

УНИВЕРЗИТЕТ СИНГИДУМУ
БЕОГРАД
ДЕПАРТМАН ЗА ПОСЛЕДИПЛОМСКЕ СТУДИЈЕ
И МЕЂУНАРОДНУ САРАДЊУ

ДОКТОРСКА ДИСЕРТАЦИЈА

**ИНТЕГРАЦИЈА СОПСТВЕНИХ КРИПТОЛОШКИХ
СИСТЕМА У СТАНДАРДНУ РАЧУНАРСКО-
ТЕЛЕКОМУНИКАЦИОНУ ИНФРАСТРУКТУРУ**

МЕНТОР
проф. др Младен Веиновић

КАНДИДАТ
мр Александар Јевремовић

БЕОГРАД, 2011. година

Сажетак

Професионални системи - влада, војска, полиција, безбедносне агенције и сл. - захтевају највиши ниво заштите комуникација. Ти системи се не могу ослонити на јавно доступне системе заштите. Такође, не могу се ослонити ни на безбедносне системе који нису развијени у оквиру сопствене организације и чији су принципи рада, као и евентуални пропусти у дизајну и реализацији, познати ван организације. У циљу постизања највишег нивоа заштите комуникација професионални системи морају развити сопствене системе заштите, отпорне на све познате типове напада. Комплетан развој комуникационог система подразумева развој компонената на свим нивоима - од корисничког интерфејса до комуникационе инфраструктуре. Такав развој захтева огромне ресурсе и улагања, као и време за развој, тестирање и уклањање грешака. У овом раду се разматрају могућности за искоришћавање постојећих јавних комуникационих инфраструктура и технологија за развој и употребу сопственог система који нуди највиши ниво заштите комуникација. У оквиру рада представљен је детаљан процес развоја и карактеристике решења заснованог на интеграцији сопствених шифарских алгоритама на мрежном комуникационом слоју.

Abstract

Professional systems used by government, military, police, security agencies, etc require highest possible level of communication protection. Those systems cannot rely on publicly available protection systems. Also, they cannot rely on security systems which has not been developed within organization and whose principles of operation or possible flaws in design and implementation are known outside the organization. In the course of achieving highest level of communication protection, professional systems have to develop protection systems of their own, which will be resistant to all known types of attacks. Complete development of communication system implies component development on every level - from user interface to communication infrastructure. That kind of development requires vast resources and investments, as well as proper time for development, testing and debugging. In this paper are discussed utilization possibilities of available public communication infrastructures and technologies for development and usage of system which offers highest possible level of communication protection. Within the paper is presented detailed development process and characteristics of solution based on integrating own ciphers on network communication layer.

Индекс слика

1. Број корисника Интернета (у милионима) по регионима у 2010. години	17
2. <i>OSI</i> и <i>TCP/IP</i> модели	19
3. Принцип рада <i>A5/1</i> шифарског алгоритма	21
4. Алгоритам метода „сирове силе“	24
5. Изглед, структура и коришћење <i>COPACOBANA</i> уређаја	28
6. Структура напада са посредовањем у комуникацији	29
7. Модел рефлектованог напада са циљем онемогућавања услуге	30
8. Модел дистрибуираног напада са циљем онемогућавања услуге	30
9. Заштита на нивоу апликативног протокола	33
10. Заштита на нивоу транспортног протокола	38
11. Заштита на нивоу мрежног протокола	54
12. Архитектура комуникације заштићене <i>IPsec</i> системом заштите	55
13. Различити модели заштите коришћењем <i>IPsec</i> система	56
14. Однос <i>RFC</i> докумената значајних за <i>IPsec</i>	57
15. Процес слања података са коришћењем <i>IPsec</i> система заштите	58
16. Транспортни режим рада	59
17. Тунелски режим рада	59
18. Структура <i>IP</i> датаграма са <i>AH</i> протоколом у транспортном режиму рада	60
19. Структура <i>IP</i> датаграма са <i>AH</i> протоколом у тунелском режиму рада	61
20. Структура <i>IP</i> датаграма са <i>ESP</i> протоколом у транспортном режиму рада	62
21. Структура <i>IP</i> датаграма са <i>ESP</i> протоколом у тунелском режиму рада	63
22. Организација компонената значајних за <i>IPsec</i>	66
23. Итерације у процесу шифровања код <i>AES</i> алгоритма	73

24. Алгоритам за шифровање код <i>DES</i> алгоритма	74
25. Алгоритам израчунавања кључева	74
26. Проширивање скупа шифарских алгоритама језгра сопственим алгоритмом	84
27. Избор сопственог алгоритма у конзолном конфигурационом програму	86
28. Опис сопственог алгоритма у конзолном конфигурационом програму	87
29. Избор сопственог алгоритма у графичком конфигурационом програму	87
30. Основна шема реализације предложеног система заштите	90
31. Оригинална <i>IPsec/IKEv2</i> архитектура	90
32. Измењена <i>IPsec/IKEv2</i> архитектура	91
33. Модел система се претходно размењеном базом кључева	92
34. Коришћење базе кључева са додатком главног кључа	93
35. Модел са коришћењем базе кључева као сесијских кључева	94
36. Шема рачунарске мреже коришћене у експерименту	96
37. Дијаграм са оствареним брзинама протока по алгоритму	99
38. Перформансе <i>Intel->AMD</i> , без шифровања	99
39. Перформансе <i>AMD->Intel</i> , без шифровања	100
40. Перформансе <i>Intel->AMD</i> , <i>AES</i> алгоритам	101
41. Перформансе <i>AMD->Intel</i> , <i>AES</i> алгоритам	101
42. Перформансе <i>Intel->AMD</i> , <i>3DES</i> алгоритам	102
43. Перформансе <i>AMD->Intel</i> , <i>3DES</i> алгоритам	103
44. Перформансе <i>Intel->AMD</i> , <i>FBISCA</i> алгоритам	103
45. Перформансе <i>AMD->Intel</i> , <i>FBISCA</i> алгоритам	104
46. Архитектура <i>Android</i> платформе	110

Индекс табела

1. <i>NSA</i> препоруке дужина кључева	26
2. <i>ECRYPT</i> препоруке дужина кључева	26
3. <i>BSI</i> препоруке дужина кључева	27
4. Алгоритми за компресовање података у <i>OpenPGP</i> систему	35
5. Алгоритми за израчунавање хеш вредности у <i>OpenPGP</i> систему	35
6. Симетрични шифарски алгоритми у <i>OpenPGP</i> систему	36
7. Асиметрични шифарски алгоритми у <i>OpenPGP</i> систему	36
8. Подршка за системе заштите у имплементацијама	51
9. Подршка за алгоритме за размену кључева	52
10. Подршка за шифарске алгоритме	52
11. Подршка за алгоритме за компресовање података	52
12. Подршка за екстензије	52
13. Карактеристике рачунара коришћених у експерименту	96
14. Резултати мерења.	105

Индекс листинга програмских кодова

1. Декларације структуре <i>TLSPlaintext</i> и референтних структура	43
2. Декларације структуре <i>TLSCompressed</i>	44
3. Декларације структуре <i>TLS_CIPHERTEXT</i>	44
4. Декларације структуре <i>GenericBlockCipher</i>	45
5. Псеудо-код рутине <i>Key Expansion</i>	71
6. Структура за повезивање шифарског алгорита са језгром	85
7. Садржај <i>Kconfig</i> који се односи на сопствени алгоритам	88
8. Код за допуну <i>Makefile</i> фајла за укључивање сопственог алгорита	88

Садржај

1. Увод.....	10
1.1. Контекст истраживања.....	10
1.2. Мотивација.....	11
1.3. Циљеви истраживања.....	12
1.4. Специфични појмови.....	13
1.5. Структура тезе.....	13
2. Безбедност у савременим телекомуникацијама.....	15
2.1. Савремена комуникациона инфраструктура.....	15
2.1.1. Јавна телефонска мрежа.....	16
2.1.2. Интернет.....	17
2.1.2.1. OSI и TCP/IP модели комуникације.....	18
2.1.2.2. Интернет телефонија.....	20
2.1.3. Мобилна телефонија.....	20
2.1.3.1. Четврта генерација мобилне телефоније.....	22
2.2. Неки типови и модели напада.....	24
2.2.1. Криптоанализа.....	24
2.2.2. Разбијање шифрата применом сирове силе.....	24
2.2.3. Напад посредовањем у комуникацији.....	29
2.2.4. Напад ускраћивањем услуге.....	30
2.3. Најчешће коришћени системи за заштиту комуникације.....	33
2.3.1. Заштита на нивоу апликативног протокола.....	33
2.3.1.1. Заштита електронске поште.....	34
2.3.1.2. Протокол SSH.....	38
2.3.2. Заштита на нивоу транспортног протокола.....	40
2.3.2.1. Transport Layer Security протокол.....	41
2.3.2.1.1. Протокол TLS Record.....	43
2.3.2.1.2. Протоколи TLS-а за успостављање везе.....	48
2.3.2.1.3. Управљање кључевима.....	50
2.3.2.1.4. Криптографска израчунавања.....	51
2.3.2.1.5. Имплементације TLS протокола.....	52
2.3.2.1.6. Интеграција сопственог шифарског алгоритма.....	54

2.3.3. Заштита на нивоу мрежног протокола.....	56
2.3.3.1. Internet Protocol Security систем заштите.....	57
2.3.3.1.1. Режи́ми рада.....	61
2.3.3.1.2. Протокол Authentication Header.....	62
2.3.3.1.3. Протокол Encapsulating Security Payload.....	64
2.3.3.1.4. Имплементације IPsec протокола.....	66
2.3.3.1.4.1. Пројекат KAME.....	67
2.3.3.1.4.2. Имплементација у језгру Линукс оперативног система.....	68
2.3.3.1.5. Internet Key Exchange протокол.....	69
2.4. Стандардни шифарски алгоритми.....	71
2.4.1. Advanced Encryption Standard, AES.....	72
2.4.2. Data Encryption Standard (DES, 3DES).....	76
2.4.3. RSA.....	77
2.5. Трендови.....	78
2.6. Стандарди, референтна тела и организације.....	80
2.6.1. Међународне организације.....	80
2.6.2. Националне организације.....	81
3. Развој сопственог система заштите на мрежном слоју.....	82
3.1. Избор комуникационе инфраструктуре.....	82
3.2. Избор носећих технологија.....	82
3.2.1. Избор комуникационих протокола.....	83
3.2.2. Избор оперативног система.....	83
3.2.3. Избор хардверске платформе.....	84
3.3. Успостављање чистог развојног и оперативног рачунарског система.....	85
3.4. Интеграција сопственог шифарског алгоритма у IPsec систем заштите.....	86
3.4.1. Модел и процес интегрисања сопственог шифарског алгоритма.....	86
3.4.1.1. Повезивање алгоритма са језгром оперативног система.....	87
3.4.1.2. Регистрација шифарског алгоритма.....	88
3.4.1.3. Конфигурисање и превођење измењеног оперативног система.....	89
3.5. Модели управљања кључевима са криптоанализом.....	92
4. Експериментална анализа перформанси решења.....	98
4.1. Циљеви експерименталне анализе.....	98

4.2. Модел експерименталног окружења.....	99
4.3. Структура експеримента.....	101
4.4. Резултати.....	101
4.5. Дискусија резултата.....	109
5. Области примене предложеног решења.....	111
5.1. Примена решења на WAN и Интернет мрежама.....	111
5.2. Примена решења у локалној мрежи.....	111
5.3. Примена решења у мобилној телефонији.....	112
5.3.1. Android платформа.....	113
5.3.1.1. Основна безбедносна архитектура.....	114
5.3.1.2. Откривени безбедносни пропуси.....	116
5.3.1.3. Интеграција сопственог шифарског алгоритма.....	118
6. Закључак.....	119
6.1. Сумирање циљева истраживања.....	119
6.2. Остварени резултати и допринос.....	119
6.3. Даљи правци рада.....	120
6.3.1. Развој интегрисаног хардверско-софтверског решења.....	120
6.3.2. Хардверска реализација шифарског алгоритма.....	121
6.3.3. Анализа погодних медија за пренос и складиштење кључева.....	122
7. Литература.....	123
7.1. RFC документи.....	125
7.2. Објављени радови у научним часописима и конференцијама.....	126

1. Увод

Професионални системи - влада, војска, полиција, безбедносне службе - захтевају поуздане системе који нуде висок ниво заштите садржаја комуникација од прислушкивања и измена. Комплетан развој таквих система захтева огромно време и ресурсе, а њихов рад захтева посебну телекомуникациону инфраструктуру. У овом раду се разматрају могућности развоја професионалних система заштите коришћењем готових, отворених компонената, као и њихово коришћење на стандардним телекомуникационим инфраструктурама.

1.1. Контекст истраживања

Развој телекомуникационих технологија и инфраструктура довели су до огромног повећања могућности преноса података на даљину у погледу брзине, формата и географске и економске доступности. Данас, просечни грађани и компаније често користе и по неколико комуникационих инфраструктура - јавну телефонску мрежу, мобилну телефонију, Интернет мрежу, и т.д.. Развој технологија омогућава све једноставније коришћење телекомуникационих сервиса, а вредност информација које се њима преносе расте.

Једно од поља са најбржим развојем телекомуникационих технологија и сервиса јесте Интернет мрежа. Иако припадају релативно младој комуникационој инфраструктури, њене технологије су данас у стању да подрже готово све телекомуникационе сервисе присутне на осталим популарним инфраструктурама - јавној телефонској мрежи, мобилној телефонији, кабловској телевизији, и т.д. - као и велики број сервиса који на осталим инфраструктурама није могућ. Из тог разлога све већи број сервиса мигрира на Интернет мрежу и може се очекивати да ће она у будућности представљати универзалну глобалну телекомуникациону инфраструктуру.

Сложеност телекомуникационих сервиса и система такође је у порасту. Међутим, стандардизација технологија Интернет мреже омогућава развој нових решења кроз коришћење већ развијених комуникационих протокола и хардверских и софтверских компонената. Неки од ових протокола и компонената су затвореног типа, док је у неким случајевима комплетна спецификација и имплементација комуникационих система јавно доступна и слободна за сопствено прилагођавање и коришћење.

Заштита телекомуникација већ више деценија представља актуелну тему. Постоји велики број развијених решења и стандарда. Међутим, професионални системи, у које спадају влада, војска, полиција и безбедносне службе, не смеју се ослонити на решења развијена од стране спољних или страних развојних центара. Један од основних критеријума њихових безбедносних система јесте интерни развој и потпуна контрола над свим параметрима безбедности коју систем нуди. Развој и коришћење таквих система може представљати изузетно сложен, дуготрајан и скуп подухват. У овом раду се анализира могућност развоја професионалног система за заштиту комуникација који би се квалитетније и брже могао изградити кроз коришћење отворених готових компонената, а који би се могао јефтиније користити кроз интеграцију у стандардну рачунарско-телекомуникациону инфраструктуру.

1.2. Мотивација

На почетку овог истраживања утисак је био да постојећи системи за заштиту комуникација садрже извесну недореченост у погледу тежње ка остваривању апсолутне безбедности. Познавање историјских контекста у којима су данас најпопуларнији системи заштите развијани, као и ауторитета који су на те развоје имали утицаја, поставило је питање, тачније сумњу, да се на те системе не могу ослонити корисници који рукују информацијама највиших нивоа осетљивости. Због тога су, као једног од фактора, у раду, код анализе одређених популарних система заштите, наведене и неке не техничке чињенице.

Ближа (али и даља) историјска дешавања у нашем окружењу, као и на општој геополитичкој сцени, створила су потребу за дубљим и детаљнијим разматрањима појмова суверенитета и слободе. Они се, у неким другим, више теоријским условима, релативно лако могу дефинисати, а постојање једне државе и народа апсолутно условити њиховим остварењем. Наши конкретни услови, напротив, захтевају „слојевит“ приступ чији је један од корака анализа могућности за компромисно, а опет задовољавајуће практично деловање. У том погледу ова теза представља један сегмент овог широког и изузетно важног питања, а њен циљ је да на пољу рачунарских телекомуникација понуди практична решења у складу са наведеним критеријумима.

Специфичност области примене резултата ове тезе умањује могућност општег

техничког доприноса, у погледу могућности широке практичне примене развијеног решења. Међутим, један од мотива за израду ове тезе јесте и давање прегледа актуелних технологија које се за коначан циљ могу употребити, као и документовање процеса реализације решења еквивалентног предложеном.

1.3. Циљеви истраживања

Основни циљ овог истраживања јасно је изражен у самом наслову тезе, а он представља развој сопственог система заштите који се може користити на јавној рачунарско-телекомуникационој инфраструктури. Карактеристика „сопствени“ у опису циљног система заштите не односи на власништво у погледу лиценци и права коришћења, већ првенствено на могућност самосталне контроле нивоа заштите који тај систем нуди. Раније поменуто неповерење у актуелне системе заштите основни је мотив за постизање овог циља.

Дефинисање карактеристика система који нуди апсолутну заштиту, или контролу њеног нивоа, решено је још у првој половини прошлога века од стране америчког инжењера Клода Шенона у оквиру његовог главног доприноса савременој науци - теорије информација. Међутим, једно од данас актуелних питања на овом пољу јесте могућност реализације поменутог система у постојећим условима, са постојећим технологијама и на постојећој комуникационој инфраструктури. Из тог разлога се, а као природни предуслов остваривања основног постављеног циља, јавља и циљ у виду анализе тренутног стања (по свим наведеним критеријумима - услова, технологија и инфраструктуре), односно давања ажурног и што свеобухватнијег прегледа могућности: избора и алтернатива.

