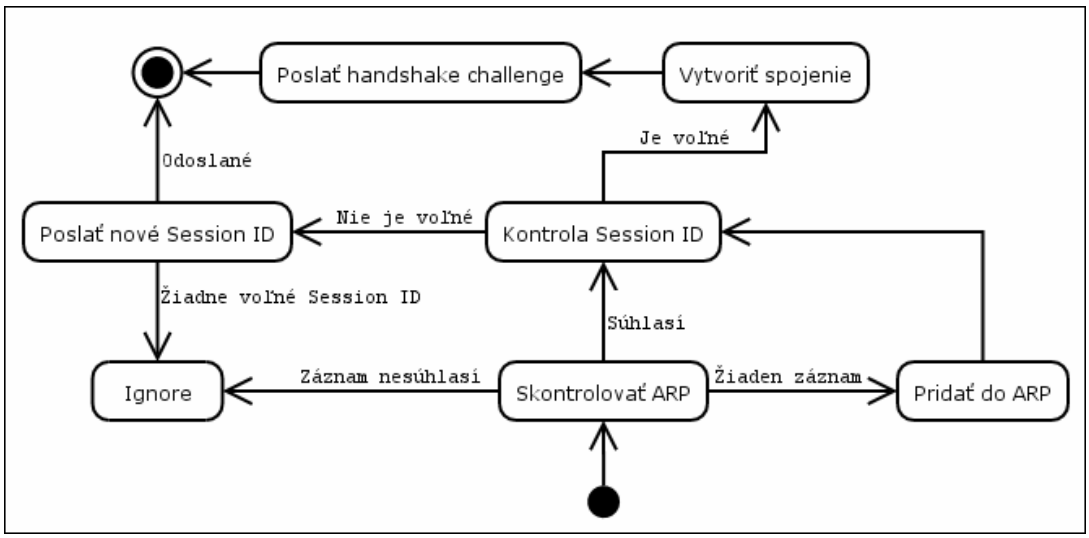
Obr. 19 Dátový

6.2.2.1 Začiatok novej komunikácie

Najprv skontrolujeme prijatú MAC adresu, či súhlasí s ARP tabuľkou. Ak v tabuľke nemáme záznam, pridáme ho. Ak záznam nesúhlasí, frame ignorujeme.

Ďalej skontrolujeme navrhované Session ID. Ak nie je voľné, vygenerujeme nové Session ID (ak nie je voľné žiadne, tak skončíme) a odošleme vlastnú žiadosť o začiatok novej komunikácie.

Ak bolo Session ID voľné, založíme záznam v tabuľke spojení a odošleme handshake challenge na dohodnutie Session Key (a taktiež overenie totožnosti odosielateľa).



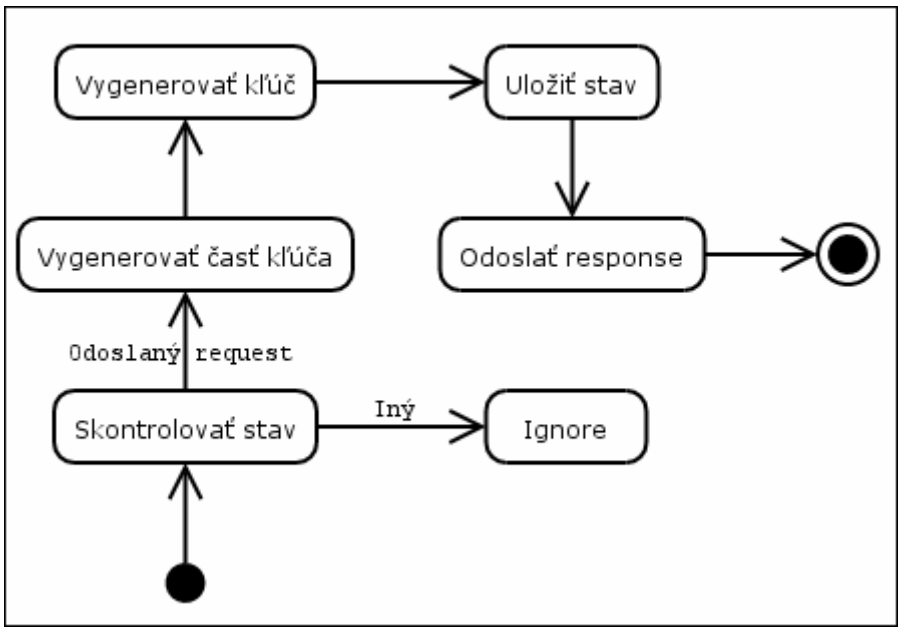
Obr. 20 Nová komunikácia

6.2.2.2 Handshake challenge

Najprv skontrolujeme, či existuje záznam pre toto spojenie a či bol odoslaný návrh na začiatok novej komunikácie. Ak nie, frame ignorujeme.

Prečítame z dát rnd1 a časť kľúča. Vygenerujeme svoju časť kľúča, spolu s prijatou časťou vygenerujeme úplný kľúč a uložíme ho.

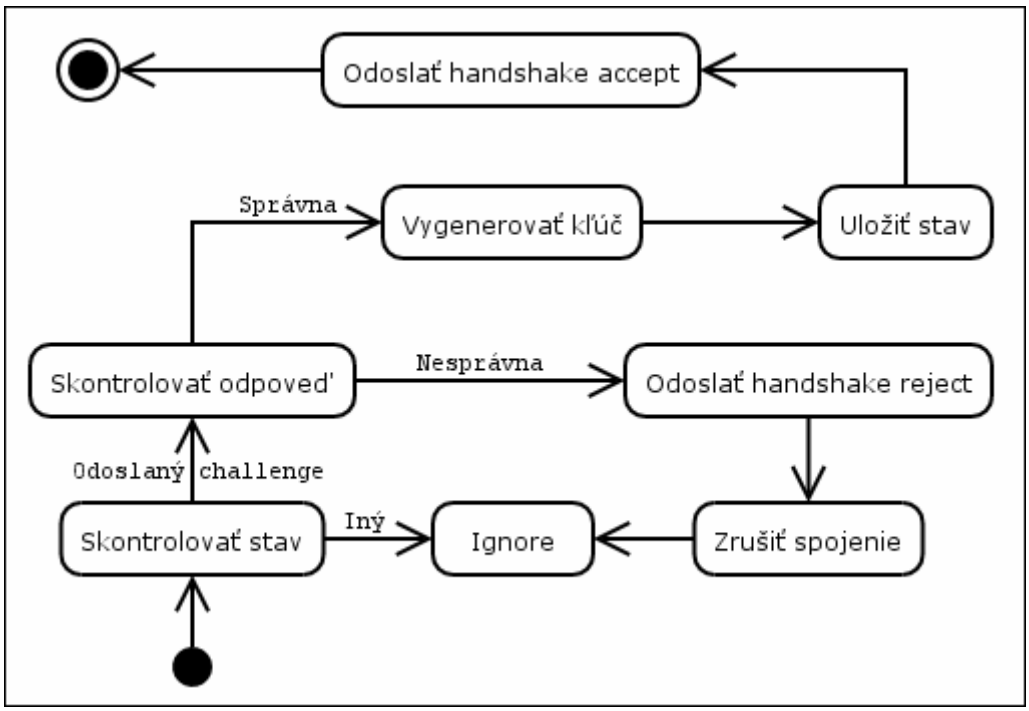
Odošleme handshake response s rnd1 + 1 a našou časťou kľúča.



Obr. 21 Handshake challenge

6.2.2.3 Handshake response

Skontrolujeme, či existuje príslušné spojenie a bol odoslaný handshake challenge. Ak nie, frame ignorujeme. Prečítame z dát rnd1 + 1 a časť kľúča. Skontrolujeme rnd1 a ak nesúhlasí s tým odoslaným odošleme handshake reject, zrušíme záznam o spojení a frame ignorujeme. Pomocou oboch častí kľúča vygenerujeme Session Key, uložíme ho a odošleme handshake accept frame s našou MAC adresou (2. adresný mód).



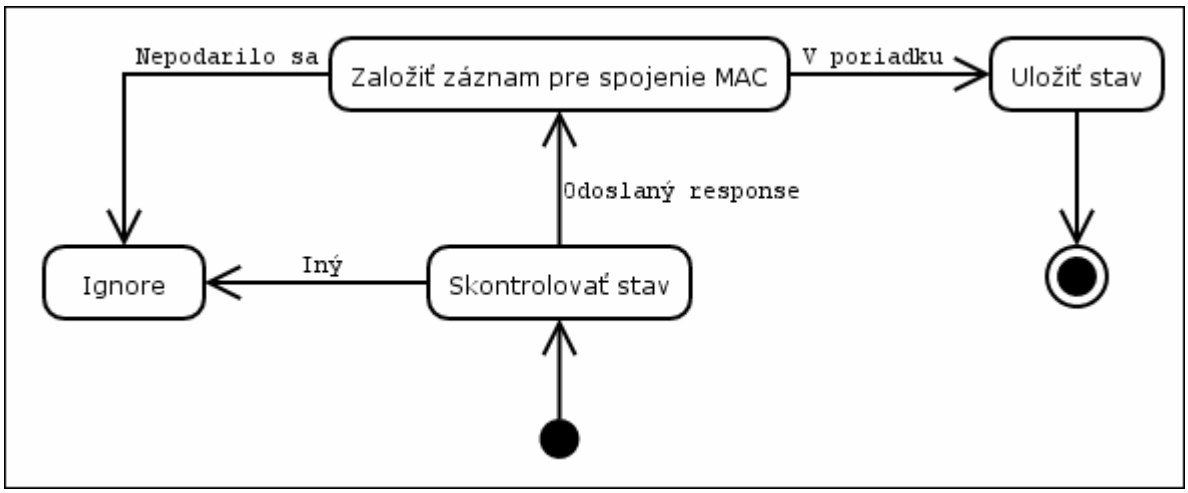
Obr. 22 Handshake response

6.2.2.4 Handshake accept

Najprv skontrolujeme stav spojenia, či existuje a bol odoslaný handshake response. Ak nie, frame ignorujeme.