Једна од специфичности безбедносних система огледа се у томе да све њихове компоненте морају задовољити исте стандарде. Уколико се у систему јаве компоненте које учествују у остваривању безбедности, или повезују систем заштите са инфраструктуром, а које нуде низак или непознат ниво заштите, тај ниво заштите се везује за комплетан систем. Из тог разлога, као посредан циљ истраживања постављено је и проналажење адекватних компонената које не угрожавају ниво безбедности система и чији се ниво безбедности може проверити кроз проверу на основу њихове отворене архитектуре.

1.4. Специфични појмови

У самом наслову рада, као и велики број пута у даљем тексту, користи се појам „сопствени криптолошки систем“. Овај појам се не односи на лично власништво, нити на ауторска права над идејом или реализацијом криптолошког система већ се њиме одређују следеће карактеристике предложеног криптолошког система:

- у систему се користе сопствени алгоритми за све операције од којих зависи ниво заштите,
- сви сопствени алгоритми су самостално кодовани на „чистим“, безбедним рачунарима, а за додатне функције коришћене су отворене програмске библиотеке чији је код проверен,
- сви сопствени алгоритми самостално су реализовани (преведени и интегрисани у коначан систем) у оквиру „чистих“ рачунарских система (о чему ће бити више речи у делу рада који се бави детаљима развоја предложеног система),

Појам који се јавља у дефиницији „сопственог криптолошког система“ јесте и „чист“ рачунарски систем. Атрибут чист одређује да се у поменутом рачунарском систему не налази ни једна компонента чији детаљи функционисања нису у потпуности познати, односно да не постоји ни једна функција у систему коју аутори нису развили или одобрили. Детаљи успостављања чистог рачунарског система дати су у делу рада који се бави његовим успостављањем за потребе развоја система.

1.5. Структура тезе

У наставку тезе, у њеном другом делу, разматрано је опште стање у савременим телекомуникацијама. Први корак овог разматрања односи се на савремене комуникационе инфраструктуре: јавну телефонску мрежу, Интернет мрежу и мобилну телефонију. Томе следи анализа познатих типова и модела напада, са једне стране, и најчешће коришћених система за заштиту комуникација са друге. Битну компоненту ових система (која се уједно и критикује и представља основ једне од главних мотивација за израду ове тезе) чине стандардни шифарски алгоритми и они су разматрани у наставку. Остатак другог дела рада односи се на актуелне трендове на пољу безбедности и списак тела и организација референтних за поље безбедности.

У трећем делу рада документован је процес развоја сопственог система заштите. Њега чини избор циљне телекомуникационе инфраструктуре на којој ће систем радити, избор носећих технологија (комуникационих протокола, оперативног система и хардверске платформе) и процес интегрисања сопственог шифарског алгоритма у *IPsec* систем заштите. Овај део рада посебно може бити од користи за практичну реализацију система сличног предложеном јер садржи комплетан процес повезивања сопственог шифарског алгоритма са језгром Линукс оперативног система.

Перформансе развијеног система анализирани су експериментално а изложене су у четвртом делу рада. У њему су изложени циљеви експерименталне анализе, модел експерименталног окружења, структура вршеног експеримента, резултати анализе и њихова критичка анализа. Значај овог поглавља огледа се у представљеној разлици у перформансама између сопственог, сложенијег алгоритма и стандардних шифарских алгоритама, као и у анализи употребљивости остварених перформанси за практичну примену.

У наредном делу рада разматрају се могуће области у којима се решење може применити. Као основне области примене идентификоване су мреже са дељеним комуникационим каналима, првенствено Интернет, и мобилна телефонија четврте генерације.

У закључном делу рада сумирани су циљеви истраживања, као и остварени резултати и доприноси. Додатно, анализирани су недостаци постигнутог решења у виду даљих праваца истраживања и развоја.

2. Безбедност у савременим телекомуникацијама

Савремене телекомуникације у већини случајева користе дељену комуникациону инфраструктуру. Дељена инфраструктура омогућава смањивање трошкова комуникације јер се трошкови за њено постављање и одржавање расподељују на већи број корисника. Код дељене комуникационе инфраструктуре најчешће постоје наменски краћи комуникациони канали од крајњег корисника до најближег чворишта. Након тога, комуникациони сигнали корисника се између чворишта, а до одговарајућих осталих комуникационих страна, заједно преносе коришћењем неке од метода мултиплексовања (фреквенцијска, временска, кодована и сл.).

Дељени комуникациони канали, са друге стране, отварају могућност за безбедносне проблеме, првенствено кроз могућност прислушкивања и измене садржаја комуникације. Још увек није редак случај да организација, која има потребе за удаљеним преносом осетљивих информација, изнајмљује комуникационе линије између својих организационих јединица и на тај начин обезбеђује безбеднију комуникацију по неколико десетина, па и стотина пута већој цени од оне по којој се могу користити дељене комуникационе линије.

У овом делу рада, у циљу анализе актуелних претњи по безбедност, као и актуелних система за заштиту, биће анализирани савремене комуникационе инфраструктуре, познати типови и модели напада, најчешће коришћени системи за заштиту комуникације, најчешће коришћени шифарски алгоритми, трендови на пољу безбедности у телекомуникацијама и референтна тела и организације. Коначан циљ ове анализе јесте да дефинише могуће инфраструктуре у које би се интегрисао сопствени систем за заштиту комуникације и неопходни ниво заштите који тај систем мора да пружи, као критеријуме који ће бити коришћени у даљем раду на развоју предложеног решења.

2.1. Савремена комуникациона инфраструктура

Данас се за размену података користи више комуникационих инфраструктура. Оне могу бити локалне и глобалне, стандардизоване или не, жичне или бежичне и т.д. За потребе овог рада фокус је постављен на глобалне комуникационе инфраструктуре базиране првенствено на отвореним технологијама. У оквиру анализе типова савремене

комуникационе инфраструктуре биће обрађене јавна телефонска мрежа, Интернет мрежа и мобилна телефонија.

2.1.1. Јавна телефонска мрежа

Јавна телефонска мрежа једна је од још увек најмасовније коришћених инфраструктура за комуникацију. Ова инфраструктура је реализована углавном жичним комуникационим каналима, увезаним преко локалних, регионалних, националних и интернационалних централа. Такође, пословни системи могу имати интерну телефонску мрежу која је повезана са јавном телефонском мрежом. Основни уређај за коришћење јавне телефонске мреже јесте телефон, уређај за основну двосмерну аудио-комуникацију, али се она може користити и код (разних) других уређаја - факс уређај, модем, и сл.

Јавна телефонска мрежа је, поред своје основне намене, у великом проценту служила и за повезивање корисника са Интернет мрежом, односно, за њихово повезивање са Интернет провајдерима путем модема. Ова мрежа се, због своје распрострањености, и данас користи за приступ корисника Интернету, првенствено путем *xDSL* технологија.

За стандардизацију јавне телефонске мреже задужена је организација под називом Међународна Телекомуникациона Унија (енгл. *International Telecommunication Union, ITU*). Ова организација има седиште у Женеви и представља агенцију Уједињених нација а чини је 191 земља чланица. Циљ организације је развој и имплементација јавне телефонске мреже и пратећих технологија. Последња препорука *E.164* ове организације, објављена 1997. године под називом „Јавни међународни план за нумерисање телекомуникација“, дефинисала је следећа правила за одређивање телефонских бројева:

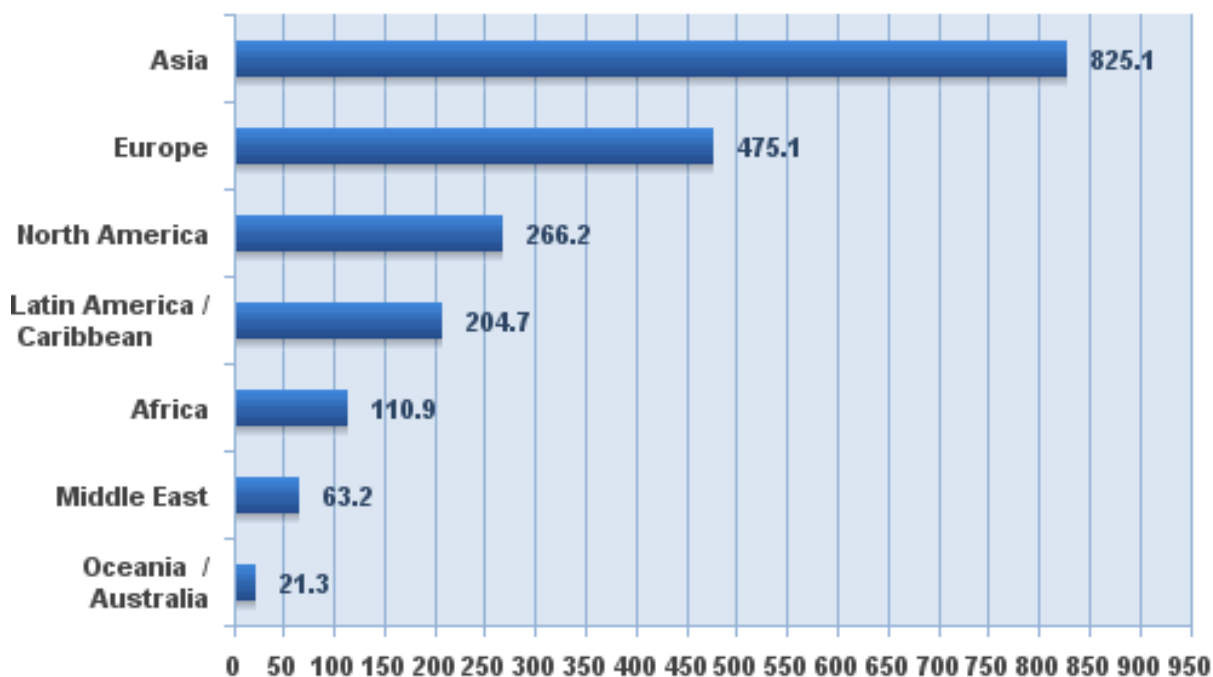
1. телефонски број може имати највише 15 цифара;
2. прве цифре (једна до три) телефонског броја одређују земљу којој он припада;
3. средишње цифре телефонског броја чине национални одредишни код (енгл. *national destination code, NDC*);
4. остале цифре телефонског броја представљају број претплатника (енгл. *subscriber number, SN*);
5. национални одредишни код и број претплатника представљају део телефонског

броја са националним значајем.

Интеграција сопственог система заштите комуникације у јавну телефонску мрежу првенствено се може извести на нивоу комуникационих уређаја, првенствено телефонских апарата. Међутим, тренд преласка са класичне на Интернет телефонију указује на то да је оптималнији избор вршити интеграцију сопственог система заштите кроз технологије које се у њој користе. Када је у питању модемско коришћење јавне телефонске мреже, оно подлеже истим правилима и структури комуникације као и када су у питању рачуарске комуникације путем Интернет мреже. Из наведених разлога сматрамо оптималнијим решењем фокусирање интеграције система заштите кроз Интернет технологије.

2.1.2. Интернет

Интернет мрежа је највећа глобална комуникациона мрежа, са највећим бројем корисника и највећим могућностима, и као таква представља циљну мрежу у којој би предложени систем могао да се користи за заштиту комуникација. Процењује се (слика 1) да је ову мрежу у току 2010. године користило готово две милијарде људи широм света.



Слика 1. Број корисника Интернета (у милионима) по регионима у 2010. години

(Извор: <http://www.internetworldstats.com>)

Једна од, за овај рад, изузетно значајних карактеристика Интернет мреже јесте њена заснованост на отвореним протоколима и технологијама. Иако у пракси постоји могућност и одређен број реализација затворених протокола и технологија, битно је да су спецификације свих кључних технологија и протокола дате у одговарајућим документима које су креирале референтне организације. Та чињеница одговара потребама код реализације решења представљеног у овом раду - коришћење отворених технологија и имплементација за развој и коришћење сопственог затвореног система заштићене комуникације.

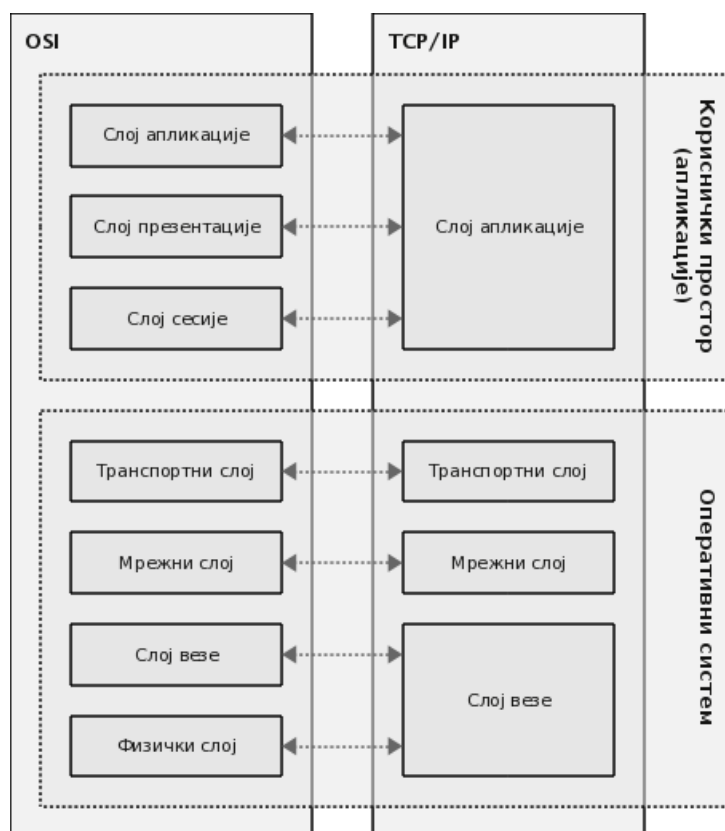
2.1.2.1. OSI и TCP/IP модели комуникације

У раним фазама развоја рачунарских мрежа већину рачунарских система су чинили *UNIX mainframe* рачунари са прикљученим корисничким терминалима. Иако су кориснички терминали били повезани комуникационим каналима са *mainframe*-ом таква мрежа се не може у сматрати пуном смислу речи рачунарском пре свега због недостатка рачунске моћи терминала - веза између терминала и *mainframe*-а имала је за задатак пренос корисничких инструкција до *mainframe*-а и резултата обраде до терминала. У таквој ситуацији је ексклузивно право на развој хардвера, софтвера и комуникационих канала углавном имао само један произвођач који је своја решења држао затвореним за остале произвођаче. Комуникација између решења различитих произвођача је најчешће била немогућа услед некомпатибилности између хардверских интерфејсова и формата података.

Пад цене рачунске моћи, преко све јефтинијих технологија за развој микропроцесора и рачунарске опреме, довео је до појављивања већег броја произвођача рачунарских система. Такав развој омогућио је децентрализацију рачунарске моћи која је указала на потребу за комплекснијом комуникацијом између радних станица. Такође, постојање већег броја произвођача ублажило је различитост и некомпатибилност њихових решења а убрзо се увидела и потреба за универзалним комуникационим стандардима.

На ову тему се крајем седамдесетих година двадесетог века огласила и Интернационална Организација за Стандардизацију (енгл. *International Organization for Standardization, ISO*) развијањем модела за комуникацију између разнородних система. Модел је објављен 1984. године и назван је *Open System Interconnection Basic Reference Model* или, скраћено, *OSI* модел. Овај модел је понудио фазно превођење формата

података кроз седам слојева па се стога назива и *OSI* седмослојни модел. Прихватањем овог *ISO* стандарда произвођачи су били у могућности да остваре потпуну комуникацију са системима без увида у њихову интерну спецификацију и формат података.



Слика 2. *OSI* и *TCP/IP* модели

Једна од главних мана *OSI* модела јесте непотребно залажење у интерне делове рачунарских система тј. дефинисање компоненти које нису директно задужене за међу-системску комуникацију. Као последица тога јавио се већи број корисничких апликација које нису у потпуности поштовале *OSI* стандард а ипак су биле у могућности да несметано комуницирају коришћењем нижих слојева модела. Оваква тенденција се резутовала појављивањем једноставнијег Интернет модела (*TCP/IP*) који даје већу слободу при избору архитектуре апликативног софтвера. Такође, овај модел апстрактно гледа и на најниже слојеве *OSI* модела с обзиром на то да се компоненте тих слојева најспорије развијају и то углавном од стране великих организација.

2.1.2.2. Интернет телефонија

Интернет телефонија представља један од сервиса који се могу користити путем Интернет мреже и мрежа заснованих на Интернет технологијама. Овај сервис се често среће и под називом „глас преко Интернет протокола“ (енгл. *Voice over IP, VoIP*). Основни задатак Интернет телефоније је да омогући миграцију популарних сервиса са јавне телефонске мреже (пренос гласа и факсимила) на Интернет технологије.

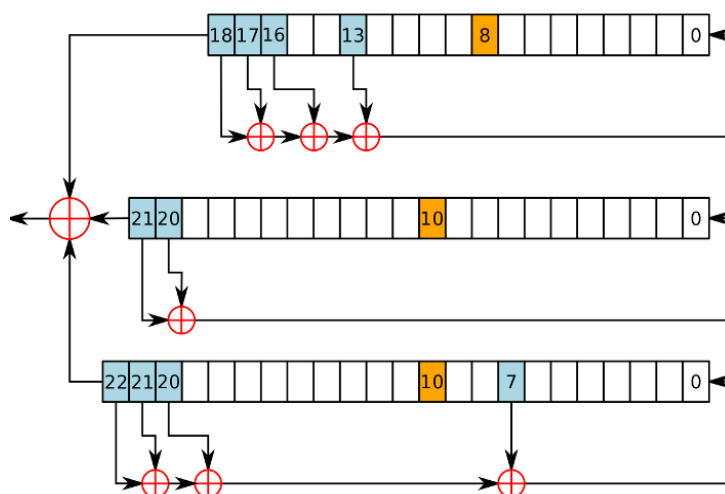
Интернет телефонија се реализује коришћењем више различитих затворених и отворених протокола. Историјски најзначајнији сет протокола за реализацију Интернет телефоније је *H.323*, као *ITU-T* препорука из 1996. године. Овај сет протокола омогућава пренос гласа и видео сигнала а подржан је у већини система за Интернет телефонију и видео-конференције. Поред *H.323* протокола у пракси се све чешће среће и *Session Initiation Protocol (SIP)* као протокол за успостављање и контролу сесија Интернет телефоније.

Једна од за овај рад најзначајнијих карактеристика Интернет телефоније је коришћење *IP* протокола од стране протокола који се користе за њену реализацију. То значи да се реализацијом система заштите на нивоу *IP* протокола могу штитити и подаци комуникационих сесија Интернет телефоније. У делу рада под називом „Области примене предложеног решења“ детаљније ће се разматрати допринос предложеног решења заштити аудио и видео података који се преносе у сервисима Интернет телефоније.

2.1.3. Мобилна телефонија

Основни безбедносни механизам мобилне телефоније чини провера аутентичности претплатника коришћењем тајног кључа познатог обема странама (претплатнику и оператеру мобилне телефоније) и *challenge response* система провере. У трећој генерацији мобилне телефоније за додатну проверу идентитета могу се користити додатни системи као што су *Universal Subscriber Identity Module*. Овај систем омогућава коришћење кључева веће битске дужине, као и проверу аутентичности мреже оператера мобилне телефоније. У ранијим *GSM* реализацијама не постоји провера аутентичности мреже што оставља могућност обмане корисника.

Садржај комуникација остварених путем мобилне телефоније сматра се заштићеним у смислу да се не преноси у изворном облику већ га је могуће шифровати неким од одговарајућих алгоритама. У најпопуларније шифарске алгоритме везане за мобилну телефонију спадају *A5/1*, *A5/2* и *A5/3*.



Слика 3. Принцип рада *A5/1* шифарског алгоритма

Алгоритам *A5/1* је секвенцијални шифарски алгоритам развијен 1981. године, а који се користи за заштиту садржаја мобилних комуникација у САД и европским земљама. Принцип рада овог алгоритма у почетку није био доступан али је од 1994. године познат јавности путем „цурења“ информација, а 1999. године је урађен комплетан реверзни инжењеринг. Године 2000. Алекс Бирјуков, Ади Шамир и Дејвид Вагнер представили су [1] могућност крипто-анализе у реалном времену, уз разбијање кључа у периоду од једне секунде до два минута. Овај резултат је заснован на резултатима професора Јована Голића из 1997. године, представљеним на конференцији *EUROCRYPT* [2]. Исте године су Ели Бихем и Ор Данкелман објавили [3] другачији метод напада, ефикаснији у погледу сложености рада ($2^{39,91}$ наспрам 2^{48} корака) и количине генерисаног садржаја.

Без обзира на наведене пропусте и низак ниво заштите који пружа овај алгоритам, он је само у 2000. години искоришћен за заштиту комуникација око 130 милиона корисника мобилне телефоније. Патрик Екдал и Томас Јохансонс су 2003. године објавили још један тип напада на *A5/1* алгоритам [4]. Године 2009. немачки стручњак за рачунарску безбедност, Карстен Нол, објавио је нову могућност разбијања заштите коју нуди овај алгоритам уз изјаву да је разбијање могуће извршити путем јавно доступних софтверских компонената са доступним изворним кодом [5]. У том тренутку је око

осамдесет процената свих мобилних комуникација било заштићено овим алгоритмом.

Алгоритам *A5/2* користи се у регионима ван САД и европских земаља. Шифрате које генерише овај алгоритам могуће је разбити коришћењем једноставних напада који подразумевају само поседовање шифрата, без додатних параметара. Истог месеца 1999. године, када је алгоритам и објављен, Јан Голдберг и Дејвид Вагнер извршили су крипто-анализу овог алгоритма и показали да се он може разбити у реалном времену коришћењем хардверско/софтверских компонената јавно доступних крајњим корисницима. Овај алгоритам је од 2006. године престала да подржава *GSM* асоцијација и уместо њега предложила коришћење *A5/1* алгоритма.

2.1.3.1. Четврта генерација мобилне телефоније

Четврта генерација мобилне телефоније представља следећи еволутивни корак на пољу мобилних комуникација. Предвиђа се да ће ова генерација телекомуникационих услуга превладати у периоду од 2012. до 2015. године. Неке од главних карактеристика ове генерације мобилне телефоније јесу комплетна архитектурна промена и далеко веће брзине преноса података.

Радна група задужена за развој стандарда четврте генерације мобилне телефоније дефинисала је следеће циљеве које та генерација треба да оствари:

1. номинална брзина преноса од 100Mb/s код покретних корисника и 1Gb/s код релативно фиксираних корисника [6];
2. минимална брзина преноса од 100Mb/s између најудаљенијих корисника у глобалној мрежи [6];
3. спектрално ефикасан систем [7];
4. висок мрежни капацитет – више истовремених корисника по приступној ћелији [8];
5. течан прелазак између различитих хетерогених мрежа [9];
6. општа покривеност и глобални роинг [10];
7. висок квалитет мултимедијалних сервиса (пренос звука у реалном времену, висока брзина преноса података, *HDTV* видео садржај и сл.) [10];

8. интероперабилност са постојећим стандардима за бежично умрежавање рачунара [11];
9. базираност на пакетском преносу путем Интернет протокола [10].

Једна од најважнијих карактеристика четврте генерације мобилне телефоније, са аспекта безбедности, јесте заснованост на *IP* протоколу. То значи да ће бити могуће применити постојеће системе заштите развијане за рачунарске мреже и у мрежи мобилне телефоније, односно за заштиту података који се преносе између мобилних уређаја.

У четвртој генерацији мобилне телефоније ће се користити шеста верзија Интернет протокола. Ово представља логичан избор с обзиром на бројност мобилних уређаја и време планирано за масовни прелазак на четврту генерацију мобилне телефоније, односно, на недостатак адреса Интернет протокола четврте генерације. Гледано са безбедносног аспекта, једна од значајних карактеристика шесте генерације Интернет протокола јесте интегрисана подршка за *IPsec* сет протокола, односно, компатибилност са решењем представљеним у овом раду.

2.2. Неки типови и модели напада

Данас у пракси постоји велики број различитих типова напада на рачунарске системе и комуникационе канале. Постоје различити типови напада а они се (углавном) могу поделити у суштински различите групе у оквиру којих различити напади користе исте принципе, а разликују се у практичној реализацији и мети напада. У оквиру овог дела рада биће представљена три суштински различита типа напада - два (разбијање шифрата применом сирове силе, посредовање у комуникацији) на која се мора обезбедити имунитет сопственог решења и један (ускраћивање услуге) од кога се у постављеним оквирима рада није могуће бранити.

2.2.1. Криптоанализа

Термин криптоанализа односи се на део криптологије везан за анализу и дешифрирање шифрата без познавања једног или више параметара (кључева, трансформације и сл.) помоћу којих је он креиран. Иако је криптоанализа супротстављена криптографији (која је усмерена на развој шифарских система) она ипак чини и изузетно важан алат за

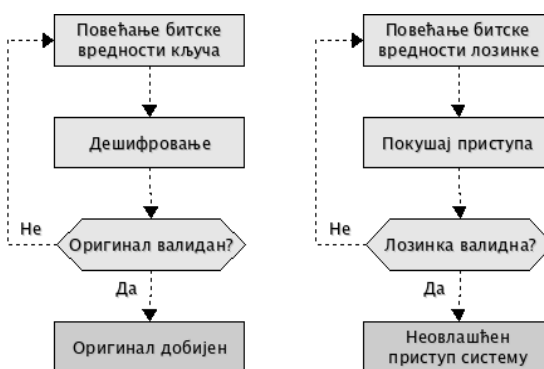
проверу ефикасности шифарских система. Криптоанализа представља најсложенију област криптологије.

Резултат успешне криптоанализе јесте добијање оригинала на основу шифрата. Уколико процес криптоанализе да такав резултат, анализирани шифарски систем се сматра „пробијеним“. Уколико се криптоанализа примењује на дигиталне потписе, њени циљеви могу бити фалсификовање потписа или проналажење тајног кључа.

Један од значајних примера криптоанализе јесте њена примена на *RSA* алгоритам [12] који данас представља *de facto* стандард у асиметричном шифровању. Носеће функције овог алгоритма већ деценијама су предмет анализе математичара и професионалаца на пољу безбедности. До сада нису објављени резултати криптоанализе овог алгоритма, што није случај са многим шифарским алгоритмима који су се раније користили, или се још увек користе.

2.2.2. Разбијање шифрата применом сирове силе

Један од најелементарнијих начина за откривање заштићених приступних параметара систему или оригиналу шифрата (корисничког имена и лозинке, тајног кључа и сл.) јесте коришћењем тзв. „сирове силе“ (енгл. *brute force attack*). Овај приступ у основној варијанти подразумева редно испробавање различитих приступних параметара све до погађања одговарајућих.



Слика 4. Алгоритам метода „сирове силе“

Главни недостатак примене метода „сирове силе“ јесте време потребно за разбијање лозинки и кључева веће битске дужине, јер сваки додатни бит лозинке или кључа дуплира потребно време (експоненцијални раст). Из тог разлога се у пракси (углавном код разбијања приступних лозинки) често користе модификовани алгоритми и речници

често коришћених израза.

Принципи и могућности одбране од овог типа напада знатно се разликују у зависности од тога да ли се штити приступ систему или шифровани подаци који се преносе путем мреже. За случај заштите система од неовлашћеног приступа као пример се може узети *UNIX* оперативни систем, његови безбедносни механизми и њихова еволуција.

У раним фазама развоја рачунара, њихова процесорска снага и број корисника били су (са безбедносног аспекта) занемарљиви те су приступне лозинке корисника чуване у шифрованом облику (хеш функције) у јавно доступном фајлу */etc/passwd*. Међутим, упоредо са развојем процесорске снаге рачунара и растом броја корисника јавили су се први случајеви злоупотребе приступних параметара осталих корисника система који су били откривени применом метода „сирове силе“ на шифрате лозинки. Из овог разлога шифрати лозинки су измештени у фајл */etc/shadow* коме је приступ одобрен само администратору система и процесу који врши пријављивање корисника на систем.

Следећи безбедносни проблем који се јавио јесте појава алата који су за метод примене „сирове силе“ искористили системску наредбу за промену идентитета *su* и на тај начин искористили посредан приступ хеш вредности лозинке путем процеса за пријављивање корисника на систем. Ови алати су омогућавали максимално искоришћавање перформанси система за разбијање хеш вредности лозинке тиме што се могло испитати више десетина, стотина или хиљада вредности (у зависности од процесорске снаге рачунара) у току једне секунде. Овај безбедносни проблем је решен изменом система за пријаву корисника на систем тако да су додата чекања од по неколико секунди између два покушаја пријављивања. На *MS Windows* оперативним системима постоји и опција да се налози аутоматски закључавају након неколико неуспешних покушаја пријављивања. Исти принцип се користи и код тзв. „паметних“ картица и њима придруженим приступним *PIN* кодовима. Овај метод заштите, иако ефикасан у заштити од неовлашћеног приступа, често се може злоупотребити од стране нападача у циљу онемогућавања нормалног функционисања система и рада легалних корисника.

За разлику од заштите система од неовлашћеног приступа, код преноса података рачунарским и телекомуникационим мрежама заштиту није могуће засновати на недоступности шифрата нападачу. У таквој ситуацији се као решење најчешће предлаже коришћење кључева веће битске дужине. Постоје различите препоруке [13] за

избор дужине кључа у циљу одбране од напада методом примене „сирове силе“ а у зависности од начина шифровања, области примене, важности података и сл.:

National Security Agency (NSA)			
Ниво	Симетрично	ECC	Хеш
<i>Secret</i>	128	256	256
<i>Top Secret</i>	256	384	384

Табела 1. NSA препоруке дужина кључева

ECRYPT							
Ниво	Примена	Сим.	Асим.	Дис. логаритми		ECC	Хеш
				Кључ	Група		
1	<i>Attacks in "real-time" by individuals Only acceptable for authentication tag size</i>	32	-	-	-	-	-
2	<i>Very short-term protection against small organizations Should not be used for confidentiality in new systems</i>	64	816	128	816	128	128
3	<i>Short-term protection against medium organizations, medium-term protection against small organizations</i>	72	1008	144	1008	144	144
4	<i>Very short-term protection against agencies, long-term protection against small organizations Smallest general-purpose level, Use of 2-key 3DES restricted to 240 plaintext/ciphertexts, protection from 2008 to 2011</i>	80	1248	160	1248	160	160
5	<i>Legacy standard level Use of 2-key 3DES restricted to 106 plaintext/ciphertexts, protection from 2008 to 2018</i>	96	1776	192	1776	192	192
6	<i>Medium-term protection Use of 3-key 3DES, protection from 2008 to 2028</i>	112	2432	244	2432	244	244
7	<i>Long-term protection Generic application-independent recommendation, protection from 2008 to 2038</i>	128	3248	256	3248	256	256
8	<i>"Foreseeable future" Good protection against quantum computers</i>	256	15424	512	15424	512	512

Табела 2. ECRYPT препоруке дужина кључева

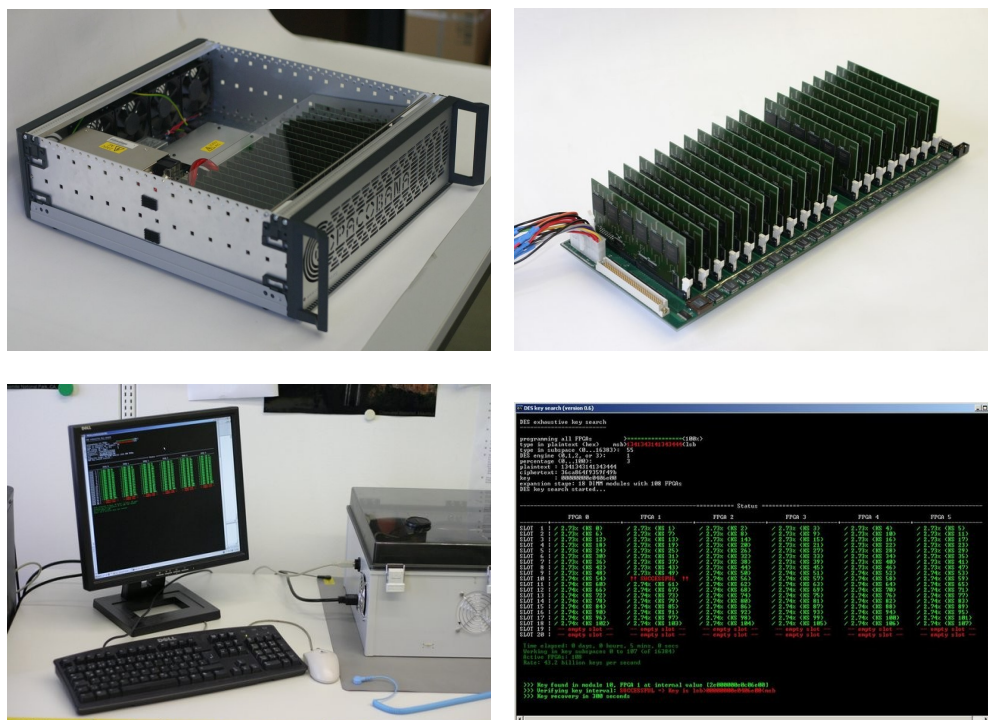
Bundesamt für Sicherheit in der Informationstechnik (BSI)						
Година	Асиметрично	Тајни кључ	Лог. величина	ECC	Хеш	
2008	1280	160	1280	180	SHA-1* RIPEND-160 SHA-224	SHA-256 SHA-384 SHA-512
2009	1536	160	1536	180	SHA-1* RIPEND-160 SHA-224	SHA-256 SHA-384 SHA-512
2010	1728	224	2048	224	SHA-1** RIPEND-160 SHA-224	SHA-256 SHA-384 SHA-512
2011 ... 2015	1967	224	2048	224	SHA-224 SHA-256	SHA-384 SHA-512
(*) Предложена употреба само за дигиталне сертификате.						
(**) Предложена употреба само за дигиталне сертификате са минималном ентропијом од 20 битова.						

Табела 3. BSI препоруке дужина кључева

Време потребно за разбијање шифрата применом „сирове силе“ у општем случају зависи од два параметра – дужине коришћеног кључа (величине скупа из кога се бира кључ) и фреквенције којом се испробавају различити кључеви. Очекивана вероватноћа [14] да ће кључ бити погођен износи 2^{N-1} проверених комбинација где је N битска дужина кључа.

У циљу остваривања што краћег времена за разбијање шифрата најчешће се користе брзи рачунари са више процесора, кластери рачунара или наменски уређаји. Пример наменског уређаја за разбијање шифрата применом „сирове силе“ је уређај *COPACOBANA* [15] развијен на немачким универзитетима *Bochum* и *Kiel*.

Овај уређај је намењен разбијању свих симетричних система шифровања који користе кључеве дужине до 64 бита. У систем је уграђена подршка за *DES* алгоритам а систем се може проширити и сопственим модулима. Такође, могуће је и серијско повезивање до 127 *COPACOBANA* уређаја који би паралелно радили на истом задатку. Просечна брзина претраге система је око 43 милијарде кључева у секунди а снага система је 2006. године упоређена са 22.865 Pentium 4 рачунара. Цена једног уређаја је октобра 2006. године износила 8.980 евра.