Prečítame prijatú MAC adresu, skontrolujeme s ARP cache a založíme spojenie. Ak sa to nepodarí, frame ignorujeme.

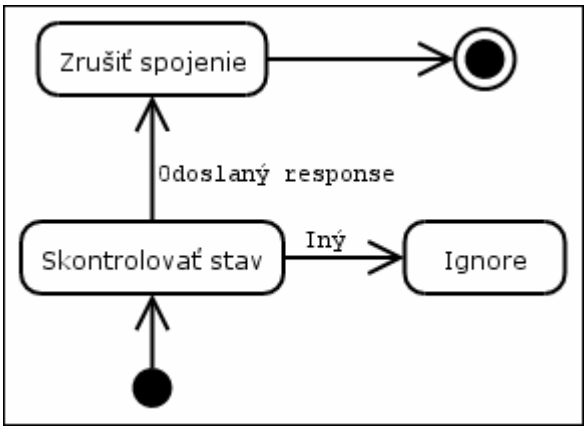
Nakoniec uložíme stav spojenia ako aktívny a nastavíme timestamp na aktuálny čas.



Obr. 23 Handshake accept

6.2.2.5 Handshake reject

Skontrolujeme, či spojenie existuje a bol odoslaný handshake response. Ak áno, zrušíme spojenie, inak frame ignorujeme.



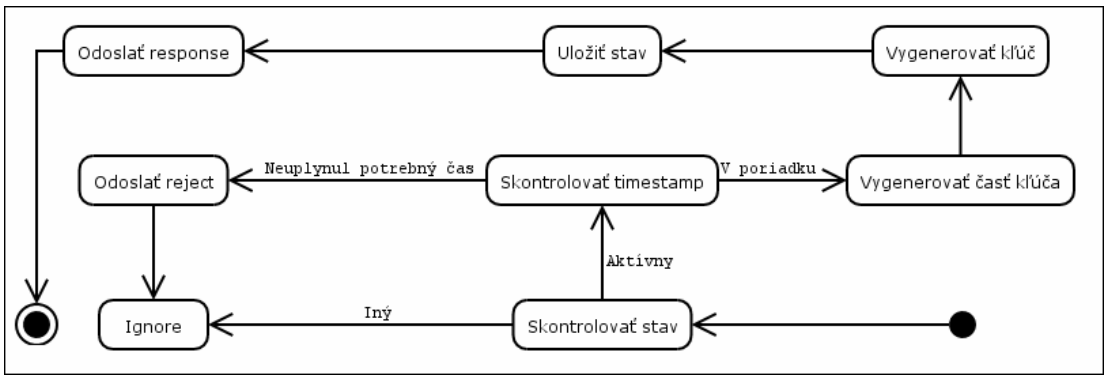
Obr. 24 Handshake reject

6.2.2.6 Žiadosť o nový Session Key

Skontrolujeme, či frame patrí do existujúceho aktívneho spojenia. Ak nie, frame ignorujeme.

Ďalej vykonáme podrobnejšiu kontrolu timestamp – od poslednej zmeny Session Key musel uplnynúť dostatočný čas. Ak neuplynul, frame ignorujeme.

Ak je timestamp v poriadku, vygenerujeme našu časť kľúča. Spolu s prečítanou časťou kľúča vygenerujeme celý kľúč, údaje uložíme a pošleme odpoveď obsahujúcu našu časť kľúča.

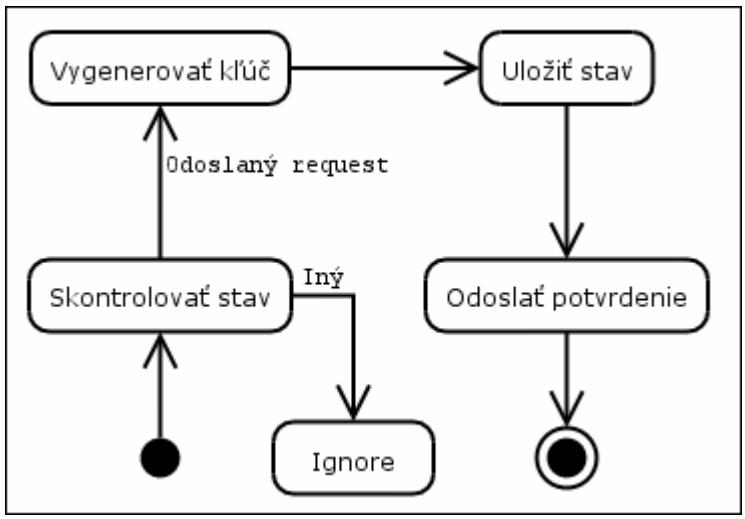


Obr. 25 Nový Session Key

6.2.2.7 Odpoveď na žiadosť o nový Session Key

Najprv skontrolujeme existenciu spojenia a jeho stav. Ak nebola odoslaná žiadosť o zmenu Session Key frame ignorujeme.

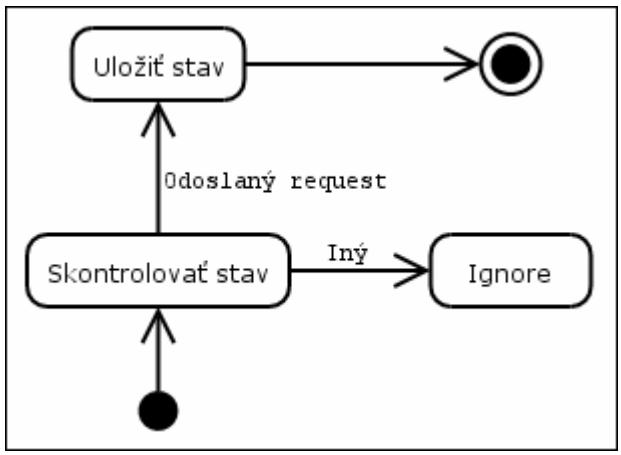
Prečítame z dát časť kľúča a spolu s našou časťou vygenerujeme nový Session Key. Aktualizujeme tabuľku spojení a odošleme potvrdenie.



Obr. 26 Odpoveď na nový Session Key

6.2.2.8 Zamietnutie žiadosti o nový Session Key

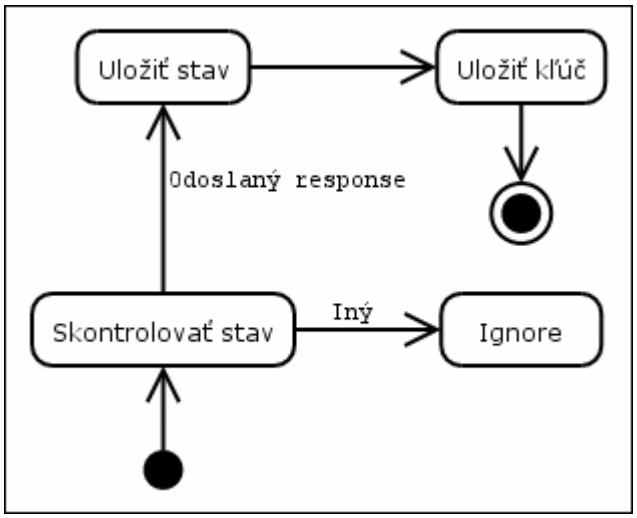
Skontrolujeme, či spojenie existuje a bola odoslaná žiadosť o zmenu kľúča. Ak áno, odstránime dočasné dáta uložené pri odoslaní žiadosti. Inak frame ignorujeme.



Obr. 27 Zamietnutie nového Session Key

6.2.2.9 Potvrdenie nového Session Key

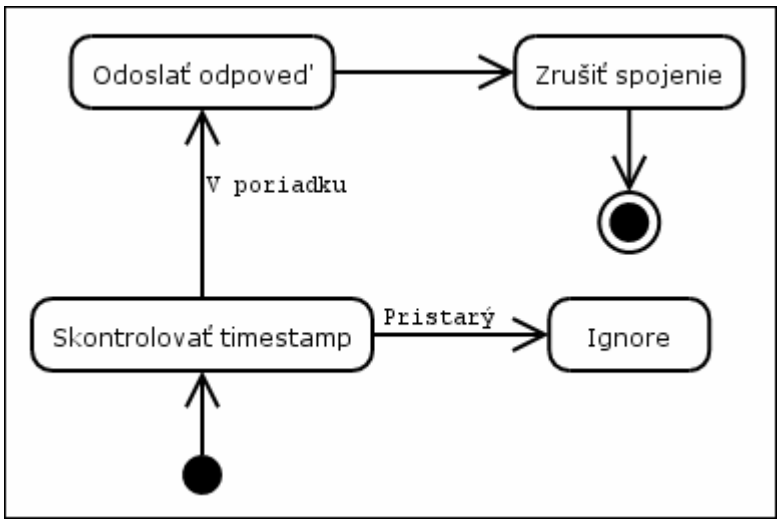
Overíme, že spojenie existuje a bola odoslaná odpoveď na žiadosť (ak nie, frame ignorujeme). Starý kľúč nahradíme novým a aktualizujeme tabuľku spojení.