Слика 5. Изглед, структура и коришћење *COPACOBANA* уређаја

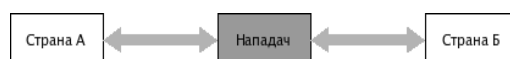
Као што је раније поменуто и приказано, у литератури и од стране различитих организација које се баве безбедношћу комуникација, као подразумевани метод заштите од разбијања шифрата применом сирове силе наводи се коришћење кључева веће битске дужине јер се полази од основне претпоставке да је разбијање шифрата симетричних алгоритама могуће само применом „сирове силе“. Међутим, једна од теза на којима се заснива овај рад јесте да веће битске дужине могу представљати препреку за нападаче који за разбијање користе рачунарске системе скромнијих процеских моћи али не и за нападаче који располажу кластерима супер-рачунара или наменских уређаја. У складу са тим, као нова компонента безбедности предлаже се (а у даљем раду ће бити и детаљно образложена и објашњена) употреба сопственог шифарског алгорита који је (пored тајног кључа) такође непознат потенцијалном нападачу.

2.2.3. Напад посредовањем у комуникацији

Један од најопаснијих напада у рачунарским и телекомуникационим мрежама јесте напад путем посредовања између страна које комуницирају (енгл. *Man in the middle attack*). С обзиром на то да данашње глобалне рачунарске и телекомуникационе мреже најчешће подразумевају велики број инфраструктурних посредника у комуникацији

(рутери, репетитори и сл.), могућности за извршавање оваквог типа напада су све веће.

Напад посредовања у комуникацији се, пре свега, односи на заштићене (шифроване) и ауторизоване комуникације. Код незаштићених и неауторизованих комуникација су подразумевано остварени сви предуслови за успешан напад. Основни резултати посредовања у комуникацији јесу могућност прислушкивања (односно снимања) комуникације и могућност измене њеног садржаја. Предуслов за успешно извршавање напада јесте успешно лажно представљање код обе стране, односно, симулирање идентитета друге стране.



Слика 6. Структура напада са посредовањем у комуникацији

Процес извођења напада посредовања у комуникацији подразумева прихватање, читање, евентуалну измену, и прослеђивање порука које стране размењују. Код успешно изведеног напада регуларне комуникационе стране имају илузију да комуницирају директно и немају никакву могућност да утврде да је дошло до прислушкивања.

Постоји више механизма за одбрану од напада посредовања у комуникацији: коришћење *PKI* инфраструктура, тајних кључева, посебних канала за размену, итд. У овом раду су описана решења која се базирају на употреби симетричних алгоритама са тајним кључевима који се размењују одвојеним комуникационим каналима. Такав приступ се може сматрати безбедним од напада посредовања у комуникацији.

2.2.4. Напад ускраћивањем услуге

Онемогућавање услуге (енгл. *Denial of Service, DoS*) је тип напада усмерен ка комуникационој структури, односно, њеном капацитету (пропусној моћи). Циљ напада овог типа није да нападачу омогући приступ или контролу над удаљеним системом (или подацима) већ да заузимањем свих комуникационих капацитета онемогуће приступ систему од стране регуларних корисника.

Постоји више типова напада са циљем онемогућавања услуге. Најједноставнији типови напада овог типа се базирају слању што веће количине података жртви у што мањем временском периоду.

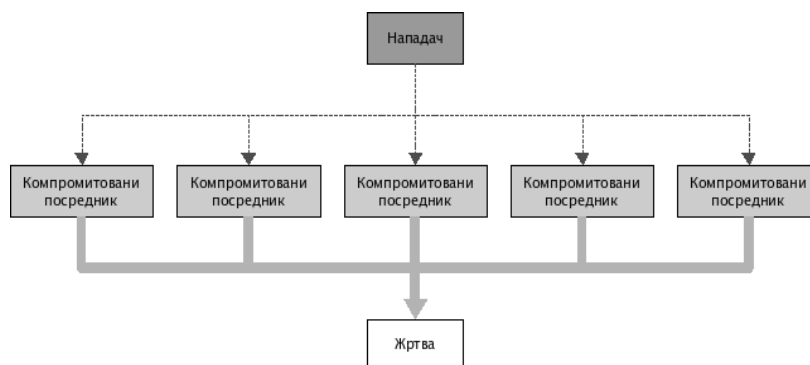
Нешто напреднији тип напада је напад са модификованим пакетима мрежног нивоа.

Пример оваквог типа напада је тзв. рефлектовани напад. Код овог типа напада нападач посреднику (који није компромитован – под нападачевом контролом) шаље пакете који захтевају одговор а код којих је за изворишна адреса (адреса нападача) замењена адресом жртве. Ово је уједно и најпримитивнији облик сакривања извора напада.



Слика 7. Модел рефлектованог напада са циљем онемогућавања услуге

Један од најопаснијих типова напада са циљем онемогућавања услуге је дистрибуирани напад са циљем онемогућавања услуге (енгл. *Distributed Denial of Service, DDoS*). Главна карактеристика оваквих напада је то што нападач не зависи од капацитета сопствених комуникационих канала већ за напад може истовремено користити капацитете хиљада компромитованих посредника.



Слика 8. Модел дистрибуираног напада са циљем онемогућавања услуге

Код овог типа напада самом нападу претходи процес компромитовања што већег броја посредника (који се понекад називају и „зомбијима“). За компромитовање посредника најчешће се користи малициозни софтвер. На компромитоване посреднике се инсталира софтвер за удаљену контролу који, када нападач добије адресу жртве, започиње слање података. Два значајна проблема при одбрани од овог типа напада су та што се јавља велико оптерећење комуникационих канала и што постоји велики број извора напада.

Средином 2008. године примећен је нови тип напада са циљем онемогућавања услуге. Овај напад није усмерен на загушивање комуникационог канала жртве већ на њен

мрежни интерфејс. Редослед корака у извршавању овог типа напада је следећи:

1. Кориснички програм на страни нападача са нивоа *IP* протокола шаље жртви *IP* датаграм са садржајем који је идентичан садржају *TCP* пакета за отварање сокета (иницирање везе).
2. Нападнути рачунар након пријема датаграма прослеђује садржај транспортном нивоу - *TCP* протоколу – који отвара нови сокет сматрајући да се ради о регуларном захтеву.
3. Процес се понавља док се комуникација са нападнутим рачунаром не онемогући превеликим бројем отворених сокета.

Напади са циљем онемогућавања услуге не представљају реткост на Интернет мрежи а готово да их нема у локалним мрежама. Претпоставља се чак и да је ово један од облика информационих ратова између земаља и великих привредних субјеката. На пример, по речима Хозеа Назариа, мрежна компанија Арбор Нетворкс из Лексингтона је 20. јула 2008. године приметила [16] велику количину саобраћаја из Русије усмерену ка сајтовима грузијског парламента након чега су ти сервери искључени. Такође, оригиналан садржај сајта грузијског парламента на адреси <http://mfa.gov.ge> замењен је садржајем који су поставили руски хакери. Тачан идентитет нападача никада није утврђен.

Напади са циљем онемогућавања услуге могу проузроковати озбиљне проблеме корисницима који зависе од комуникационих канала високе пропусне моћи (Интернет провајдери, велики пословни субјекти, он-лајн пословања и сл.). Постоје различити начини одбране од оваквих напада (конфигурација рутера, алтернативни линкови итд.). Овај рад се, међутим, неће детаљније бавити проблемом напада са циљем онемогућавања услуге јер они излазе ван оквира постављене тезе.

2.3. Најчешће коришћени системи за заштиту комуникације

Данас се у пракси рекламира и среће велики број различитих система заштите. Ови системи су пројектовани у складу са различитим наменама, окружењима и начинима употребе. Неки од њих су пројектовани за крајње кориснике, а неки пре представљају радне оквире за развој сопствених безбедносних конфигурација.

Категоризацију система заштите могуће је обавити по више критеријума:

- системи за заштиту издвојених система и системи за заштиту мрежа,
- системи са доступним изворним кодом и затворени системи,
- системи који користе јавне алгоритме и системи са сопственим алгоритмима и т.д.

Преглед постојећих система заштите комуникације за потребе овог рада извршена је категоризација на основу слоја *OSI* и *TCP/IP* референтних модела на коме ти системи функционишу. У складу са тим, дат је преглед система заштите комуникације који функционишу на:

- нивоу апликативног протокола,
- нивоу транспортног протокола и
- нивоу мрежног протокола.

Системи за заштиту комуникације на нижим слојевима (физичком и слоју везе) нису разматрани с обзиром на постављени циљ рада - интеграцију система заштите у глобалну телекомуникациону инфраструктуру.

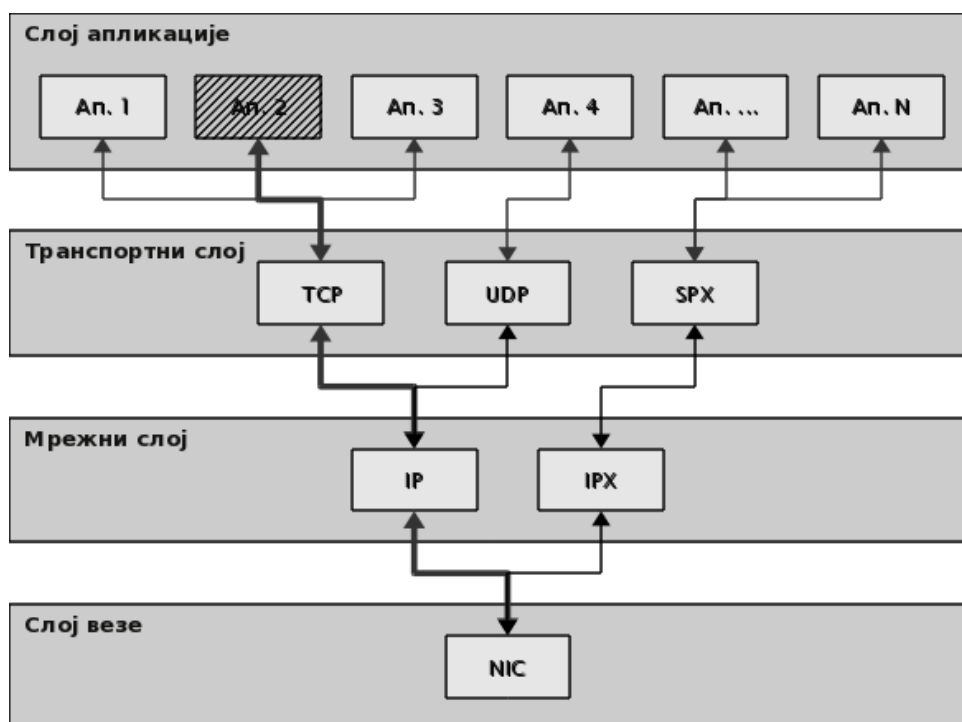
2.3.1. Заштита на нивоу апликативног протокола

Апликативни слој представља најфлексибилније подручје за развој сопствених система за заштиту комуникације из тог разлога што се при развоју протокола на овом нивоу не мора водити рачуна о компатибилности са протоколима виших нивоа. Безбедност сопственог система заштите на апликативном слоју зависи првенствено од нивоа заштите који нуде коришћени алгоритми, процедуре и окружење у коме се систем примењује.

Када је у питању измена постојећих протокола у циљу интеграције сопствених система заштите, олакшавајућу околност чини то што је њихова спецификација јавно доступна. Са друге стране, као проблем се јавља то што је већина имплементација затворена, односно, њихов изворни код није доступан. То значи да се конкретна имплементација мора развијати „од нуле“ или се као основа мора искористити нека од имплементација са јавно доступним изворним кодом.

Основни недостатак интеграције сопствених безбедносних система на нивоу апликативног протокола јесте тај што се развојем/интеграцијом једног безбедносног

система покрива само један апликативни протокол.



Слика 9. Заштита на нивоу апликативног протокола

Уколико корисници комуникацију врше путем различитих протокола, тада се за сваки од протокола мора извршити засебан развој/интеграција. Додатно, у таквој ситуацији се сет софтвера који користе крајњи корисници мора ограничити да не би дошло до размене важних података протоколима који нису заштићени.

2.3.1.1. Заштита електронске поште

Електронска пошта представља један од најчешће коришћених сервиса на Интернет мрежи. Једна од значајних карактеристика овог сервиса је та да се у оквиру мрежног саобраћаја који се њиме остварује налази највећи проценат података генерисаних од стране корисника. У складу са тим постоји више система за заштиту података који се преносе путем сервиса електронске поште. Ови системи се углавном базирају на шифровању података. Један од најпопуларнијих система заштите електронске поште јесте *OpenPGP*.

OpenPGP систем обезбеђује услуге заштите и интегритета података електронске поште и фајлова коришћењем дигиталних потписа, шифровања, компресије и *Radix/64* конвертовања [RFC 4880]. Додатно, овај систем обезбеђује и сервисе за управљање

кључевима и сертификатима. *OpenPGP* комбинује симетрично и асиметрично шифровање за обезбеђивање поверљивости. Код симетричног шифровања један кључ се користи само једном, за шифровање једне поруке или фајла. Нови сесијски кључ генерише се као (псеудо)случајан број и шаље се уз поруку, шифрован јавним кључем примаоца. Кораци у коришћењу *OpenPGP* система заштите су:

1. Пошиљалац креира поруку,
2. *OpenPGP* генерише (псеудо) случајан број као симетрични кључ за шифровање поруке,
3. *OpenPGP* шифрује сесиони кључ јавним кључем примаоца и такав шифрат ставља на почетак поруке,
4. *OpenPGP* шифрује садржај поруке и њиме наставља поруку започету шифратом сесионог кључа,
5. прималац својим тајним кључем дешифрује сесијски кључ придружен поруци,
6. прималац коришћењем дешифрованог сесијског кључа дешифрује садржај поруке.

Уколико је у основној поруци дефинисано више прималаца, за сваког од њих се на почетак поруке ставља сесијски кључ шифрован његовим јавним кључем. Уколико се у систему користи и дигитално потписивање, дигитални потпис поруке се израчунава пре шифровања, а затим се шифрује по истом принципу као и садржај поруке. Процес дигиталног потписивања подразумева следеће кораке:

1. Пошиљалац креира поруку,
2. систем израчунава хеш вредност поруке,
3. систем израчунава дигитални потпис коришћењем хеш вредности и тајног кључа пошиљаоца,
4. дигитални потпис се придружује поруци,
5. прималац прихвата поруку и израчунава њену хеш вредност,
6. израчуната хеш вредност се упарује са јавним кључем пошиљаоца и упоређује са добијеним дигиталним потписом.

OpenPGP систем заштите нуди и могућност компресовања порука. Компресовање се врши након израчунавања дигиталног потписа, а пре шифровања садржаја. У заглавље поруке се поставља идентификациони број алгорита за компресовање:

<i>ID</i>	Алгоритам
0	нема компресовања
1	<i>ZIP</i> [RFC 1951]
2	<i>ZLIB</i> [RFC 1950]
3	<i>BZip2</i>
100-110	приватни и експериментални алгоритми

Табела 4. Алгоритми за компресовање података у *OpenPGP* систему

Као алгоритми за израчунавање хеш вредности порука користе се следећи алгоритми (идентификациони број алгорита се поставља у заглавље поруке):

<i>ID</i>	Алгоритам
1	<i>MD5</i>
2	<i>SHA-1</i>
3	<i>RIPE-MD/160</i>
4	резервисано
5	резервисано
6	резервисано
7	резервисано
8	<i>SHA256</i>
9	<i>SHA384</i>
10	<i>SHA512</i>
11	<i>SHA224</i>
100-110	приватни и експериментални алгоритми

Табела 5. Алгоритми за израчунавање хеш вредности у *OpenPGP* систему

Као симетрични шифарски алгоритми користе се:

<i>ID</i>	Алгоритам
0	нема шифровања
1	<i>IDEA</i>
2	<i>3DES (168bit)</i>
3	<i>CAST5 (128bit)</i> [RFC 2144]

4	<i>Blowfish (128bit)</i>
5	резервисано
6	резервисано
7	<i>AES (128bit)</i>
8	<i>AES (192bit)</i>
9	<i>AES (256bit)</i>
10	<i>Twofish (256bit)</i>
100-110	приватни и експериментални алгоритми

Табела 6. Симетрични шифарски алгоритми у *OpenPGP* систему

Као асиметрични шифарски алгоритми користе се:

<i>ID</i>	Алгоритам
1	<i>RSA</i> (шифровање и потписивање)
2	<i>RSA</i> (само шифровање)
3	<i>RSA</i> (само потписивање)
16	<i>Elgamal</i> (само шифровање)
17	<i>DSA</i>
18	резервисано
19	резервисано
20	резервисано
21	резервисано
100-110	приватни и експериментални алгоритми

Табела 7. Асиметрични шифарски алгоритми у *OpenPGP* систему

На основу саме спецификације *OpenPGP* види се да је могућност додавања нових (сопствених) алгоритама за шифровање, потписивање и компресовање пројектована карактеристика система. У самој спецификацији система као пример се наводи додавање симетричног шифарског алгоритма. Процес додавања сопственог симетричног шифарског алгоритма подразумева дефинисање константе за њега. Уколико алгоритам подразумева рад са блоковима битске дужине различите од 64 и 128, доступна је документација у којој је дефинисано подешавање *OpenPGP-CFB* режима рада. Као пример је наведена и модификација *DSA* алгоритма у циљу коришћења кључева битске дужине од 3072. Уколико се жели јавна употреба предложеног новог алгоритма у *OpenPGP* систему потребно је обратити се *IETF* организацији и проћи кроз њен процес концензуса.

2.3.1.2. Протокол *SSH*

SSH (*Secure Shell*) протокол омогућава безбедан приступ и рад на удаљеним рачунарским системима [RFC4251]. Првенствено се користи на *UNIX* базираним оперативним системима. Три основне компоненте овог протокола чине протокол транспортног слоја, протокол за проверу аутентичности и протокол везе.