Obr. 28 Potvrdenie nového Session Key

6.2.2.10 Žiadosť o zrušenie spojenia

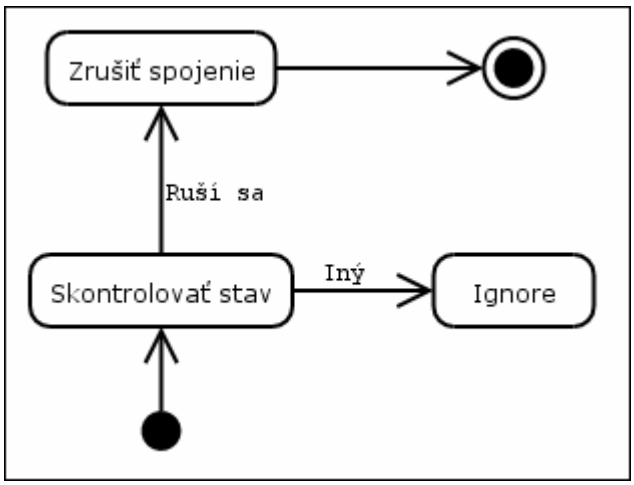
Skontrolujeme timestamp. Ak je príliš starý, frame ignorujeme (pravdepodobne replay). Inak odošleme odpoveď a spojenie zrušíme.



Obr. 29 Zrušenie spojenia

6.2.2.11 Odpoveď na žiadosť o zrušenie spojenia

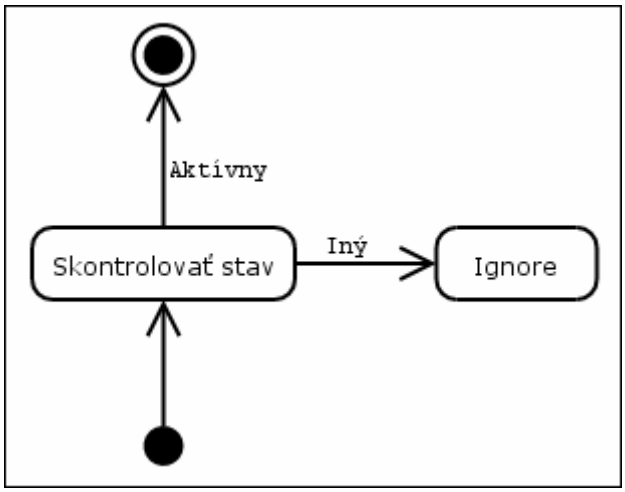
Skontrolujeme, či bola odoslaná požiadavka na zrušenie spojenia. Ak áno, spojenie zrušíme, inak frame ignorujeme.



Obr. 30 Odpoveď na zrušenie spojenia

6.2.2.12 Keepalive

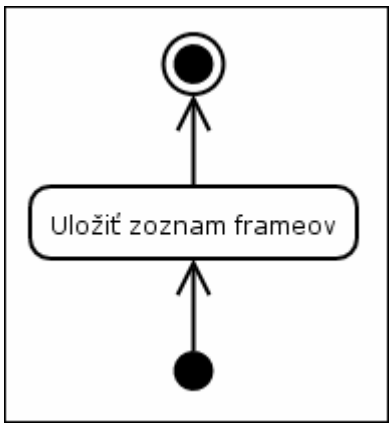
Treba skontrolovať stav spojenia. Ak je spojenie aktívne, nastavíme timestamp v tabuľke spojení na aktuálny čas.



Obr. 31 Keepalive

6.2.2.13 Resend request

Nastavíme požiadavku na odoslanie žiadaných framov. Po ukončení najbližšieho cyklu bude požiadavka spracovaná. Prípadne skontrolujeme, či boli žiadané framy naozaj odosielané (podľa sekvenčného čísla v tabuľke spojení).



Obr. 32 Resend request

7 Efektivita

7.1 Dátová efektivita

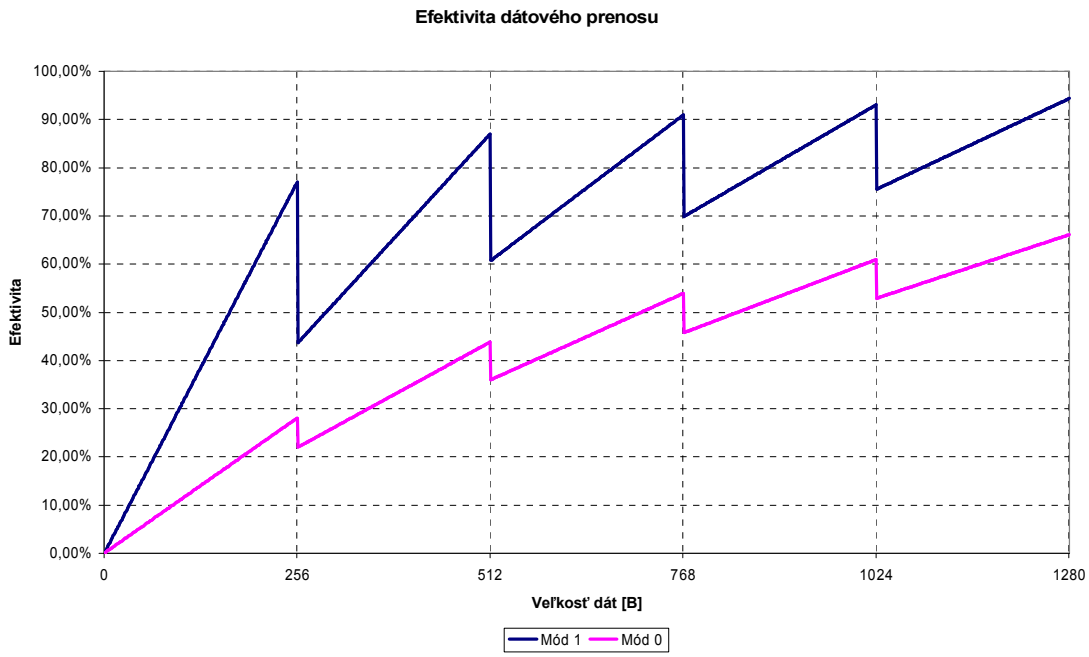
Zaujíma nás, koľko informácií navyše je potrebných na prenesenie určitého množstva dát.

Keďže chceme dáta doručiť správnemu adresátovi, je nutné k nim pribaliť riadiace informácie, čiže hlavičku. Veľkosť hlavičky závisí od adresného módu a dĺžky adresy. Dĺžka adresy závisí taktiež od adresného módu, ale aj od použitého šifrovacieho a podpisovacieho algoritmu. Všeobecne platí, že hlavička pre prvý adresný mód je (výrazne) dlhšia, ako pre druhý adresný mód.

Ďalšie informácie navyše prenášame v podobe digitálneho podpisu. Jeho veľkosť závisí od použitého algoritmu na podpisovanie a dĺžky podpisovacieho kľúča.

Taktiež väčšina šifrovacích algoritmov pracuje s blokmi dát, takže vyžadujú vstup istej pevnej dĺžky (alebo jej násobku). Ak dáta nemajú správnu dĺžku, je treba ich doplniť výplňou. Veľkosť výplne teda závisí od použitého šifrovacieho algoritmu a veľkosti bloku s ktorým tento algoritmus pracuje. Ak sa použijú prúdové šifry, výplň je zbytočná.

Ako príklad si zoberme 2048 bitové RSA šifrovanie pre prvý adresný mód a ľubovoľnú šifru s veľkosťou bloku 256 bajtov pre druhý adresný mód. Podpisovací algoritmus použijeme 320 bitový ECDSA (veľkosť súkromného kľúča je 160 bitov, veľkosť verejného kľúča aj výsledného podpisu je 320 bitov). V tomto prípade má hlavička v prvom adresnom móde veľkosť 616 octetov, podpis 40 octetov a dĺžka dát a výplne bude násobok 256 octetov. V druhom adresnom móde má hlavička veľkosť 36 octetov, podpis 40 octetov a dáta a výplň budú násobok 256 octetov. Minimálna veľkosť framu pre prvý adresný mód je teda 912 octetov, pre druhý adresný mód to je 332 octetov. Ak uvažujeme prostredie Ethernetu, tak veľkosti framov sú pre prvý adresný mód 912, 1168 a 1424 bajtov (pričom prenášame 0-768 bajtov dát). Ako vidíme, efektivita sa blíži k 50%. Pre druhý adresný mód sú veľkosti framov 332, 588, 844, 1100 a 1356 bajtov (prenáša sa 0-1280 bajtov dát). Efektivita sa v tomto prípade blíži k 95%. Samozrejme tu platí, že čím viac dát sa prenáša, tým vyššia je efektivita.



Obr. 33 Efektivita dátového prenosu

7.2 Výpočtová náročnosť

V tejto sekcii sa pokúsime pozrieť na výpočtovú zložitosť. Činnosti náročné na výpočty sú dve: šifrovanie a podpisovanie (resp. dešifrovanie a overovanie podpisu), ostatné činnosti sú približne rovnako náročné, ako pri bežných modeloch data-link layer. Ako pre šifrovanie, tak aj pre podpisovanie platí, že výpočtová náročnosť závisí od použitého algoritmu.

Pri šifrovaní napríklad vieme, že symetrické šifry sú menej náročné ako asymetrické šifry; z toho teda vieme, že spracovanie framov v druhom adresnom móde bude rýchlejšie ako v prvom. Preto model využíva prvý adresný mód hlavne na administráciu spojení a nie na prenos dát.

Pri podpisovaní nie je rozdiel medzi adresnými módmi, keďže obidva používajú rovnaký algoritmus.

Keďže výpočtová náročnosť úzko závisí od použitého algoritmu, uvádzame tu tabuľku porovnávajúcu najpoužívanejšie šifrovacie, podpisovacie a hašovacie algoritmy.

Náročnosť odosielania by sa dala odhadnúť ako čas potrebný na šifrovanie dát + čas potrebný na haš dát + čas potrebný na vygenerovanie podpisu z hašu (pri niektorých algoritmoch je hašovanie zahrnuté do podpisovania). Pri prijímaní framu je to čas

potrebný na dešifrovanie + čas potrebný na haš dát + čas potrebný na dešifrovanie podpisu.

Ak by sa používali špecializované čipy pre konkrétne algoritmy a v maximálnej miere využil paralelizmus (napr. paralelne šifrovať prvé bloky dát a vytvárať podpis, ak je to možné), bolo by možné dosahovať pri symetrických blokových šifrách rýchlosti rádovo 100 Mbit/s, pri prúdových rádovo 1Gbit/s. Pri asymetrických šifrách rýchlosti rádovo 100-1000 kbit/s.

Algoritmus	Pentium 4 2.6GHz WinXP				AMD Opteron 1.6GHz Linux			
	Dáta [MB]	Čas [s]	MB/s	Mbit/s	Dáta [MB]	Čas [s]	MB/s	Mbit/s
CRC-32	2050	6,399	320,36	**2687,40**	1020	3,55	287,32	**2410,25**
Adler-32	4100	3,525	1163,12	**9756,96**	4100	5,14	797,67	**6691,30**
MD2	16	4,006	3,99	**33,50**	16	5,32	3,01	**25,23**
MD5	1020	4,726	215,83	**1810,49**	512	3,36	152,38	**1278,26**
SHA-1	256	3,766	67,98	**570,23**	512	5,11	100,20	**840,50**
SHA-256	256	5,758	44,46	**372,96**	256	4,38	58,45	**490,29**
SHA-512	64	5,618	11,39	**95,56**	512	5,52	92,75	**778,07**
HAVAL (pass=3)	512	4,717	108,54	**910,53**	512	3,71	138,01	**1157,67**
HAVAL (pass=4)	256	3,695	69,28	**581,19**	512	4,97	103,02	**864,18**
HAVAL (pass=5)	256	3,796	67,44	**565,72**	512	6,22	82,32	**690,51**
Tiger	128	3,364	38,05	**319,19**	1020	4,88	209,02	**1753,36**
RIPE-MD160	256	4,867	52,60	**441,23**	256	3,69	69,38	**581,97**
Panama Hash LE	1020	3,375	302,22	**2535,22**	2050	4,85	422,68	**3545,70**
Panama Hash BE	1020	4,637	219,97	**1845,24**	2050	6,47	316,85	**2657,91**
Whirlpool	64	5,288	12,10	**101,53**	128	5,38	23,79	**199,58**
MDC/MD5	256	5,377	47,61	**399,38**	128	3,57	35,85	**300,77**
Luby-Rackoff/MD5	64	4,307	14,86	**124,65**	64	5,01	12,77	**107,16**

Tab. 9 rýchlosť hašovacích funkcií

Algoritmus	Pentium 4 2.6GHz WinXP				AMD Opteron 1.6GHz Linux			
	Dáta [MB]	Čas [s]	MB/s	Mbit/s	Dáta [MB]	Čas [s]	MB/s	Mbit/s
ARC4	512	4,517	113,35	**950,85**	512	5,74	89,20	**748,25**
SEAL-3,0-BE	1020	3,485	292,68	**2455,20**	1020	4,91	207,74	**1742,64**
SEAL-3,0-LE	2050	4,937	415,23	**3483,22**	1020	4,71	216,56	**1816,64**
WAKE-CFB-BE	512	5,498	93,12	**781,19**	512	3,43	149,27	**1252,18**
WAKE-CFB-LE	512	3,615	141,63	**1188,10**	1020	5,57	183,12	**1536,15**
WAKE-OFB-BE	512	3,855	132,81	**1114,13**	1020	6,21	164,25	**1377,84**
WAKE-OFB-LE	512	3,836	133,47	**1119,65**	1020	5,89	173,17	**1452,70**
Panama Cipher LE	1020	4,036	252,73	**2120,01**	2050	5,94	345,12	**2895,06**
Panama Cipher BE	1020	5,317	191,84	**1609,25**	2050	6,46	317,34	**2662,02**

Tab. 10 rýchlosť symetrických prúdových šifier

Algoritmus	Pentium 4 2.6GHz WinXP				AMD Opteron 1.6GHz Linux			
	Dáta [MB]	Čas [s]	MB/s	Mbit/s	Dáta [MB]	Čas [s]	MB/s	Mbit/s
DES	128	5,998	21,34	**179,02**	128	5,22	24,52	**205,70**
DES-XEX3	128	6,159	20,78	**174,34**	128	5,53	23,15	**194,17**
DES-EDE3	64	6,499	9,85	**82,61**	32	3,5	9,14	**76,70**
IDEA	64	3,375	18,96	**159,07**	128	5,38	23,79	**199,58**
RC2	64	5,548	11,54	**96,77**	64	5,02	12,75	**106,95**
RC5 (r=16)	256	4,286	59,73	**501,05**	256	5,21	49,14	**412,18**
Blowfish	256	3,976	64,39	**540,11**	128	3,7	34,59	**290,20**
3-WAY	128	3,665	34,92	**292,97**	128	3,47	36,89	**309,44**
TEA	128	5,378	23,80	**199,65**	128	6,14	20,85	**174,88**
SAFER (r=8)	128	6,279	20,39	**171,01**	64	3,77	16,98	**142,41**
GOST	128	3,505	36,52	**306,35**	128	4,59	27,89	**233,93**
SHARK (r=6)	128	3,826	33,46	**280,64**	256	5,91	43,32	**363,36**
CAST-128	256	5,988	42,75	**358,63**	256	5,02	51,00	**427,79**
CAST-256	128	5,889	21,74	**182,33**	128	3,46	36,99	**310,33**
Square	128	4,176	30,65	**257,12**	256	4,19	61,10	**512,53**
SKIPJACK	128	6,329	20,22	**169,65**	64	4,91	13,03	**109,34**
RC6	128	3,385	37,81	**317,21**	256	4,23	60,52	**507,68**
MARS	128	4,586	27,91	**234,13**	256	4,4	58,18	**488,06**
Rijndael (128-bit)	256	4,196	61,01	**511,79**	256	5,16	49,61	**416,18**
Rijndael (192-bit)	256	4,817	53,15	**445,81**	256	5,99	42,74	**358,51**
Rijndael (256-bit)	256	5,308	48,23	**404,57**	128	3,4	37,65	**315,81**
Rijndael (128) CTR	256	4,436	57,71	**484,10**	256	5,22	49,04	**411,40**
Rijndael (128) OFB	256	4,837	52,93	**443,97**	256	5,52	46,38	**389,04**
Rijndael (128) CFB	256	5,378	47,60	**399,31**	256	6,16	41,56	**348,62**
Rijndael (128) CBC	256	4,617	55,45	**465,13**	256	5,51	46,46	**389,74**
Twofish	128	4,075	31,41	**263,49**	256	4,57	56,02	**469,91**
Serpent	128	6,069	21,09	**176,92**	128	3,59	35,65	**299,09**
SHACAL-2 (128-bit)	128	6,279	20,39	**171,01**	256	5,45	46,97	**394,03**
SHACAL-2 (512-bit)	128	6,279	20,39	**171,01**	256	5,45	46,97	**394,03**
Camellia (128-bit)	64	3,355	19,08	**160,02**	128	4,22	30,33	**254,44**
Camellia (192-bit)	64	4,437	14,42	**121,00**	128	5,47	23,40	**196,30**
Camellia (256-bit)	64	4,416	14,49	**121,57**	128	5,48	23,36	**195,94**

Tab. 11 rýchlosť symetrických blokových šifier

Dešifrovanie Algoritmus	Pentium 4 2.6GHz WinXP			AMD Opteron 1.6GHz Linux		
	Iterácií	Čas [s]	Operácia [ms]	Iterácií	Čas [s]	Operácia [ms]
RSA 1024	1050	5,007	**4,77**	2510	5,000	**1,99**
Rabin 1024	795	5,007	**6,30**	2082	5,000	**2,40**
LUC 1024	634	5,007	**7,90**	1539	5,000	**3,25**
DLIES 1024	350	5,008	**14,31**	1001	5,000	**5,00**
LUCELG 512	2744	5,007	**1,82**	7123	5,000	**0,70**
RSA 2048	177	5,028	**28,41**	491	5,000	**10,18**
Rabin 2048	159	5,027	**31,62**	437	5,000	**11,44**
LUC 2048	110	5,037	**45,79**	302	5,010	**16,59**
DLIES 2048	60	5,027	**83,78**	165	5,000	**30,30**
LUCELG 1024	772	5,007	**6,49**	2186	5,000	**2,29**
ECIES over GF(p) 168	1062	5,007	**4,71**	2206	5,000	**2,27**
ECIES over GF(2^n) 155	633	5,007	**7,91**	1137	5,000	**4,40**