Протокол транспортног слоја који обезбеђује аутентичност, интегритет и поверљивост - најчешће користи услуге *TCP/IP* протокола али се могу искористити и остали протоколи који омогућавају поуздан пренос података. Улога овог протокола је да обезбеди безбедан канал користећи небезбедну инфраструктуру. Овај протокол обезбеђује и јединствени идентификатор сесије који се може користити од стране протокола вишег нивоа.

Протокол за проверу аутентичности корисника на клијентској страни - такође функционише на транспортном нивоу. Механизми овог протокола користе поменути идентификатор сесије који обезбеђује протокол транспортног слоја.

Протокол везе који мултиплексује шифровани тунел у више логичких канала - лежи на протоколу за аутентичност.

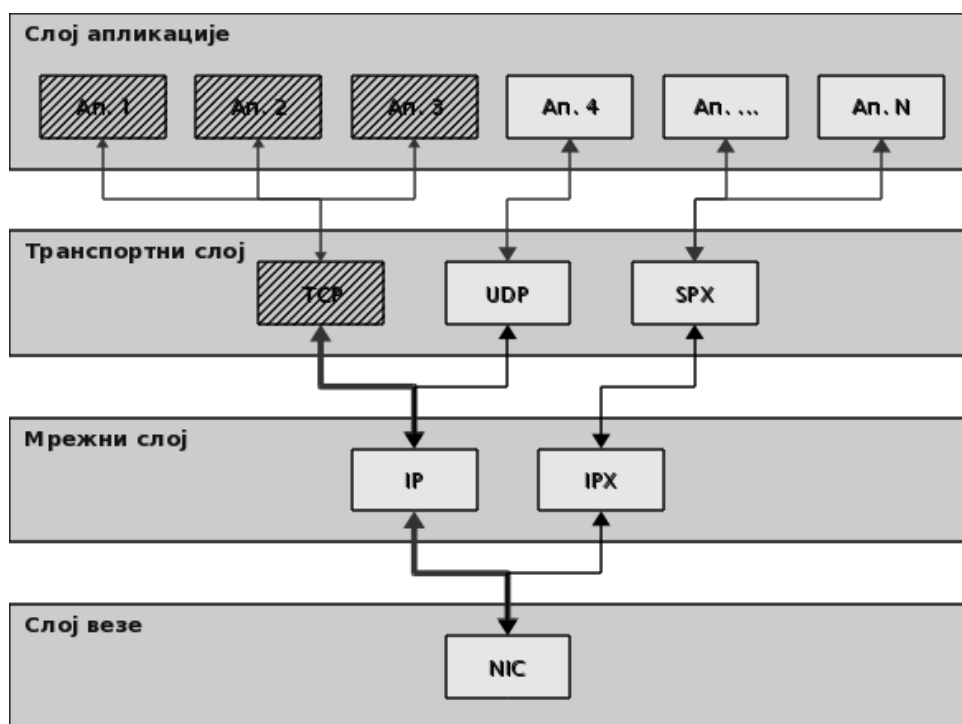
Протокол везе који обезбеђује канале који се могу користити за различите примене - најчешће за сесије алфанумеричког корисничког интерфејса и тунеловање *X11* протокола. У оквиру *SSH* протокола алгоритми и методи за шифровање, проверу интегритета, израчунавање хеш вредности, компресију и размену кључева дефинисани су по називу. Неки од алгоритама су обавезни за све имплементације. У оквиру спецификације алгорита подразумевана је могућност проширивања скупа алгоритама одређене намене сопственим алгоритмом.

Сесијски кључеви се код *SSH* протокола генеришу коришћењем (псеудо)случајног генератора. Квалитет овог генератора један је од пресудних фактора безбедности коју нуди *SSH* протокол.

SSH протокол може бити рањив нападима ускраћивања услуге. Релативно захтевне процедуре на серверу нападач може искористити за заузимање меморијских и процесорских ресурса сервера. Ово питање првенствено зависи од конкретне имплементације протокола.

2.3.2. Заштита на нивоу транспортног протокола

Реализација сопственог система заштите комуникације на нивоу транспортног протокола *OSI* и *TCP/IP* референтних модела, уместо на нивоу апликативног протокола, омогућила би ширу примену јер би се на тај начин једном реализацијом могле обезбедити комуникације већег броја апликација.



Слика 10. Заштита на нивоу транспортног протокола

На пример, уградњом безбедносних функција у *TCP* протокол могао би се заштитити огроман број апликативних протокола који користе поменути транспортни протокол. Међутим, измена самог *TCP* протокола представља веома сложен захват који би могао имати значајне последице везане за употребљивост окружења у коме би се тако измењен протокол примењивао. На пример, измењена варијанта *TCP* протокола, са циљем комуникације преко небезбедне мреже (нпр. Интернета), могла би довести до престанка рада сервиса локалне мреже на којима је присутна оригинална верзија *TCP* протокола (нпр. сервиса за мрежно штампање докумената).

Једно од најелегантнијих решења за имплементацију сопственог система заштите на транспортном нивоу јесте интеграција његових компонената у наменски протокол - протокол за заштиту података на транспортном нивоу. Најраспрострањенији протокол овог типа јесте *TLS* (енгл. *Transport Layer Security*). Он је развијен на основу верзије 3.0

популарног *SSL* (енгл. *Secure Sockets Layer*) протокола. Претходне верзије *SSL* протокола (верзија 2.0 и раније) садрже озбиљне безбедносне пропусте везане за интегритет порука, *Man-in-the-middle* напад и напад одсецањем података. Додатно, извозне регулативе код ранијих верзија ограничавају коришћене кључеве на дужину од 40 битова.

2.3.2.1. *Transport Layer Security* протокол

Основни задатак *TLS* протокола [17] је обезбеђивање приватности и интегритета података у оквиру комуникације између две апликације. За развој спецификације протокола задужена је наменска радна група при *IETF* организацији. Развој овог протокола започет је на основу верзије 3.0 *SSL* протокола. До сада, група је објавила спецификацију верзија 1.0 [RFC2246], 1.1 [RFC4346] и 1.2 [RFC5246] протокола, спецификацију одређених проширења [RFC 2817, 2818, 3546, 3749, 4366, 5054, 5081, 5705, 5746], и спецификације компатибилних шифарских алгоритама и њихових режима рада [RFC: 2712, 3268, 4132, 4279, 4492, 4785, 5288, 5289, 5469, 5487, 5489]. Сам *TLS* протокол састоји се од два слоја:

1. *TLS Record* протокол
2. *TLS Handshake* протокол

TLS Record протокол се налази на нижем нивоу и ослања се на неки транспортни протокол, најчешће на *TCP* протокол. Овај протокол је задужен за остваривање две основне карактеристике безбедне комуникације:

1. Приватност комуникације - заснована на симетричној криптографији. Јединствени кључеви се генеришу код сваког остваривања нове везе а успостављају се коришћењем *TLS Handshake* протокола. Сам *TLS Record* протокол се може користити и без функције шифровања.
2. Поузданост везе - заснована на провери интегритета порука базираној на хеш алгоритму (нпр. *SHA-1*). Провера интегритета пренесених порука није обавезна функционалност *TLS Record* протокола али се у пракси готово увек користи.

TLS Handshake протокол за задатак има да провери аутентичност комуникационих страна, као и да утврди параметре (шифарски алгоритам и кључеве) који ће бити

коришћени за заштиту комуникације. Ови параметри се утврђују пре саме комуникације, односно, пре преношења података добијених од протокола апликативног нивоа. *TLS Handshake* протокол се енкапсулира у *TLS Record* протокол. Три основне карактеристике безбедности коју остварује *TLS Handshake* протокол су:

1. Идентитет комуникационих страна утврђује се коришћењем асиметричног шифарског алгоритма (нпр. *RSA*, *DSA* и сл.). Провера аутентичности је неопходна, макар за једну комуникациону страну.
2. Уговарање кључа се одвија у оквиру заштићене комуникације и он није доступан посредним чворовима у комуникацији.
3. Води се рачуна о интегритету уговореног кључа тако да се он не може мењати од стране нападача.

TLS протоколом могу се штитити различити апликативни протоколи. Као основни циљеви при дизајнирању овог протокола постављени су:

1. Безбедност - заштићена комуникација две стране путем коришћења алгоритама за шифровање.
2. Интероперабилност - различити аутори протокола на апликативном нивоу могу користити *TLS* протокол и размену безбедносних параметара без увида у туђ изворни код.
3. Проширивост - обезбеђивање оквира у који се могу уградити нови системи за размену кључева и алгоритми за шифровање. На тај начин се избегава потреба за развојем нових протокола исте намене и за увођењем комплетно нових безбедносних библиотека.
4. Релативна ефикасност - с обзиром на интензивно коришћење централног процесора за извршавање криптографских операција у *TLS* протокол је уграђена опциона шема за кеширање у оквиру сесије. Њоме се смањује број веза које треба упоставити из почетка. Такође, при дизајнирању овог протокола обраћена је пажња и на одржавање мрежне активности на оптималном нивоу.

При дизајнирању протокола који би користио сам *TLS* систем мора се водити рачуна о могућим нападима на њега. То у пракси значи да дизајнери протокола морају имати у виду које безбедносне функције *TLS* протокол нуди, а које не. На пример, сам запис и

дужина записа нису подаци који се шифрују у оквиру пакета. Уколико ови подаци имају значај за безбедност у оквиру самог апликативног протокола потребно је уградити додатне системе заштите.

2.3.2.1.1. Протокол *TLS Record*

TLS Record протокол је слојевит протокол, а на сваком од слојева поруке могу садржати поља везана за дужину, опис и сам садржај. Овај протокол од апликативног протокола прихвата податке које треба пренети, фрагментује их у блокове, компресује их (опциона функционалност), израчунава кодове за проверу аутентичности (енгл. *Message Authentication Code, MAC*), и шаље резултат наведених операција другој комуникационој страни. На пријемној страни добијени подаци се дешифрују, проверавају, декомпресују, састављају и испоручују клијентском протоколу на апликативном нивоу.

TLS Record протокол се користи од стране четири различита (типа) протокола:

1. *TLS Handshake* протокола,
2. протокола за слање контролних протокола,
3. протокола за измену параметара коришћеног шифарског алгорита, и
4. протокола на нивоу апликативног слоја.

За основно проширивање самог *TLS* протокола подразумева се додавање нових типова садржаја које подржава *TLS Record* протокол. За примену сопствених проширења на Интернет мрежи потребно је слање захтева организацији *IANA* и регистрација проширења у њеном *TLS Content Type* регистру. У случају пријема недефинисаног записа, очекивано понашање *TLS* имплементације је слање упозорења *unexpected_message*.

Сви записи *TLS Record* протокола обрађују се у оквиру тренутног слања или пријема. За слање и пријем користе се три алгорита: алгоритам за компресију података, шифарски алгоритам и *MAC* алгоритам. Параметри који се користе у оквиру ових алгорита за процес слања и пријема записа чине:

- комуникациона страна - клијент или сервер (условна терминологија).

- *PRF* алгоритам - алгоритам задужен за генерисање кључева на основу тајне вредности.
- Алгоритам за шифровање - спецификација алгоритма за шифровање. Подразумеване вредности спецификације су: дужина кључева, тип алгоритма (блоковски или секвенцијални), дужине иницијализационих вектора.
- *MAC* алгоритам - спецификација алгоритма који обезбеђује аутентичност порука.
- Алгоритам за компресовање података - спецификација свих параметара алгоритма којим се врши компресовање и декомпресовање података.
- *Master secret* - тајна вредност дужине 48 бајтова, позната обема странама у комуникацији.
- Случајна вредност клијента - (псеудо)случајна вредност, израчуната на страни клијента.
- Случајна вредност сервера - (псеудо)случајна вредност, израчуната на страни сервера.

Након постављања наведених параметара и генерисања кључева веза се проглашава успостављеном. Стање везе се обнавља након сваког пренетог записа. У параметре стања везе спадају:

- Стање алгоритма за компресију података.
- Стање алгоритма за шифровање података (посебно значајно код секвенцијалних алгоритама).
- *MAC* кључ генерисан за активну везу.
- Секвентни број. Овај број се одржава независно за слање и пријем података. Иницијална вредност овог броја, након успостављања везе, постављена је на нулу. Највећа могућа вредност секвентног броја је $2^{64}-1$. Након достизања највеће могуће вредности она се не враћа на нулу већ се поновно уговара. Секвентни број се повећава након сваког слања, односно пријема записа.

Једна од битних функционалности *TLS Record* протокола јесте и фрагментовање података добијених од стране протокола који користи услуге самог *TLS* протокола.

Добијени подаци се деле у блокове (*TLSPlaintext* записе) највеће дозвољене дужине 2^{14} бајтова.

```

struct {
    uint8 major;
    uint8 minor;
} ProtocolVersion;

enum {
    change_cipher_spec(20), alert(21), handshake(22),
    application_data(23), (255)
} ContentType;

struct {
    ContentType type;
    ProtocolVersion version;
    uint16 length;
    opaque fragment[TLSPlaintext.length];
} TLSPlaintext;

```

Листинг 1. Декларације структуре *TLSPlaintext* и референтних структура

Структуру фрагмената чине следећа поља (параметри):

1. Идентификација протокола вишег нивоа који тренутно користи услуге *TLS* протокола.
2. Верзија протокола која се користи. Тренутно актуелна верзија *TLS* протокола је 1.2. Међутим, у употреби су и верзије 1.1 и 1.0. У току успостављања везе клијент и сервер одређују верзију која ће се користити (највиша верзија из пресека подржаних верзија обе стране).
3. Бајтовска дужина фрагмента, мања или једнака 2^{14} бајтова.
4. Садржај фрагмента. Чине га подаци (апликативног) протокола који користи услуге *TLS* протокола.

TLS Record протокол има подршку за компресовање података који се смештају у фрагмент. Алгоритам којим се врши компресовање и декомпресовање података мора ту функцију вршити без губитака. Активан алгоритам за компресовање дефинисан је параметром *CompressionMethod* а иницијално је постављен на вредност *null*. Могући алгоритми за компресовање података код *TLS* протокола дефинисани су у документу *RFC3749*. Функција компресовања се над добијеним подацима извршава пре

шифровања и израчунавања кодова за проверу аутентичности. Применом функције компресовања структура података *TLSPplaintext* преводи се у структуру података *TLSCompressed*.

```
struct {
    ContentType type;          /* same as TLSPplaintext.type */
    ProtocolVersion version; /* same as TLSPplaintext.version */
    uint16 length;
    opaque fragment[TLSCompressed.length];
} TLSCompressed;
```

Листинг 2. Декларације структуре *TLSCompressed*

Применом функција за шифровање и израчунавање кодова за проверу аутентичности структура података *TLSCompressed* се преводи у структуру података *TLSCiphertext*. На пријемној страни се одвија супротан процес, односно, структуру *TLSCiphertext* се преводи у структуру *TLSCompressed*.

```
struct {
    ContentType type;
    ProtocolVersion version;
    uint16 length;
    select (SecurityParameters.cipher_type) {
        case stream: GenericStreamCipher;
        case block:  GenericBlockCipher;
        case aead:   GenericAEADCipher;
    } fragment;
} TLSCiphertext;
```

Листинг 3. Декларације структуре *TLSCiphertext*

MAC кодови се користе за проверу аутентичности порука а у комбинацији са секвентним бројем омогућавају уклањање вишеструко примљених фрагмената, као и исправљање њиховог редоследа. У израчунавању ових кодова учествују следећи параметри везе и фрагмента:

1. Генерисани *MAC* кључ (на нивоу везе),
2. секвентни број,
3. апликативни протокол из структуре *TLSCompressed*: *TLSCompressed.type*,
4. верзија *TLS* протокола из структуре *TLSCompressed*: *TLSCompressed.version*,
5. дужина фрагмента из структуре *TLSCompressed*: *TLSCompressed.length* и

6. садржај фрагмента из структуре *TLSCompressed: TLSCompressed.fragment*.

MAC кодови се израчунавају пре шифровања података фрагмента. Секвенцијални шифарски алгоритми шифрују цео блок, укључујући и *MAC* код. Уколико секвенцијални алгоритам не користи векторе за синхронизацију (нпр. алгоритам *RC4*), завршно стање алгоритма при шифровању једног пакета представља иницијално стање при шифровању наредног. Код секвенцијалних алгоритама вредност параметра *length* код *TLSCiphertext* структуре једнака је збиру вредности параметара *TLSCompressed.length* и *SecurityParameters.mac_length*.

Блоковски шифарски алгоритми (нпр. *AES*, *3DES*) конвертују *TLSCompressed.fragment* структуру у блоковску *TLSCiphertext.fragment* структуру.

```
struct {
    opaque IV[SecurityParameters.record_iv_length];
    block-ciphered struct {
        opaque content[TLSCompressed.length];
        opaque MAC[SecurityParameters.mac_length];
        uint8 padding[GenericBlockCipher.padding_length];
        uint8 padding_length;
    };
} GenericBlockCipher;
```

Листинг 4. Декларације структуре *GenericBlockCipher*

Параметре структуре *GenericBlockCipher* чине.

1. *Initialization Vector, IV*

Подразумевано је (псеудо)случајно генерисање иницијализационог вектора у циљу немогућности његовог погађања од стране нападача. Верзије *TLS* протокола раније од верзије 1.1 нису имале подршку за иницијализациони вектор већ је у ту сврху коришћен претходно шифровани блок. Такав приступ, међутим, отвара простор за одређени тип напада (погађањем иницијализационог вектора) те је, почевши од верзије 1.1 протокола, овај параметар укључен. Код блоковских алгоритама дужина иницијализационог вектора је једнака дужини блока.

2. *padding*

Допуњавање садржаја до бајтовске дужине дељиве дужином блока блоковског шифарског алгоритма подразумевано је за рад блоковских шифарских алгоритама.

3. *padding_length*

Валидна дужина допуне садржаја пакета је од нула до 255 бајтова. На тај начин је ограничена употреба блоковских шифарских алгоритама на највећу дужину блока од 256 бајтова.