Tab. 12 rýchlosť dešifrovania asymetrických šifier

*(pozn. * znamená použitie predpočítaných údajov)*

Šifrovanie	Pentium 4 2.6GHz WinXP			AMD Opteron 1.6GHz Linux		
Algoritmus	Iterácií	Čas [s]	Operácia [ms]	Iterácií	Čas [s]	Operácia [ms]
RSA 1024	27607	5,007	**0,18**	69170	5,000	**0,07**
Rabin 1024	3008	5,007	**1,66**	8248	5,000	**0,61**
LUC 1024	23544	5,008	**0,21**	58983	5,000	**0,08**
DLIES 1024	1159	5,007	**4,32**	3210	5,000	**1,56**
DLIES 1024*	1061	5,007	**4,72**	2911	5,000	**1,72**
LUCELG 512	2433	5,007	**2,06**	5700	5,000	**0,88**
LUCELG 512*	2402	5,007	**2,08**	5688	5,000	**0,88**
RSA 2048	11022	5,007	**0,45**	29518	5,000	**0,17**
Rabin 2048	1254	5,007	**3,99**	3477	5,000	**1,44**
LUC 2048	8756	5,007	**0,57**	24129	5,000	**0,21**
DLIES 2048	260	5,007	**19,26**	706	5,000	**7,08**
DLIES 2048*	269	5,017	**18,65**	740	5,000	**6,76**
LUCELG 1024	532	5,008	**9,41**	1456	5,000	**3,43**
LUCELG 1024*	531	5,007	**9,43**	1459	5,000	**3,43**
ECIES over GF(p) 168	759	5,008	**6,60**	1667	5,000	**3,00**
ECIES over GF(p) 168*	1327	5,007	**3,77**	2332	5,000	**2,14**
ECIES over GF(2^n) 155	414	5,007	**12,09**	699	5,000	**7,15**
ECIES over GF(2^n) 155*	1071	5,008	**4,68**	2273	5,000	**2,20**

Tab. 13 rýchlosť šifrovania asymetrických šifier

Podpis	Pentium 4 2.6GHz WinXP			AMD Opteron 1.6GHz Linux		
Algoritmus	Iterácií	Čas [s]	Operácia [ms]	Iterácií	Čas [s]	Operácia [ms]
RSA 1024	1053	5,007	**4,75**	2415	5,000	**2,07**
Rabin 1024	816	5,008	**6,14**	2018	5,000	**2,48**
RW 1024	951	5,007	**5,26**	2335	5,000	**2,14**
LUC 1024	644	5,007	**7,77**	1542	5,000	**3,24**
NR 1024	2260	5,007	**2,22**	6205	5,000	**0,81**
NR 1024*	4376	5,007	**1,14**	10231	5,000	**0,49**
DSA 1024	2301	5,008	**2,18**	6251	5,000	**0,80**
DSA 1024*	4444	5,007	**1,13**	10332	5,000	**0,48**
LUC-HMP 512	2365	5,007	**2,12**	5732	5,000	**0,87**
LUC-HMP 512*	2401	5,008	**2,09**	5676	5,000	**0,88**
ESIGN 1023	9837	5,007	**0,51**	23513	5,000	**0,21**
ESIGN 1536	4898	5,008	**1,02**	12643	5,000	**0,40**
RSA 2048	178	5,007	**28,13**	495	5,000	**10,10**
Rabin 2048	158	5,028	**31,82**	439	5,000	**11,39**
RW 2048	176	5,017	**28,51**	497	5,000	**10,06**
LUC 2048	110	5,007	**45,52**	303	5,000	**16,50**
NR 2048	511	5,007	**9,80**	1412	5,000	**3,54**
NR 2048*	1530	5,007	**3,27**	3818	5,000	**1,31**
LUC-HMP 1024	522	5,007	**9,59**	1474	5,000	**3,39**
LUC-HMP 1024*	527	5,007	**9,50**	1465	5,000	**3,41**
ESIGN 2046	4253	5,008	**1,18**	10363	5,000	**0,48**
ECNR over GF(p) 168	1497	5,007	**3,34**	3366	5,000	**1,49**
ECNR over GF(p) 168*	2629	5,007	**1,90**	4797	5,000	**1,04**
ECNR over GF(2^n) 155	827	5,007	**6,05**	1380	5,000	**3,62**
ECNR over GF(2^n) 155*	2132	5,007	**2,35**	4398	5,000	**1,14**

Tab. 14 rýchlosť digitálneho podpisovania

Overenie	Pentium 4 2.6GHz WinXP			AMD Opteron 1.6GHz Linux		
Algoritmus	**Iterácií**	**Čas [s]**	**Operácia [ms]**	**Iterácií**	**Čas [s]**	**Operácia [ms]**
RSA 1024	27406	5,007	**0,18**	67424	5,000	**0,07**
Rabin 1024	3117	5,007	**1,61**	8099	5,000	**0,62**
RW 1024	55580	5,007	**0,09**	124746	5,000	**0,04**
LUC 1024	24321	5,008	**0,21**	61335	5,000	**0,08**
NR 1024	1964	5,007	**2,55**	5451	5,000	**0,92**
NR 1024*	2738	5,007	**1,83**	6335	5,000	**0,79**
DSA 1024	2007	5,007	**2,49**	5498	5,000	**0,91**
DSA 1024*	2795	5,007	**1,79**	6450	5,000	**0,78**
LUC-HMP 512	2312	5,007	**2,17**	5655	5,000	**0,88**
LUC-HMP 512*	2269	5,007	**2,21**	5702	5,000	**0,88**
ESIGN 1023	27679	5,007	**0,18**	75703	5,000	**0,07**
ESIGN 1536	11940	5,007	**0,42**	33692	5,000	**0,15**
RSA 2048	11054	5,007	**0,45**	29297	5,000	**0,17**
Rabin 2048	1294	5,007	**3,87**	3464	5,000	**1,44**
RW 2048	24521	5,007	**0,20**	65328	5,000	**0,08**
LUC 2048	9072	5,008	**0,55**	24478	5,000	**0,20**
NR 2048	454	5,007	**11,03**	1249	5,000	**4,00**
NR 2048*	982	5,008	**5,10**	2311	5,000	**2,16**
LUC-HMP 1024	520	5,007	**9,63**	1459	5,000	**3,43**
LUC-HMP 1024*	499	5,007	**10,03**	1464	5,000	**3,42**
ESIGN 2046	11024	5,007	**0,45**	32473	5,000	**0,15**
ECNR over GF(p) 168	794	5,008	**6,31**	1240	5,000	**4,03**
ECNR over GF(p) 168*	1618	5,007	**3,09**	2947	5,000	**1,70**
ECNR over GF(2^n) 155	655	5,007	**7,64**	1123	5,000	**4,45**
ECNR over GF(2^n) 155*	1232	5,008	**4,06**	2473	5,000	**2,02**

Tab. 15 rýchlosť overovania digitálneho podpisu

8 Záver

Problematika bezpečnosti na úrovni data-link layer nie je v súčasnosti veľmi rozšírená. Pri bezpečnosti sa kladie dôraz skôr na vyššie vrstvy. Jedinú výnimku tvoria bezdrôtové siete, ale aj u nich je bezpečnosť závislá od nastavenia konkrétnej siete a teda často býva nedostačujúca až žiadna.

Za hlavný prínos tejto práce považujeme navrhnutý model. Snažili sme sa model spraviť flexibilný, ale na druhú stranu s istou zaručenou bezpečnosťou bez ohľadu na konkrétne nastavenie.

Prípadné rozširenia tejto práce môžu byť nasledujúce:

- implementácia modelu
    1. špeciálne zariadenia

        vlastné sieťové karty špeciálne navrhnuté pre náš model
    2. adaptéry

        zariadena zapojené medzi existujúcu sieťovú kartu a prenosové médium, transformujúce bežnú sieťovú komunikáciu na komunikáciu nášho modelu
    3. softwareové riešenie

        úprava firmwareu a driverov pre sieťové karty tak, aby pracovali v súlade s našim modelom
- úprava modelu na koexistenciu s používanými technológiami

    upraviť štruktúru framov, povoliť nešifrovanú a nepodpisovanú komunikáciu; pravdepodobne by to vyžadovalo TTP na preklad adresných módov

9 Zoznam použitej literatúry

[1] wikipedia.org, rôzne články, Október 2006

http://en.wikipedia.org/wiki/Block_cipher,
http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation,
http://en.wikipedia.org/wiki/Cryptographic_hash_function,
http://en.wikipedia.org/wiki/Digital_signature,
http://en.wikipedia.org/wiki/Electronic_signature,
http://en.wikipedia.org/wiki/Encryption,
http://en.wikipedia.org/wiki/Hash_function,
http://en.wikipedia.org/wiki/Internet_protocol_suite,
http://en.wikipedia.org/wiki/Message_authentication_code,
http://en.wikipedia.org/wiki/OSI_model,
http://en.wikipedia.org/wiki/Open_Systems_Interconnection,
http://en.wikipedia.org/wiki/Public_key_certificate,
http://en.wikipedia.org/wiki/Stream_cipher,
http://en.wikipedia.org/wiki/Stream_cipher_attack,

[2] Menezes, Alfred J., van Oorschot, Paul C., Vanstone, Scott A., *Handbook of Applied Cryptography*, CRC Press, Október 1996, ISBN: 0-8493-8523-7, http://www.cacr.math.uwaterloo.ca/hac/

[3] ISO/IEC 7498-1:1994, *Information technology - Open Systems Interconnection - Basic Reference Model: The Basic Model*,

http://standards.iso.org/ittf/PubliclyAvailableStandards/s020269_ISO_IEC_7498-1_1994(E).zip

[4] ISO 7498-2:1989 *Information processing systems - Open Systems Interconnection - Basic Reference Model - Part 2: Security Architecture*

[5] IEEE Std 802.10-1998 *IEEE Standards for Local and Metropolitan Area Networks: Standard for Interoperable LAN/MAN Security (SILS)*, September 1998, ISBN 0-7381-1419-7, http://standards.ieee.org/getieee802/download/802.10-1998.pdf

[6] IEEE Std 802.1X-2001, *IEEE Standard for Local and metropolitan area networks- Port-Based Network Access Control*, ISBN 0-7381-2927-5, http://standards.ieee.org/getieee802/download/802.1X-2001.pdf

[7] RFC 2284, *PPP Extensible Authentication Protocol (EAP)*, Marec 1998 http://www.ietf.org/rfc/rfc2284.txt

[8] Pernecký, M. *Security of wireless networks based on ANSI/IEEE 802.11*, 2005 diplomová práca na FMFI UK

http://www.dcs.fmph.uniba.sk/diplomovky/obhajene/Detail.php?id=35

[9] Soriano, M. et al. *A Particular Solution to Provide Secure Communications in an Ethernet Environment.* ACM Conference on Computer and Communications Security 1993: p.17-25, http://portal.acm.org/citation.cfm?doid=168588.168591

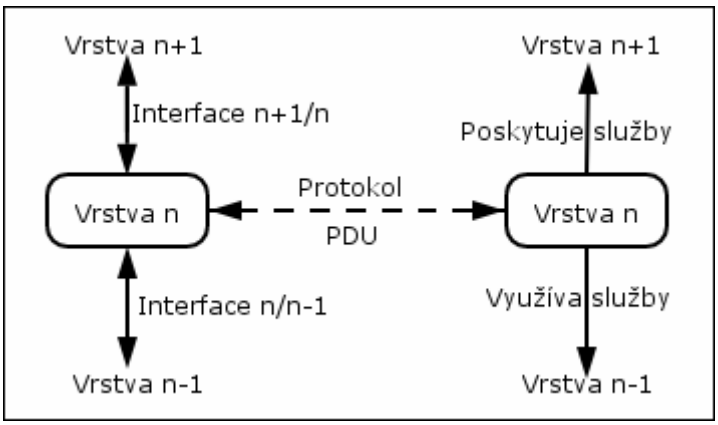
[10] Wei Dai, *Crypto++ 5.2.1 Benchmarks*, Speed Comparison of Popular Crypto Algorithms, 2004, http://www.eskimo.com/~weidai/benchmarks.html,

http://www.eskimo.com/~weidai/amd64-benchmarks.html

(elektronické verzie použitej literatúry, ktoré sú voľne šíriteľné, sú priložené na CD)
na CD je taktiež priložený pseudokód popisujúci činnosť modelu (verzia bez TTP)

Príloha A OSI model

ISO Open Systems Interconnection Reference Model je abstraktný popis komunikácie a designu počítačových sietí. Štandardizovaný bol v ISO 7498:1984.
Hlavná myšlienka je rozdeliť funkcie medzi vrstvy. Každá vrstva využíva iba funkcie vrstvy pod sebou a poskytuje funkcionalitu len vrstve nad sebou.



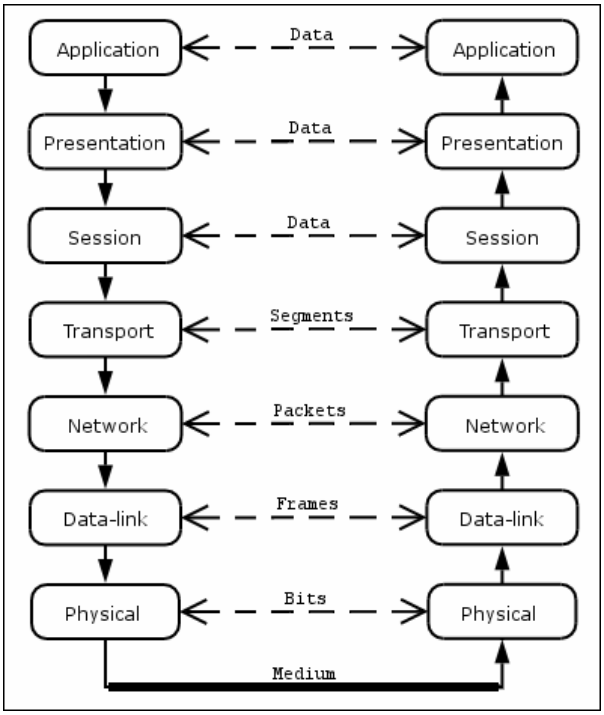
Obr. 34 n-tá vrstva vrstvového modelu

OSI je abstraktný model, takže jeho hlavnú časť tvorí popis komunikácie medzi jednotlivými vrstvami, takže ak sa tieto špecifikácie dodržiavajú, rôzne vrstvy od rôznych výrobcou budú spolu komunikovať správne.
V praxi sa tento model neujal, keďže je podľa niektorých príliš zložitý a navyše prišiel v čase, keď už sa rozmáhal TCP/IP model.

Application	Application
Presentation	
Session	
Transport	Transport
Network	Network
Data-link	Link
Physical	
OSI model	TCP/IP model

Tab. 16 OSI vs. TCP/IP

OSI model je rozdelený na 7 vrstiev: Physical, Data-link, Network, Transport, Session, Presentation a Application. Každá vrstva komunikuje so svojou partnerskou vrstvou prostredníctvom PDU a využíva pri tom služby vrstvy bezprostredne pod ňou.



Obr. 35 OSI model - komunikácia

A.1 Physical Layer

Fyzická vrstva sa zaoberá prenášaním údajov na bitovej úrovni. Zabezpečuje aby ak jedna strana odošle bitovú 1, druhá strana prijme bitovú 1 a nie bitovú 0.

Fyzická vrstva definuje všetky elektrické a fyzické špecifikácie zariadení. Typické otázky sú koľko voltov reprezentuje bitovú 1 a 0, ako dlho trvá vysielanie jedného bitu, či sa môže vysielať súčasne oboma smermi, ako sa zakladá a ruší spojenie, alebo koľko pinov má sieťový konektor a na čo sa ktorý pin používa.

Taktiež má táto vrstva na starosti efektivitu komunikácie: zdieľanie zdrojov medzi viac užívateľov, riadenie toku, riešenie zahltenia, kolízie a podobne.

Príklady: RS-232, 10BASE-T, 100BASE-TX, ISDN, SONET, ...

A.2 Data-link Layer

Linková vrstva má za úlohu zabezpečiť transfer dát medzi uzlami v jednej sieti, prípadne aj detekciu a opravu chýb, ktoré vznikli na fyzickej vrstve. Adresná schéma je fyzická, adresy sú pevné a nemusia mať žiadnu logickú štruktúru (žiadna hierarchia adries).

Táto vrstva sa skladá z dvoch podvrstiev, MAC podvrstva a LLC podvrstva.

Príklady: Ethernet, FDDI, Wireless (IEEE 802.11), Token ring, Frame Relay, ...

A.2.1 Media Access Control Sublayer

Zaoberá sa štruktúrou framu, rozpoznáva hranice framov, určuje prístup k médiu (priority, kolízie). Ďalej poskytuje detekciu prenosových chýb a adresáciu (vrátane filtrovania prijatých framov podľa adresy prijímateľa).
Príklad: časti niektorých data-link layer štandardov

A.2.2 Logical Link Control Sublayer

Zaoberá sa multiplexovaním a demultiplexovaním protokolov vyšších vrstiev, prípadne kontrolou toku a (na požiadanie) retransmisiou stratených framov.
Príklady: IEEE 802.2, časti niektorých data-link layer štandardov

A.3 Network Layer

Sieťová vrstva poskytuje funkcionálne a procedurálne možnosti ako prenášať údaje (rôznej dĺžky) zo zdroja do cieľa cez jednu alebo viac sietí a udržiavať pri tom kvalitu služby požadovanú vyššou vrstvou. Táto vrstva vykonáva routing, kontrolu toku, segmentáciu (a desegmentáciu) a prípadne aj kontrolu chýb. Adresná schéma na tejto úrovni je logická, konkrétne adresy určuje administrátor. Adresácia je hierarchická.
Najznámejší príklad je IP, taktiež niektoré routovacie protokoly (RIP, OSPF, ...).

A.4 Transport Layer

Účelom transportnej vrstvy je poskytovať transparentný prenos dát medzi koncovými používateľmi, čim odbremeňuje vyššie vrstvy od nutnosti poskytovania spoľahlivého a efektívneho dátového prenosu. Transportná vrstva má na starosti spoľahlivosť daného spojenia. Doručuje informácie bez chýb a v správnom poradí. Na dosiahnutie tohoto používa kontrolu chýb (s preposlaním chybne prijatých/nedoručených dát) a kontrolu toku dát.
Najznámejšie príklady sú TCP a UDP.

A.5 Session Layer

Relačná vrstva poskytuje mechanizmus správy dialógu medzi aplikačnými procesmi koncového používateľa. Poskytuje buď duplexnú alebo poloduplexnú komunikáciu, zodpovedá za checkpointing, odloženie, ukončenie a reštart spojenia. V niektorých prípadoch zabezpečuje správu tokenov (ak nie je možné komunikovať súčasne oboma smermi).