2.3.2.1.2. Протоколи *TLS*-а за успостављање везе

TLS систем заштите садржи три протокола који служе за успостављање безбедносних параметара везе (за потребе *TLS Record* протокола), за проверу аутентичности страна, и за извештавање о евентуалним грешкама. *TLS Handshake* протокол је задужен за успостављање сесије и успостављање њених следећих параметара:

- Идентификатор сесије - произвољна секвенца бајтова за једнозначно одређивање комуникационе сесије,
- сертификат учесника - *X509v3* дигитални сертификат учесника у комуникацији,
- алгоритам за компресовање - који ће се алгоритам за компресовање података користити пре функције шифровања,
- параметри шифарског алгоритма - функција за одређивање кључева, алгоритам за шифровање података, и алгоритам за израчунавање контролних MAC кодова,
- параметар *master secret* - низ бајтова дужине 48 бајтова, познат само учесницима у комуникацији и
- индикатор да ли се сесија може користити за успостављање више веза.

Наведени параметри сесије користе се за одређивање безбедносних параметара који ће, у оквиру *TLS Record* протокола, бити коришћени за запиту података који се преносе. У оквиру једне сесије, уколико је тако назначено, може се успоставити више веза.

Протокол *Change Cipher Spec* задужен је за измене процеса шифровања, у току комуникације. Овај протокол подржава само један тип поруке која се преноси у шифрованом облику (користећи тренутно актуелне параметре шифровања). Ову поруку може послати било која од страна, сервер или клијент, а њоме се прималац обавештава да ће наредни блокови бити шифровани по новим параметрима и кључевима за шифровање. Пошиљалац, одмах након слања поменуте поруке, мора захтеване параметре да примени у сопственом *TLS Record* протоколу, док је то задатак примаоца

одмах након пријема поруке. У оквиру успостављања сесије поруке *Change Cipher Spec* протокола шаљу се након успостављања основних безбедносних параметара, а пре слања поруке да је сесија успостављена.

Протокол *Alert* у оквиру *TLS* система заштите служи за обавештавање друге комуникационе стране о упозорењима и фаталним грешкама до којих је дошло. У случају да се овим протоколом пренесе порука о фаталној грешци, долази до тренутног раскида везе а сесија у којој је до такве грешке дошло не може се користити за успостављање нових веза. Поруке *Alert* протокола такође се компресују и шифрују а могу бити један од два типа:

- Поруке о прекиду везе - обавештавају другу страну у комуникацији да је веза прекинута. У случају да једна страна нема информацију да је веза раскинута могу се јавити одређени безбедносни пропусти. Раскид везе је симетричан, односно свака страна треба да пошаље другој страни обавештење да раскида везу, тј. да више неће слати податке у оквиру ње.
- Поруке о евидентираним грешкама - обавештавају другу страну да је дошло до грешке и, уколико је у питању фатална грешка а не само упозорење, да се веза раскида. Након раскида везе потребно је да обе стране уклоне из меморије све безбедносне параметре везане за њу. Типови порука о грешкама могу бити: *bad_record_mac*, *decryption_failed_RESERVED*, *record_overflow*, *decompression_failure*, *handshake_failure*, *no_certificate_RESERVED*, *bad_certificate*, *unsupported_certificate*, *certificate_revoked*, *certificate_expired*, *certificate_unknown*, *illegal_parameter*, *unknown_ca*, *access_denied*, *decode_error*, *decrypt_error*, *export_restriction_RESERVED*, *protocol_version*, *insufficient_security*, *internal_error*, *user_canceled*, *no_renegotiation*, *unsupported_extension*. Детаљан опис сваког од наведених типова порука може се наћи у званичном *RFC* документу *TLS* протокола.

TLS Handshake прокол је задужен за успостављање безбедносних параметара сесије, који ће бити коришћени у комуникацији - верзија протокола, шифарски алгоритам, аутентичност друге стране, *master secret* параметар. Успостављање наведених параметара обавља се у оквиру следећих корака:

1. Размена *hello* порука за одређивање алгоритама, (псеудо) случајних вредности, и

да ли сесија може бити настављена.

2. Размена криптографских параметара потребних за утврђивање *premaster secret* параметра.
3. Размена сертификата и криптографских информација потребних за проверу аутентичности комуникационих страна.
4. Генерисање *master secret* параметра на основу размењених *premaster secret* параметра и (псеудо)случајних вредности.
5. Достављање успостављених безбедносних параметара *TLS Record* протоколу.
6. Омогућавање потврђивања идентичности успостављених безбедносних параметара (у циљу успешности комуникације и спречавања напада путем посредовања нападача у успостављању сесије).

Значајна напомена везана за процес успостављања безбедне везе јесте та да постоје одређени типови напада којима нападач може навести комуникационе стране да не искористе алгоритме и параметре оптималне по питању нивоа заштите који пружају. Из тог разлога је потребно да се на апликативном нивоу одреди који је минимални ниво заштите потребан за успостављање комуникације, односно који алгоритми смеју бити коришћени за потребе безбедне размене података.

2.3.2.1.3. Управљање кључевима

TLS Record протокол захтева постојање алгоритма за генерисање кључева у зависности од тренутног стања успостављене везе, и у зависности од безбедносних параметара успостављених *TLS Handshake* протоколом. Успостављени *master secret* параметар користи се за одређивање следећих кључева:

1. *MAC* кључа клијента
2. *MAC* кључа сервера
3. кључа шифарског алгоритма клијента
4. кључа шифарског алгоритма сервера

Код неких алгоритама се на основу *master secret* параметра одређују и иницијализациони вектори на страни клијента и сервера.

2.3.2.1.4. Криптографска израчунавања

За потребе остваривање безбедне везе *TLS Record* протокол захтева утврђивање сета алгорита који ће се користити, *master secret* параметра, и (псеудо)случајних вредности на страни сервера и клијента. У сет алгоритама спадају алгорита за проверу аутентичности страна, шифарски алгорита, алгорита за компресовање података, и алгорита за израчунавање контролних *MAC* кодова. Све ове вредности, осим *master secret* параметра, размењују се у оквиру успостављања везе. Сам *master secret* параметар се израчунава на обе стране.

Код свих метода размене кључева користи се исти алгорита за израчунавање *master secret* параметра на основу *pre master secret* параметра. Безбедносна је препорука уклањање *pre master secret* параметра из радне меморије одмах након израчунавања *master secret* параметра. У израчунавању *master secret* параметра учествују следеће вредности:

1. *master secret* параметар
2. (псеудо)случајна вредност сервера
3. (псеудо)случајна вредност клијента

Када се користи *RSA* алгорита за проверу аутентичности и размену кључева, процес израчунавања *master secret* параметра подразумева следеће кораке:

1. Клијент генерише *pre master secret* параметар дужине 48 бајтова.
2. Параметар *pre master secret* шифрује се јавним кључем сервера и шаље му се.
3. Сервер дешифрује *pre master secret* параметар својим тајним кључем.
4. Обе стране израчунавају *master secret* параметар на основу *pre master secret* параметра.

Када се користи *Diffie-Hellman* протокол за размену кључева, уговорени кључ се користи као *pre master secret* параметар. Почетни бајтови уговореног кључа, који садрже само нуле, одбацују се пре употребе. Параметре *Diffie-Hellman* протокола одређује сервер. Превођење *pre master secret* параметра у *master secret* параметар одвија се коришћењем стандардне процедуре.

2.3.2.1.5. Имплементације *TLS* протокола

За примену сопствених алгоритама у *TLS* систему заштите комуникације, као један од најједноставнијих начина њихове интеграције, идентификована је њихова уградња у неку од имплементација (библиотека) са отвореним изворним кодом. Анализиране су имплементације које имају подршку за Линукс оперативни систем као циљни систем за имплементацију сопственог решења. У најпопуларније имплементације које испуњавају постављене услове спадају *GnuTLS* [18], *OpenSSL* [19] и *NSS* [20]. У разматрање је још укључена и библиотека *yaSSL* [21].

Једна од најзначајнијих карактеристика *GnuTLS* имплементације јесте њена доступност под *LGPLv2.1+* лиценцом. Имплементација се у виду библиотеке развија под окриљем фондације за слободан софтвер (*Free Software Foundation*). У безбедносне протоколе подржане овом имплементацијом спадају: *SSL 3.0*, *SSL 3.0*, *TLS 1.0*, *TLS 1.1*, и *TLS 1.2*. Верзија 2.0 *SSL* протокола није подржана услед идентификованих безбедносних пропуста. Ова имплементација се извршава на већини оперативних система из *UNIX* породице, као и на *MS Windows* оперативним системима.

OpenSSL имплементација, развијена у оквиру *OpenSSL* пројекта, заснована је на *SSLeay* библиотеци коју су развили Ерик Јанг и Тим Хадсон, експерти за безбедност и оснивачи фирме *Cryptosft*. *OpenSSL* имплементација је јавно доступна под сопственом лиценцом, базираној на *Apache* лиценци, с тим да важе и сва ограничења из лиценце под којом је доступна *SSLeay* библиотека.

NSS (Network Security Services) сет библиотека дизајниран је са циљем да омогући коришћење *SSL* и *TLS* протокола, *PKCS#5*, *PKCS#7*, *PKCS#11*, *PKCS#12*, *S/MIME*, *X.509v3* сертификата, и других безбедносних стандарда у развоју мрежних апликација. Извори овог сета библиотека налазе се у библиотекама које је компанија *Netscape* развила у току развоја самог *SSL* протокола. Допринос у развоју овог сета библиотека дале су и компаније *AOL*, *Red Hat*, *Sun Microsystems*, *Google* и друге, а тренутно се репозиторијум са изворним кодом налази на серверима компаније *Mozilla*. Једна од значајних карактеристика овог сета библиотека јесте и подршка хардверских шифарских компонената кроз *PKCS#11* интерфејс. На овај начин је омогућено значајно подизање перформанси и употреба кључева који се налазе на *Smart* картицама и другим хардверским компонентама. У софтверски подржане алгоритме овог сета библиотека

спадају *RSA*, *DSA*, *ECDSA*, *Diffie-Hellman*, *EC Diffie-Hellman*, *AES*, *Triple DES*, *DES*, *RC2*, *RC4*, *SHA-1*, *SHA-256*, *SHA-384*, *SHA-512*, *MD2*, *MD5*, *HMAC* и други популарни алгоритми.

Имплементација *yaSSL* развијена је у виду клијентских и серверских библиотека. Имплементација је доступна под *GPL* лиценцом, као и под комерцијалном лиценцом. Најпопуларнији пројекат који користи ову имплементацију *SSL/TLS* протокола је *MySQL* систем за управљање базама података.

У наредним табелама приказане су подршке анализираних имплементација за различите безбедносне системе, протоколе за размену кључева, шифарске алгоритме, и алгоритме за компресовање података. Подршке се односе на верзије имплементација актуелне у тренутку припреме овог материјала (*GnuTLS* 2.8, *OpenSSL* 0.9.8, *NSS* 3.12.4, *yaSSL* 1.9.6). Подршка за верзију 2 *SSL* протокола се не сматра предношћу, чак се у неким разматрањима узима и као мана, због присутних идентификованих безбедносних пропуста у тој верзији.

	<i>SSL 2.0</i>	<i>SSL 3.0</i>	<i>TLS 1.0</i>	<i>TLS 1.1</i>	<i>TLS 1.2</i>
<i>GnuTLS</i>	не	да	да	да	да
<i>OpenSSL</i>	да	да	да	не	не
<i>NSS</i>	да	да	да	не	не
<i>yaSSL</i>	не	да	да	да	да

Табела 8. Подршка за системе заштите у имплементацијама

	<i>Anon RSA</i>	<i>RSA</i>	<i>RSA export</i>	<i>DHE RSA</i>	<i>DHE DSS</i>	<i>SRP DSS</i>	<i>SRP RSA</i>	<i>SRP</i>	<i>PSK</i>	<i>DHE PSK</i>	<i>ECC</i>
<i>GnuTLS</i>	да	да	да	да	да	да	да	да	да	да	не
<i>OpenSSL</i>	да	да	да	да	да	не	не	не	не	не	да
<i>NSS</i>	да	да	да	да	да	не	не	не	не	не	да
<i>yaSSL</i>	да	да	да	да	да	не	не	не	да	да	не

Табела 9. Подршка за алгоритме за размену кључева

	<i>AES 256 CBC</i>	<i>AES 128 CBC</i>	<i>3DES CBC</i>	<i>DES CBC</i>	<i>RC4 128 CBC</i>	<i>RC4 40 CBC</i>	<i>RC2 40 CBC</i>	<i>Carmelia</i>
<i>GnuTLS</i>	да	да	да	да	да	да	да	да
<i>OpenSSL</i>	да	да	да	да	да	?	?	да

<i>NSS</i>	да	да	да	да	да	?	?	да
<i>yaSSL</i>	да	да	да	да	да	да	да	не

Табела 10. Подршка за шифарске алгоритме

	<i>ZLIB</i>	<i>LZO</i>
<i>GnuTLS</i>	да	да
<i>OpenSSL</i>	да	не
<i>NSS</i>	не	не
<i>yaSSL</i>	да	не

Табела 11. Подршка за алгоритме за компресовање података

	<i>OpenPGP</i>	<i>SRP</i>	<i>PSK</i>	<i>TLS/IA</i>	<i>Supp. data</i>	<i>RFC 5077</i>	<i>RFC 5705</i>
<i>GnuTLS</i>	да	да	да	да	да	да	да
<i>OpenSSL</i>	не	не	не	не	не	да	да
<i>NSS</i>	не	не	не	не	не	да	не
<i>yaSSL</i>	не	не	да	не	не	да	не

Табела 12. Подршка за екстензије

2.3.2.1.6. Интеграција сопственог шифарског алгоритма

Из дате анализе *TLS* протокола за заштиту података на транспортном слоју може се закључити да овај протокол нуди, за одређене примене, довољан али не и превише широк простор за уграђивање сопствених безбедносних решења. Постојање развијених имплементација, у виду наведених библиотека са отвореним изворним кодом, елиминише потребу за развојем решења од почетка, и омогућује интервенцију само на компонентама од критичне важности за безбедност сопствено развијеног система. На пример, коришћењем готове имплементације може се извршити уградња само сопственог шифарског алгоритма, док се за потребе размене кључева могу искористити већ постојећи алгоритми.

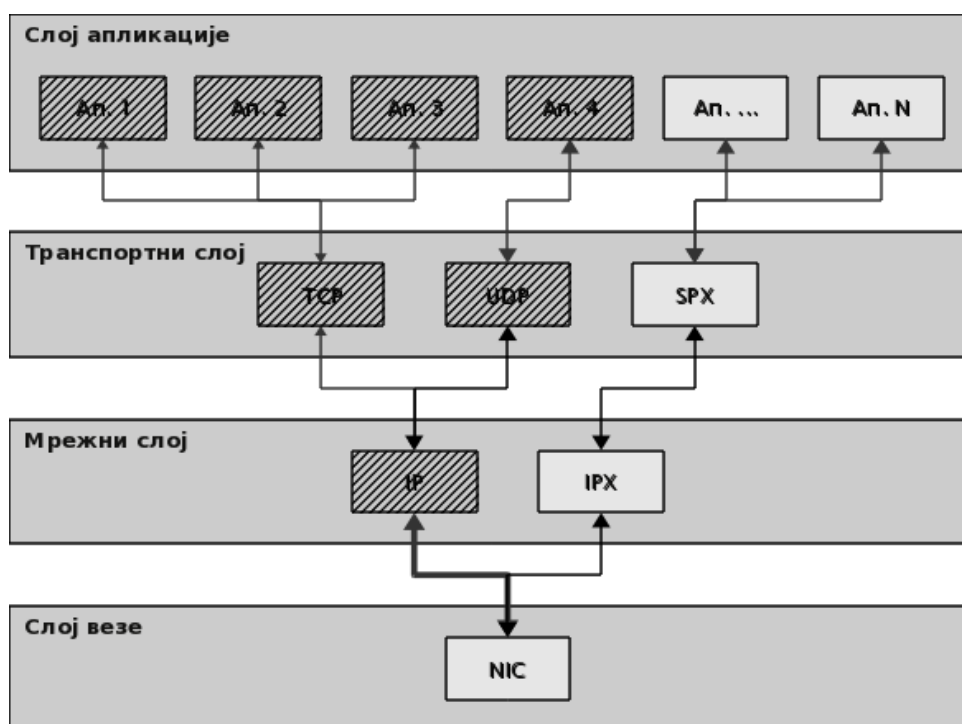
Сопствени шифарски алгоритам се у *TLS* систем заштите може интегрисати као нови алгоритам, или као измена неког од постојећих стандардних алгоритама (*AES*, *RC4* и сл.). Коришћење неког од већ интегрисаних алгоритама може поједноставити процес

интеграције сопственог алгоритма а изводљивост оваквог приступа зависи првенствено од специфичности сопственог алгоритма, односно од компатибилности сопственог алгоритма са постављеним принципима и ограничењима постојећег алгоритма који би се користио као основа.

Два основна недостатка *TLS* система заштите јесу одређена безбедносна ограничења и потреба да апликативни слој мора експлицитно захтевати коришћење овог система. Поменути безбедносна ограничења односе се на могуће сценарије у којима нападач може утицати на комуникационе стране да не искористе оптималне (доступне) системе заштите. Са друге стране, потреба за експлицитним захтевом за коришћење *TLS* протокола умањује једноставност његовог коришћења, и оставља могућност да крајњи корисници ненамерно остварују небезбедне везе и њима прослеђују поверљиве податке.

2.3.3. Заштита на нивоу мрежног протокола

Један од главних аргумената тезе која се обрађује у овом раду базира се на неким од аспеката решења развијаног у претходних неколико година. То решење је засновано на *IPsec* систему заштите података на *IP* протоколу мрежног слоја *OSI* референтног модела.