A.6 Presentation Layer

Prezentačná vrstva odbremeňuje aplikačnú vrstvu od starostí s rozdielnou syntaktickou reprezentáciou dát v rámci systému koncového používateľa. MIME kódovanie, kryptovanie a podobná manipulácia a reprezentácia dát sa odohráva na šiestej vrstve. Príkladom prezentačnej služby je konverzia súboru s EBCDIC kódovaním na kódovanie ASCII.

A.7 Application Layer

Táto vrstva implementuje rozhranie pre aplikačné procesy a poskytuje im služby. To umožňuje komunikáciu medzi aplikáciami od rôznych výrobcov.
Príklady: FTP, HTTP, POP3, SMTP, telnet, ...

Príloha B Šifrovanie

B.1 Šifrovanie

B.1.1 Úvod

Šifrovanie je proces, ako urobiť dáta nečitateľné bez znalosti špeciálnych (tajných) informácií. Šifrovanie dokáže dáta utajiť, ale pre skutočne bezpečnú komunikáciu sú potrebné aj iné techniky: na zabezpečenie integrity a autentickosti.

Šifra je algoritmus na vykonávanie šifrovania (a dešifrovania). Operácia väčšinou závisí od nejakej ďalšej informácie – kľúča. V závislosti od kľúča sa menia detaily operácie algoritmu a teda šifrovanie iným kľúčom dá iný výsledok. Princípom šifrovania je, že bez správneho kľúča by malo byť ťažké (až nemožné) správne dešifrovať zašifrované dáta. Šifier je niekoľko typov:

B.1.2 Symetrické šifry

Symetrické šifry sú také šifry, ktorých šifrovací a dešifrovací kľúč sú na sebe triviálne závislé (tj. buď sú zhodné, alebo sa dajú veľmi jednoducho jeden z druhého odvodiť). Z takého dôvodu musia byť tieto kľúče tajné, pretože znalosť jedného z nich umožňuje ako šifrovať, tak aj dešifrovať.

Dosť závažný problém, ktorý vplýva na celkovú bezpečnosť komunikácie, teda spočíva v distribúcii kľúča medzi zainteresované strany. Kľúč sa musí dostať ku všetkým, ktorí sa majú zapájať do komunikácie, ale zároveň sa nesmie dostať k nikomu inému.

Samotné symetrické šifry sa delia na dva základné typy: blokové (block ciphers) a prúdové (stream ciphers).

B.1.2.1 Blokové

Blokové správu rozdelia na bloky pevnej dĺžky (bežne 64 alebo 128 bitov), prípadne doplnia správu tak aby jej dĺžka bola násobkom dĺžky bloku. Potom sa každý blok zvlášť zašifruje.

Na šifrovanie správ dlhších ako je blok sa používa niekoľko rôznych postupov (mode of operation).

Pozn. C_i označuje i-ty blok zašifrovaného textu, P_i ozačuje i-ty blok čistého textu, O_i označuje i-ty pomocný blok, E() označuje šifrovanie, xor je súčet modulo 2 a IV je inicializačný vektor.

Electronic Codebook (ECB)

je najjednoduchší, každý blok sa zašifruje zvlášť rovnakým postupom ako ostatné bloky. Nevýhodou je, že tento postup zachováva štruktúru údajov: rovnaké bloky sú zašifrované rovnako. Taktiež tu nie je žiadna ochrana pred replay útokmi.

$C_i = E(P_i)$

Cipher-block Chaining (CBC)

každý blok (okrem prvého) sa pred zašifrovaním najprv transformuje operáciou XOR za použitia zašifrovaného predchádzajúceho bloku. Prvý blok sa transformuje XORom s IV.

Tento spôsob je najpoužívanejší. Jeho hlavnou nevýhodou je nemožnosť paralelného spracovania a takisto propagácia chýb (bitová chyba poškodí okrem dotknutého bloku aj dešifrovanie bloku nasledujúcom bezprostredne za ním).

$C_0 = IV$

$C_i = E(P_i \text{ xor } C_{i-1})$

Cipher Feedback (CFB)

tento mód približuje blokové šifry k prúdovým. Namiesto dát sa šifruje predchádzajúci zašifrovaný blok a výsledok sa XORuje s dátami. Na zašifrovanie prvého bloku sa používa IV. Propagácia chýb je nízka, jedna bitová chyba spôsobí 2 bitové chyby pri dešifrovaní (v zasiahnutom bloku a v nasledujúcom bloku na mieste pôvodnej bitovej chyby).

$C_0 = IV$

$C_i = E(C_{i-1}) \text{ xor } P_i$

Output Feedback (OFB)

podobne ako CFB sa približuje k prúdovým šifrám. Na rozdiel od CFB ale nepoužíva zašifrované dáta na šifrovanie nasledujúceho bloku. IV sa postupne šifruje a výsledok sa XORuje s dátami. Hlavná výhoda oproti iným spôsobom je prakticky žiadna propagácia chýb - bitová chyba v zašifrovanom texte spôsobí rovnakú bitovú chybu v dešifrovanom texte.

$O_0 = IV$

$O_i = E(O_{i-1})$

$C_i = O_i \text{ xor } P_i$

Tieto módy samy o sebe nezabezpečujú žiadnu integritu, tá musí byť riešená externe. Ak sa používajú nejaké prostriedky na zaručenie integrity, propagácia chýb u jednotlivých módov nemá význam (keďže pri neopravenej chybe je celá správa odmietnutá).

Príklady blokových šifier: 3-Way, AES, Akelarre, Anubis, Blowfish, Camellia, CAST-128, CAST-256, CMEA, CS-Cipher, DEAL, DES, DES-X, FEAL, FOX, FROG, G-DES, GOST, ICE, IDEA, Iraqi, KASUMI, KHAZAD, Khufu and Khafre, Libelle, LOKI89/91, LOKI97, Lucifer, MacGuffin, Madryga, MAGENTA, MARS, MISTY1, MMB, NewDES, Noekeon, RC2, RC5, RC6, REDOC, Red Pike, S-1, SAFER, SEED, Serpent, SHACAL, SHARK, Skipjack, SMS4, Square, TEA, Triple DES, Twofish, XTEA

B.1.2.2 Prúdové

Prúdové šifry šifrujú správu ako jednu postupnosť znakov. Základný princíp šifry je vygenerovať postupnosť pseudonáhodných dát na základe kľúča a ňou zašifrovať dáta (napr. operáciou XOR). Dešifruje sa rovnako, na základe kľúča sa vygeneruje reťazec a ním sa dáta dešifrujú (v prípade XOR je postup úplne rovnaký ako pri šifrovaní).

Ak by sa zašifrovaný text zmenil (napr. pridaním, alebo odobraním niekoľkých znakov), zvyšok textu by sa nepodarilo správne dešifrovať (tzv. strata synchronizácie). Existujú triedy šifier, ktoré obsahujú synchronizačné údaje schopné zachrániť časť správy (od najbližšieho checkpointu). Takéto šifry sa nazývajú samosynchronizujúce (self-synchronizing), alebo asynchrónne.

Prúdové šifry sú väčšinou rýchlejšie a menej náročné na hardware než blokové šifry, ale pri nesprávnom použití sú zraniteľné.

Substitučný útok nahradí známu časť textu za inú. Ak vieme, že v zachytenom šifrovanom texte nejaká časť zodpovedá napr. reťazcu "1000Sk", môžeme ho nahradiť vlastným, rovnako dlhým reťazcom, napr. "2000Sk". Stačí príslušnú časť zašifrovaného textu XORovať hodnotou "1000Sk" XOR "2000Sk". Z vlastností funkcie XOR vyplýva, že výsledok bude:

(kľúč xor "1000Sk")xor("1000Sk" xor "2000Sk") =
kľúč xor ("1000Sk" xor "1000Sk") xor "2000Sk" =
kľúč xor "2000Sk"

Čo po dešifrovaní správnym kľúčom dá "2000Sk". Zabrániť takýmto útokom môžeme zabezpečením integrity správy (napr. MAC).

Ďalší typ útoku sa spolieha na viacnásobné použitie rovnakého kľúča. Ak sú správy A a B zašifrované rovnakým kľúčom a útočník ich zachytí, je schopný získať A xor B. Pri znalosti jednej zo správ (resp. jej časti) je schopný zistiť druhú (resp. jej prislúchajúcu časť).

E(A) XOR E(B) =
(A XOR kľúč) XOR (B XOR kľúč) =
(A XOR kľúč) XOR (kľúč XOR B) =
A XOR (kľúč XOR kľúč) XOR B =
A XOR B.

Aj v prípade, že útočník nepozná ani A, ani B (ani ich časti), A XOR B sa dešifruje jednoduchšie, keďže aj A aj B pravdepodobne nemajú pseudonáhodnú štruktúru.

V niektorých prípadoch, keď sa šifruje spojitý tok dát, sú veľké úseky zašifrovaného textu vlastne šifrovací kľúč (x XOR 0 = x). Ak sa takáto komunikácia zachytí, jej nulové časti môžu v budúcnosti dešifrovať iné správy zašifrované rovnakým kľúčom.

Ochrana pred takýmito typmi útokov je buď používanie len jednorázových kľúčov (session-key), alebo použitie IV.

Príklady prúdových šifier: RC4, A5/1, A5/2, Chameleon, FISH, Helix, ISAAC, MUGI, Panama, Phelix, Pike, SEAL, SOBER, SOBER-128, WAKE.

B.1.3 Asymetrické šifry

Asymetrické šifry s verejným a súkromným kľúčom nepoužívajú, na rozdiel od symetrických šifier, rovnaký kľúč na šifrovanie aj dešifrovanie. Pri týchto šifrách existujú dva rôzne kľúče, verejný a súkromný. Verejný kľúč býva odvodený zo súkromného, ale platí, že z verejného kľúča nie je možné (v rozumnom čase) odvodiť súkromný kľúč.

Asymetrické šifry majú takú vlastnosť, že správa zašifrovaná použitím verejného kľúča sa dá rozšifrovať jedine pomocou súkromného kľúča.

Samotný princíp fungovania je nasledovný: účastník si vygeneruje (alebo nejak inak získa) súkromný kľúč, z neho odvodí verejný kľúč a tento verejný kľúč zverejní spolu so svojimi identifikačnými údajmi. Každý, kto mu chce poslať zašifrovanú správu si zistí jeho verejný kľúč, správu ním zašifruje a odošle ju.

Táto správa sa dá dešifrovať jedine použitím správneho súkromného kľúča, takže dešifrovať ju môže len ten, kto disponuje správnym súkromným kľúčom.

B.1.4 Zhrnutie

Výhodou symetrických šifier je ich rýchlosť a v prípade bezpečnej distribúcie kľúča aj spoľahlivosť. Asymetrické šifry sú spoľahlivé, nepotrebujú žiadnu distribúciu kľúča, ale ich nevýhodou je vyššia výpočtová zložitosť.

Bežne sa používa kombinácia symetrických a asymetrických šifier, ktorá odstraňuje nevýhody oboch prístupov. Asymetrická šifra sa použije na bezpečnú distribúciu kľúčov pre symetrickú šifru a ňou sa potom šifrujú dáta.

Bezpečnosť je rovnaká ako slabšia z použitých šifier. Takže ak sa použije dostatočne silná symetrická šifra, bezpečnosť bude rovnaká ako pri šifrovaní iba asymetrickou šifrou. Navyše výpočtová zložitosť by mala byť menšia.

Útočník môže prelomiť buď asymetrickú šifru, alebo symetrickú. Prelomiť asymetrickú šifru je v tomto prípade ešte náročnejšie ako ak by sa šifrovalo len asymetricky, pretože sa šifruje iba kľúč a teda útočník má k dispozícii menšie množstvo ciphertextu. Ak sa túto šifru podarí prelomiť, útočník získa symetrický kľúč a môže dešifrovať všetku komunikáciu.

Ak útočník prelomí symetrickú šifru (čo je ťažšie ako keby sa používala len symetrická šifra, keďže distribúcia kľúča by mala byť menej zraniteľná), dostane sa len k jednej správe (ostatné správy musí znovu prelomiť).

B.2 Elektronické podpisy

B.2.1 Úvod

Elektronické podpisy sú určené na zaručenie troch vlastností správ:

1) autentickosť

   správa pochádza naozaj od jej odosielateľa a nie od niekoho, kto sa za odosielateľa iba vydáva

2) integrita

   správa nebola počas prenosu zmenená, teda správa je po prijatí rovnaká, ako bola odoslaná

3) nepopierateľnosť

   odosielateľ nemôže v budúcnosti poprieť, že správu odoslal

Vo všeobecnosti sa k správe pripojí elektronický podpis (alebo sa prenesie k adresátovi iným kanálom) a adresát overí, či správa súhlasí s podpisom. Ak nesúhlasí, správu odmietne ako neplatnú (prenosové chyby, pokus o falšovanie).

B.2.2 Hašovanie

Hašovacie funkcie vytvárajú z dát ľubovoľnej veľkosti menší odtlačok. Tento sa dá použiť rôznymi spôsobmi v závislosti od použitej hašovacej funkcie. Jedno z využití je rozdeľovanie dát na niekoľko kategórií (napr. na ukladanie, vyhľadávanie a podobne), ďalej na detekciu (alebo opravovanie) chýb, identifikáciu rovnakých dát, či detekciu autentickosti alebo integrity.

Základná vlastnosť hašovacej funkcie je, že dve rôzne hašovacie hodnoty majú rôzne vzory ($h(A) \neq h(B) \Rightarrow A \neq B$). Na druhej strane ale rovnaká hašovacia hodnota nemusí nutne znamenať rovnaké vstupné dáta.

B.2.2.1 Kryptografické hašovacie funkcie

sú hašovacie funkcie s niektorými ďalšími vlastnosťami:

1) je ťažké nájsť vzor k hašu

    k h nájsť m také, že h=hash(m)

2) je ťažké nájsť k správe A správu B s rovnakým hašom.

    k A nájsť B také, že hash(A)=hash(B)

3) je ťažké nájsť dve rôzne správy s rovnakým hašom.

    nájsť A≠B, hash(A)=hash(B)

Kryptografické hašovacie funkcie sa používajú na určenie autentickosti a integrity správ.

Integrita sa určí porovnaním hašu s hašom správy, ak sa rovnajú, správa nebola zmenená. Tento prístup ale závisí na správnom doručení hašu správy (ak totiž útočník prepíše haš svojim vlastným, príjemca nezistí žiadne zmeny).

Autentickosť sa určuje tak, že na vygenerovanie hašu je potrebné poznať nejaké (zdieľané) tajomstvo, ktoré dokáže prijímateľ overiť. Ak teda haš spĺňa požiadavky, znamená to že ho vygeneroval niekto, kto pozná potrebné tajomstvo.

Príklady algoritmov: HAVAL, MD2, MD4, MD5, RIPEMD-128, RIPEMD-160, SHA-0, SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, Snefru, Tiger-128, Tiger-160, Tiger / Tiger2, WHIRPOOL

B.2.2.2 Message Authentication Code

Ako názov napovedá, používajú sa na autentifikáciu a integritu správ. Ich vstupom je kľúč a správa a vyprodukujú haš. MAC majú vlastnosť, že je ťažké nájsť dve správy A a B, ktoré majú rovnaký haš pre jeden (neznámy pre útočníka) kľúč. MAC sa líši od digitálnych podpisov tým, že na generovanie aj overovanie hašu sa používa rovnaký kľúč (takže neplatí vlastnosť nepopierateľnosti).

Príklady algoritmov: HMAC, CBC-MAC, OMAC/CMAC, PMAC, UMAC, Data Authentication Code, Poly1305-AES

B.2.3 Digitálne podpisy

sú založené na použití asymetrických šifier s verejným a súkromným kľúčom. Digitálny podpis sa vygeneruje zašifrovaním (hašu) správy odosielateľovým súkromným kľúčom a overuje sa dešifrovaním pomocou odosielateľovho verejného kľúča.

Keďže na správne vygenerovanie je potrebná znalosť súkromného kľúča, predpokladá sa že iba odosielateľ je schopný generovať správne digitálne podpisy. Ak sa teda podarí dešifrovať podpis odosielateľovým verejným kľúčom a podpis súhlasí so správou, vieme že bola odoslaná odosielateľom a cestou nebola zmenená (samozrejme za predpokladu, že máme autentickú kópiu odosielateľovho verejného kľúča).

Digitálne podpisy taktiež umožňujú nepopierateľnosť. Ak nebol skompromitovaný súkromný kľúč odosielateľa, jediný kto mohol vygenerovať platný digitálny podpis je on. Preto nemôže v budúcnosti poprieť, že tento podpis vygeneroval (do digitálnych podpisov sa pridávajú aj časové známky. V takomto prípade je možné odmietnuť pokus o popretie podpisu v prípade neskoršej kompromitácie kľuča).

Príklady algoritmov: Cramer-Shoup, DH, DSA, ECDH, ECDSA, EKE, ElGamal, GMR, MQV, NTRUEncrypt, NTRUSign, Paillier, Rabin, Rabin-Williams, RSA, Schnorr, SPEKE, SRP, XTR

B.2.4 Digitálne certifikáty

Digitálny certifikát sa používa na zviazanie verejného kľúča a identity. Pomocou digitálnych certifikátov je možné doručovať autentické verejné kľúče nezabezpečenou sieťou.

Typické použitie vyžaduje certifikačnú autoritu, ktorej všetky ostatné strany dôverujú (a majú autentickú kópiu jej verejného kľúča). Táto autorita potom zistí identitu strany žiadajúcej o certifikát (napr. osobným kontaktom) a túto identitu spolu s prislúchajúcim verejným kľúčom podpíše svojim súkromným kľúčom. Keď si iná strana vyžiada verejný kľúč, dostane kópiu tohoto certifikátu a môže overiť jeho autentickosť, keďže disponuje verejným kľúčom certifikačnej autority. Ak certifikát súhlasí, znamená to, že certifikačná autorita považuje daný verejný kľúč za autentický.

Vo väčších systémoch vznikajú hierarchie certifikačných autorít - jedna certifikačná autorita sa zaručuje za inú certifikačnú autoritu. Takto môžu overovať certifikáty aj strany ktoré dôverujú iným certifikačným autoritám.

Certifikáty väčšinou mávajú obmedzenú platnosť a periodicky musia byť obnovované - informácia o platnosti vtedy býva súčasťou certifikátu. Ak má certifikát byť neplatný skôr ako uplynie jeho doba platnosti, väčšinou sa pridáva do CRL. Overovanie certifikátu je vtedy podmienené prístupom do CRL, kde si entita overí, že certifikát nebol predčasne zrušený.