Слика 11. Заштита на нивоу мрежног протокола

Полазна тачка рада била је искоршћавање већ развијеног безбедносног протокола за

имплементацију сопственог шифарског алгоритма са циљем развоја сопственог решења базираног на отвореним компонентама. Такође, следећи циљ решења била је и независност од корисника решења и корисничких мрежних апликација, као и независност од постојеће телекомуникационе инфраструктуре. У складу са тим, мрежни слој *OSI* референтног модела одабран је као најпогоднији за реализацију поменутих захтева.

Због потребе за коришћењем отворених готових компонената развој и различите провере решења вршени су на Линукс оперативном систему и *IPsec* имплементацији у његовом језгру. У првим фазама развоја вршени су експерименти и са осталим *IPsec* имплементацијама на Линукс оперативном систему (*FreeS/WAN*, *Openswan* и *strongSwan*) чије ће карактеристике бити описане у овом делу рада.

Неке од резултата добијених у току бављења овом облашћу документовани су у следећим радовима на домаћим и страним научним конференцијама и часописима:

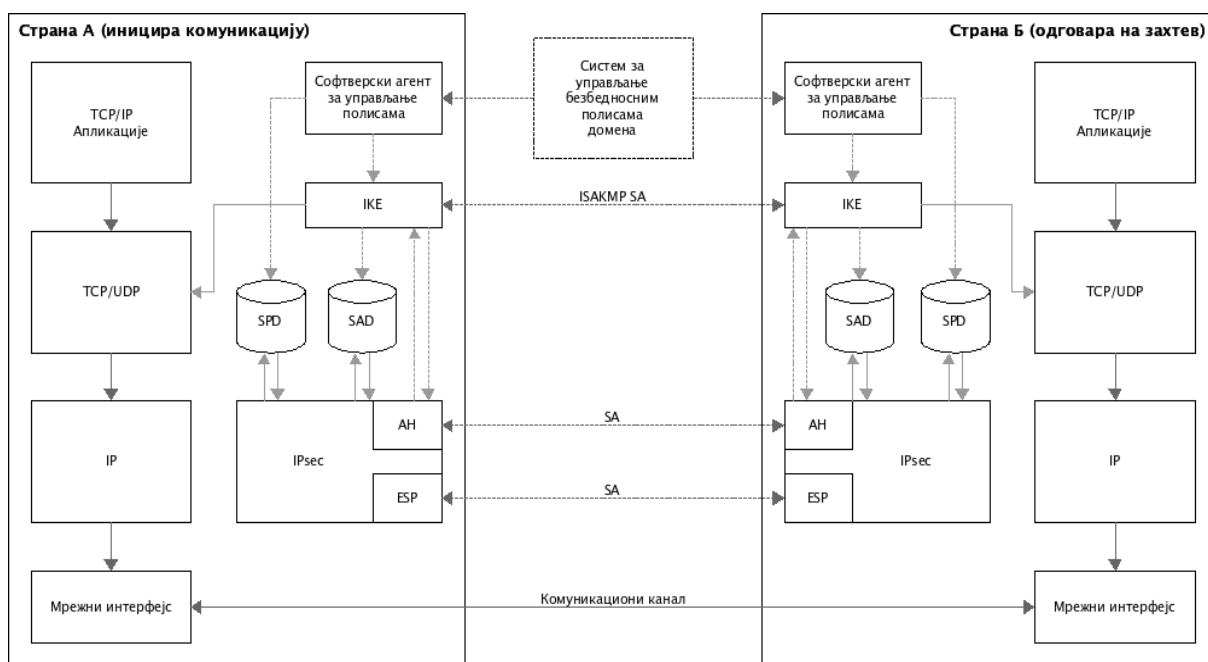
1. Јевремовић А., Веиноввић М., „Заштита података на *IP* нивоу под Линукс ОС“, Зборник радова 50. конференције Етран, Београд, 2005., стр. 114-117;
2. Јевремовић А., Веиноввић М., „*IPsec* - анализа утицаја алгоритма за шифровање на саобраћај у *LAN* мрежама“, 14. телекомуникациони форум Телфор 2006., Београд, 2006., CD издање;
3. Веиноввић М., Јевремовић А., Шимић Г., „Имплементација сопственог алгоритма заштите у *IPsec*-у под Линукс оперативним системом“, Сингидунум ревија, Vol 5 No 1, Београд, 2008., стр. 92-102;
4. Veinović M., Jevremović A., Šimić G., „*Analysis and implementation of custom cipher algorithm for IPsec under Linux OS*“, International Journal of Computer Science and Network Security, Vol. 8 No. 7, July 2008;

Такође, резултати остварени у овом окружењу послужили су нам као основа за моделовање система сродне архитектуре али на *HTTP* протоколу апликативног слоја *OSI* референтног модела. Тај систем је документован у раду:

1. Јевремовић А., Веиноввић М., Шимић Г., „Примена сопственог шифарског алгоритма за заштиту *AJAX* позива код *Web* апликација“, Зборник радова 52. конференције за електронику, телекомуникације, рачунарство, аутоматику и нуклеарну технику - Етран, Београд, 2008., CD издање;

2.3.3.1. *Internet Protocol Security* систем заштите

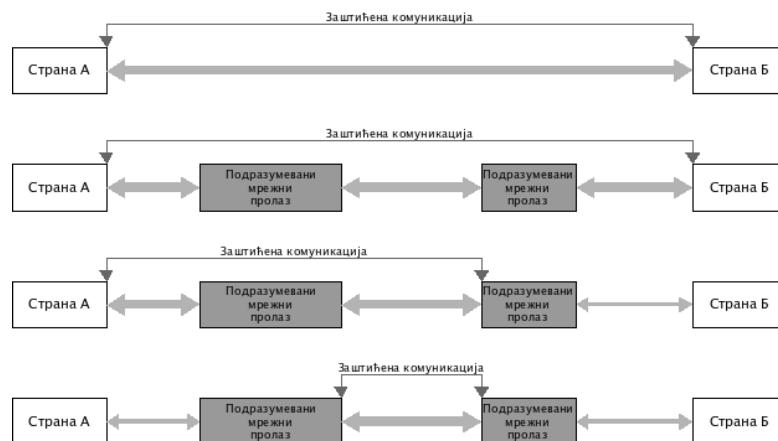
Internet Protocol Security (IPsec) протокол је решење за заштиту података који се преносе коришћењем *IP* протокола. Овај систем је реализован сетом протокола којима се остварује шифровање, провера аутентичности извора и интегритета порука које се преносе. Додатно, *IPsec* систем користи и протокол за размену кључева на почетку сесије.



Слика 12. Архитектура комуникације заштићене *IPsec* системом заштите

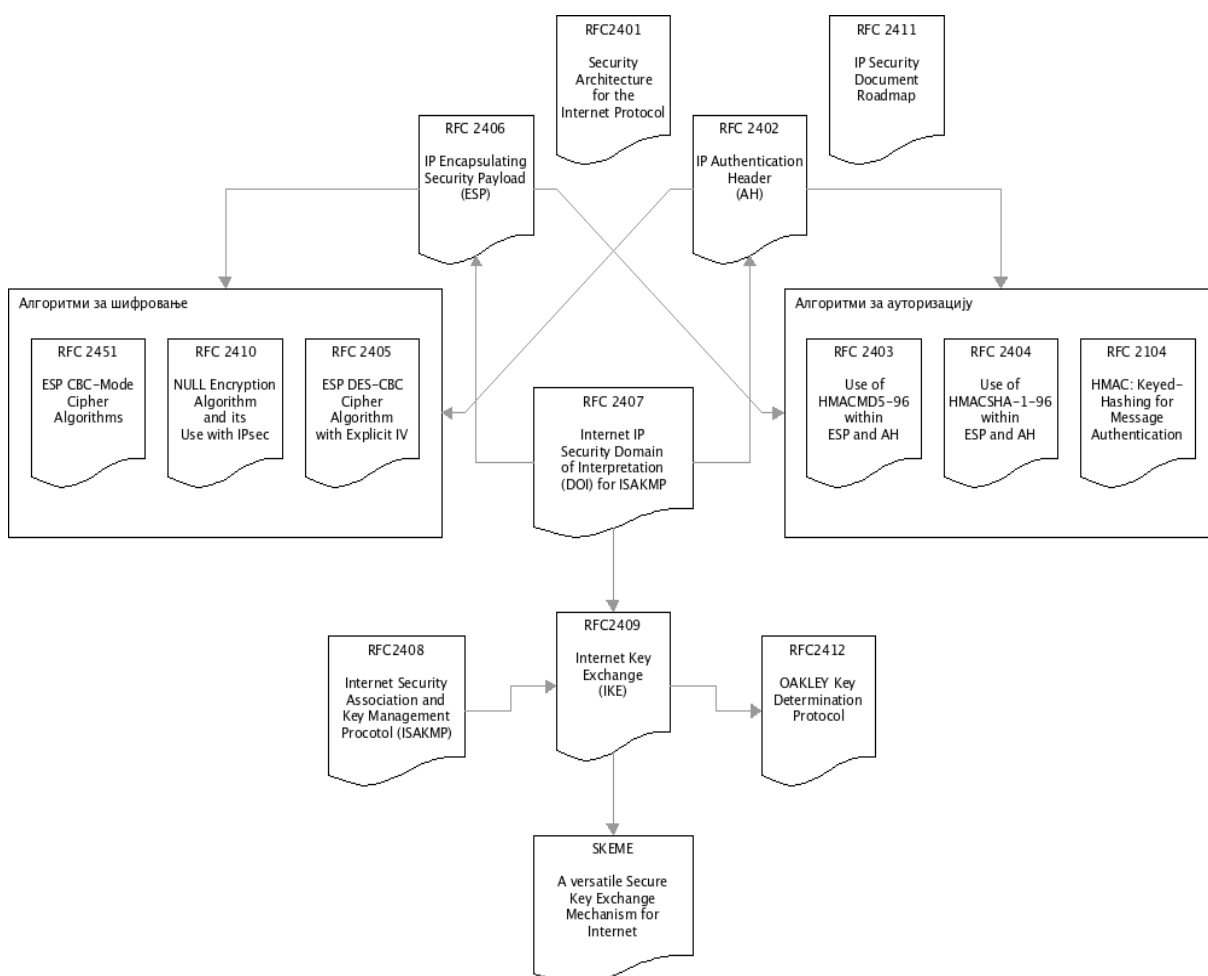
IPsec систем се може користити за заштиту комуникације између крајњих учесника, између крајњег учесника, са једне стране, и подразумеваног мрежног пролаза, са друге стране, или између подразумеваних мрежних пролаза са обе стране. У складу са тим *IPsec* подржава два режима рада: транспортни и тунелски.

За развој *IPsec* система заштите званично је задужена *IETF* група. Сам *IPsec* систем заштите је наследник *ISO* стандардизованог *NLSP (Network Layer Security Protocol)* протокола базираног на *SP3* протоколу, којег је први објавио амерички национални институт за стандарде и технологију (*NIST*) а иницијално је дизајниран од стране америчке националне безбедносне агенције (*NSA*). Данас постоји велики број докумената којим је *IPsec* систем заштите документован и стандардизован.



Слика 13. Различити модели заштите коришћењем *IPsec* система

Једна од значајних предности *IPsec* система заштите у односу на друге популарне системе заштите (*SSL*, *TLS*...) је рад на ниском нивоу у односу на *OSI* и *TCP/IP* референтне моделе. Оваква позиција обезбеђује флексибилност у погледу заштите великог броја протокола виших нивоа.



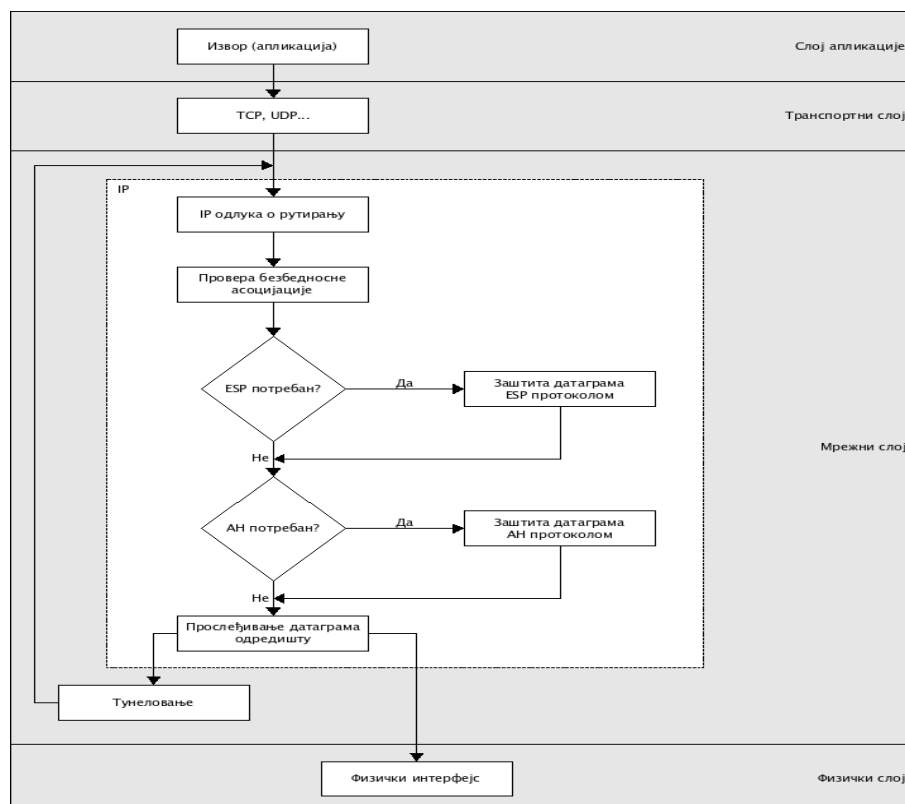
Слика 14. Однос *RFC* докумената значајних за *IPsec*

Основне компоненте *IPsec* система заштите су:

1. *Internet Key Exchange (IKE, IKE2)* – систем за управљање кључевима и успостављање безбедносних асоцијација.
2. *Authentication Header (AH)* – протокол за обезбеђивање аутентичности и интегритета.
3. *Encapsulating Security Payload (ESP)* – протокол за обезбеђивање поверљивости, аутентичности и интегритета.

Још једна значајна карактеристика *IPsec* система заштите је и подршка за *IPv6* протокол. У ствари, *IPsec* је подразумевани део овог протокола и иницијално је и развијан на њему а накнадно је портован на верзију 4.

За успешно коришћење *IPsec* протокола неопходно је успостављање правилних безбедносних асоцијација. Безбедносне асоцијације представљају правила по којима ће се успоставити безбедна комуникација између учесника. У безбедносним асоцијацијама могу бити директно постављени кључеви који ће се користити за шифровање а у ту сврху се може искористити и протокол за размену кључева путем Интернета.

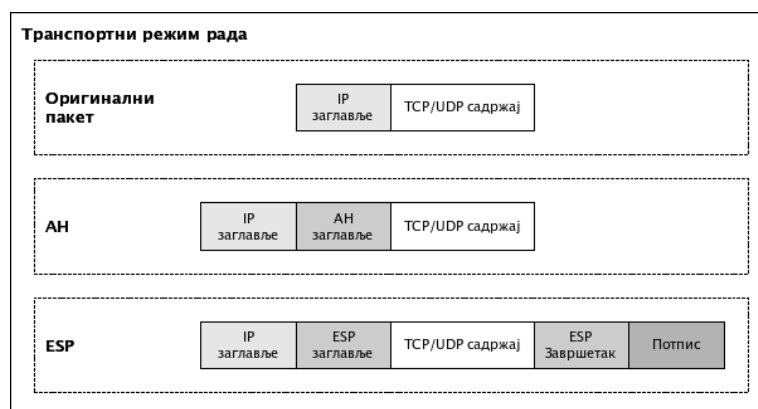


Слика 15. Процес слања података са коришћењем *IPsec* система заштите

IPsec систем заштите се може посматрати као једноставно и ефикасно решење за заштиту комуникација базираних на *IP* протоколу. Важна карактеристика овог система јесте отвореност и документованост кроз *RFC* и друге документе. Избор *IPsec* система заштите као носиоца реализације сопственог решења базираног на сопственом алгоритму, резултат је дугортајног и темељног испитивања постојећих алтернатива.

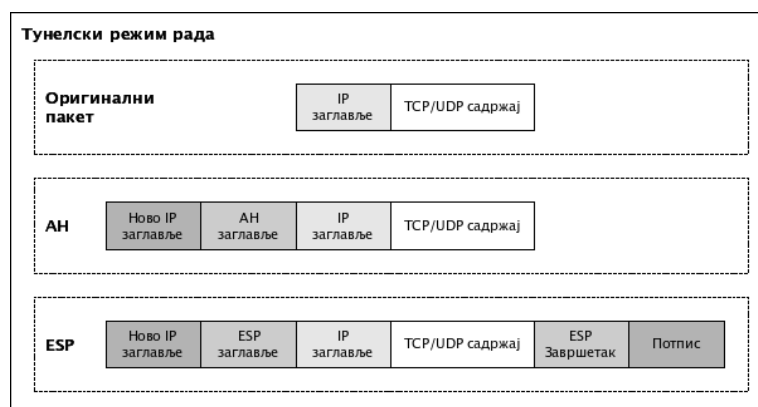
2.3.3.1.1. Режи́ми рада

Једна од значајних карактеристика *IPsec* система заштите јесте подршка за два основна режима рада: транспортни и тунелски. Основна разлика између ова два режима рада лежи у томе да ли се шифрују само подаци протокола вишег нивоа или комплетан *IP* датаграм.



Слика 16. Транспортни режим рада

У случају тунелског режима рада за сваки *IP* датаграм креира се нови *IP* датаграм који је сачињен од новог заглавља и података у виду шифрата оригиналног *IP* датаграма. Основна потреба за тунелским режимом рада произилази из потребе за превођењем мрежних адреса (енгл. *Network Address Translation*) и креирањем виртуалних приватних мрежа (енгл. *Virtual Private Network*).



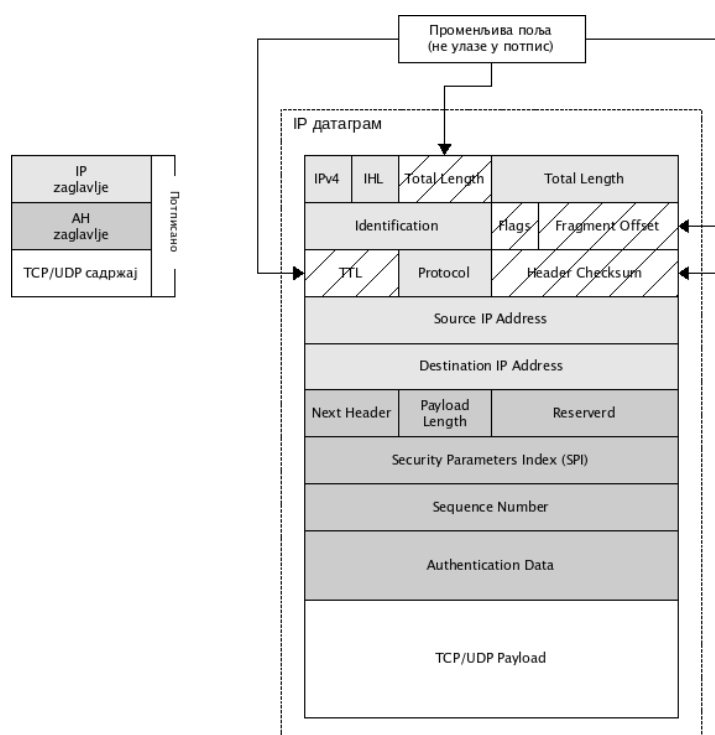
Слика 17. Тунелски режим рада

За потребе интеграције сопствених криптолошких система у *IPsec*, разлике између описаних режима рада могу се занемарити, пре свега из разлога што, гледано са аспекта интеграције сопственог шифарског алгоритма, исти принципи важе за оба режима рада а цео процес промене режима рада обавља се аутоматски у оквиру имплементације.

2.3.3.1.2. Протокол *Authentication Header*

AH (*Authentication Header*) заглавље се у оквиру *IPsec* протокола користи за обезбеђивање интегритета података и аутентичности извора *IP* датаграма [RFC 2402]. Додатно, овај протокол омогућава и заштиту од поновљених слања истих датаграма. *AH* протоколом су покривени сви статични делови *IP* датаграма – подаци протокола вишег нивоа и непроменљива поља заглавља.

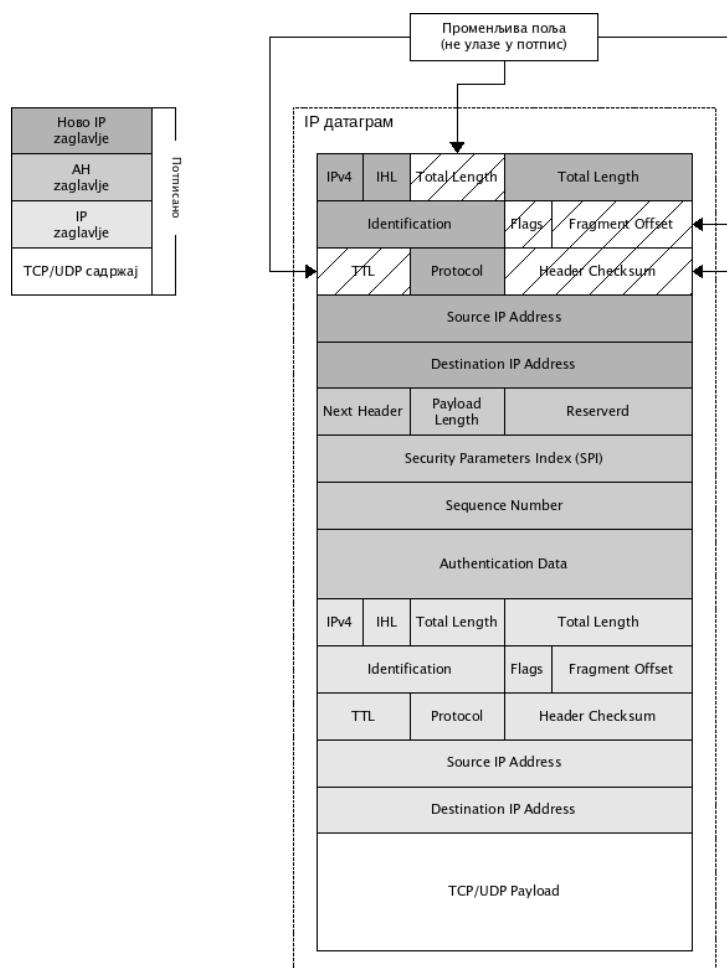
AH протокол се може користити независно или у комбинацији са *ESP* (*Encapsulating Security Payload*) протоколом. Такође, овај протокол се може користити у комуникацији између крајњих учесника у комуникацији, између мрежних пролаза или између крајњег учесника, са једне стране, и подразумеваног пролаза [22], са друге стране.



Слика 18. Структура *IP* датаграма са *AH* протоколом у транспортном режиму рада

Разлика између *AH* и *ESP* протокола на пољу обезбеђивања интегритета података и аутентичности извора огледа се у томе што *ESP* протокол подржава само рад са

подацима *IP* датаграма, односно не подржава рад са пољима заглавља (осим у случају тунелског режима рада). Структура *IP* датаграма са коришћеним *AH* протоколом разликује се у зависности од тога да ли се *IPsec* протокол користи у транспортном или тунелском режиму рада.



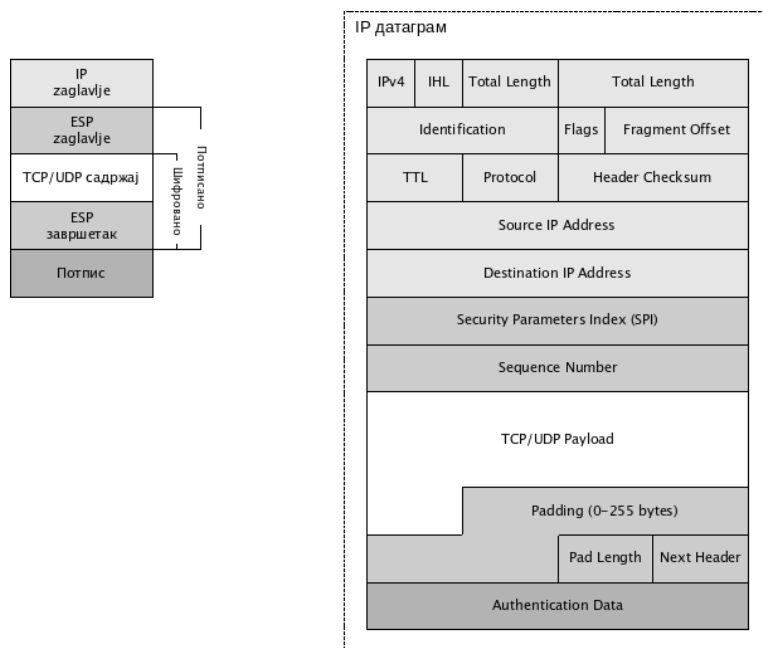
Слика 19. Структура *IP* датаграма са *AH* протоколом у тунелском режиму рада

Заглавље *AH* протокола има дужину од 24 бајта. Први бајт је поље *Next Header*. Ово поље дефинише протокол наредног заглавља. С обзиром на то да под поља заглавља која *AH* протокол укључује подразумевају и изворишна и одредишна *IP* адреса, *AH* протокол се не може користити у основном режиму рада код *IP* датаграма над којима се врши *NAT* трансформација. Код тунелског режима рада комплетан *IP* датаграм се енкапсулира а вредност поља *Next Header* се поставља на 4. Код транспортног режима рада вредност поља *Next Header* се поставља на вредност 6. Осим тунелског режима рада *IPsec* протокол поседује и екстензију *NAT-Traversal* која омогућава примену *NAT* трансформација у транспортном режиму рада.

Поље *SPI* (*Security Parameters Index*) има дужину од 32 бита. У овом пољу се налази безбедносна асоцијација која се користи за декапсулацију пакета. Поље *Sequence Number* користи се за заштиту од тзв *reply* напада. Поље *Authentication Data* садржи *HMAC* (*hash message authentication data*) дужине 96 бита. Коришћењем садржаја овог поља пријемна страна може (уз познавање тајног кључа) утврдити да ли је дошло до измена на *IP* датаграму у току преноса.

2.3.3.1.3. Протокол *Encapsulating Security Payload*

ESP (*Encapsulating Security Payload*) је кључни протокол *IPsec* архитектуре и дизајниран је да обезбеди више безбедносних сервиса [RFC 2406]. Овај протокол се може користити независно или у комбинацији са *AH* (*Authentication Header*) протоколом. Такође, овај протокол се може користити у комуникацији између крајњих учесника, између мрежних пролаза или између крајњег учесника, са једне стране, и подразумеваног пролаза, са друге стране [23].



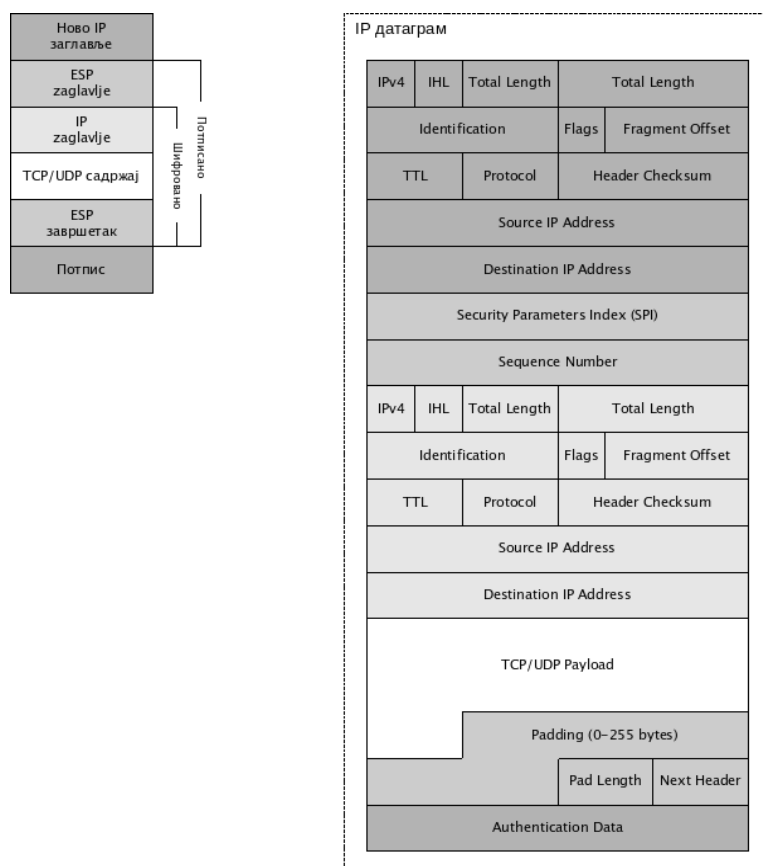
Слика 20. Структура *IP* датаграма са *ESP* протоколом у транспортном режиму рада

Сервиси које ће *ESP* протокол обезбедити одређују се код подешавања безбедносне асоцијације за комуникацију. Основни задатак *ESP* протокола је да обезбеди поверљивост података који се преносе коришћењем *IP* и *IPsec* протокола. Додатно, *ESP* обезбеђује и проверу аутентичности извора, интегритет порука (датаграма, не и целокупног садржаја комуникације) а нуди и заштиту од *reply* напада.

ESP протокол се може користити и за обезбеђивање појединачних сервиса. На пример, могуће је искористити само сервис поверљивости без коришћења сервиса интегритета и провере аутентичности извора (било путем *ESP* или *AH* протокола). Међутим, оваква конфигурација коришћених сервиса може довести до угрожавања сервиса поверљивости [24].

Сервиси провере аутентичности и интегритета порука се често заједно стављају под сервис аутентичности. Без укључивања ових сервиса није могуће искористити *ESP* протокол за заштиту од *reply* напада.

ESP протокол се може користити и са *IPv4*, и са *IPv6* протоколом. *ESP* заглавље се умеће на крају заглавља *IP* датаграма у транспортном режиму рада, односно пре енкапсулираног заглавља оригиналног *IP* датаграма у тунелском режиму рада.



Слика 21. Структура *IP* датаграма са *ESP* протоколом у тунелском режиму рада

SPI (*Security Parameters Index*) поље заглавља има дужину од 32 бита и користи се у комбинацији са *ESP* протоколом и одредишном *IP* адресом за јединствено одређивање безбедносне асоцијације датаграма. Вредности од 1 до 255 су резервисане за будућу

употребу. SPI поље заглавља је обавезно код коришћења *ESP* протокола. Вредност 0 је резервисана за локалну употребу и на мрежи се не смеју појавити датаграми који је садрже.

Поље *Sequence Number* је, такође, дужине 32 бита и повећава се са сваким послатим датаграмом. Ово поље је обавезно чак и у случајевима када се не подразумева заштита од *reply* напада. Вредност овог поља се поставља на 0 на почетку комуникације.

Најважније поље датаграма код коришћења *ESP* протокола је поље *Payload*, односно сам садржај датаграма који се штити. Осим корисничких података у овом пољу се могу наћи и подаци везани за коришћени алгоритам за шифровање (нпр. иницијализациони вектор и сл.).

2.3.3.1.4. Имплементације *IPsec* протокола

С обзиром на то да *IPsec* представља *de facto* стандард за обезбеђивање комуникација заснованих на *IP* протоколу, постоји велики број имплементација за различите оперативне системе, уређаје, протоколе и сл.

Са аспекта оперативних система често се користи принцип интегрисања имплементације у језгро оперативног система а коришћење конфигурационих алата (*ISAKMP/IKE*) из корисничког простора. Осим овога приступа постоје и примери коришћења *IPsec* имплементација независних од језгра оперативних система.

Линукс оперативни систем поседује сопствену имплементацију *IPsec* протокола у оквиру језгра почев од верзије 2.6. Пре ове имплементације на Линукс оперативном систему су најчешће коришћене *FreeS/WAN*, *Openswan* и *strongSwan* имплементације које су функционисале у виду процеса одвојених од језгра оперативног система.

Компанија Мајкрософт је почела са интеграцијом сопствене *IPsec* имплементације у оперативне системе почев од 2000. године. Данас, интегрисану подршку за овај систем заштите имају оперативни системи *Windows Vista*, *Windows Server 2008*, *Windows Server 2003*, *Windows XP*, и *Windows 2000*. Такође, поменути *IPsec* имплементација је интегрисана и у *Active Directory* сервис.

На *BSD* групи оперативних система (*FreeBSD*, *NetBSD*, *OpenBSD*) углавном се користи *KAME* имплементација. Иста имплементација се користи и на *Mac OS X* оперативном

систему. *OpenBSD* системи користе и *NRL (United States Naval Research Laboratory)* имплементацију, односно, сопствену имплементацију базирану на коду поменуте имплементације.

Производи компаније *Cisco* користе сопствену имплементацију у *IOS* оперативном систему. Оперативни системи компаније *IBM (AIX, z/OS)* користе сопствене имплементације. *Solaris*, оперативни систем компаније *Sun*, користи сопствену имплементацију.

2.3.3.1.4.1.Пројекат KAME

KAME пројекат је представљао заједнички напор шест јапанских организација који је имао за циљ да обезбеди слободну имплементацију IPsec протокола за различите варијанте BSD Unix оперативног система [25]. Пројекат је започет 1998. године а 7. новембра 2005. године је званично објављено да ће пројекат бити завршен марта 2006. године. Свој назив пројекат је добио по граду у коме су се налазиле канцеларије пројекта, *Karigome*.

Следеће организације су учествовале у пројекту:

1. *ALAXALA Networks Corporation*
2. *Fujitsu, Ltd.*
3. *Hitachi, Ltd.*
4. *Internet Initiative Japan Inc.*
5. *Keio University*
6. *NEC Corporation*
7. *University of Tokyo*
8. *Toshiba Corporation*
9. *Yokogawa Electric Corporation*

DragonFly BSD, FreeBSD, и NetBSD оперативни системи интегрисали су код за IPsec и IPv6 из KAME пројекта. OpenBSD оперативни систем садржи само код за подршку IPv6 протокола.

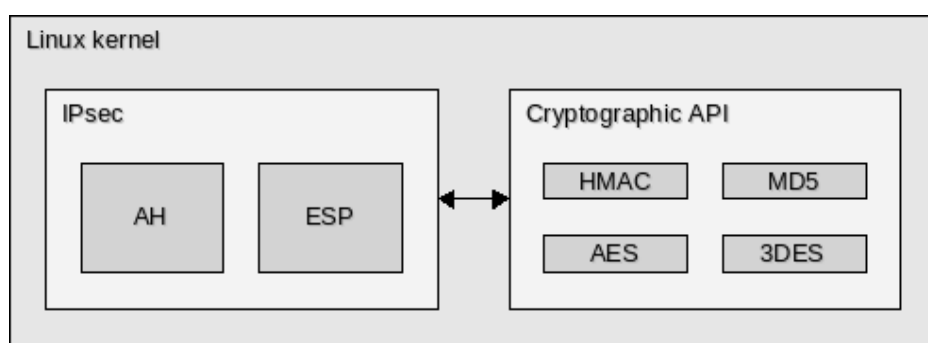
Један од значајних резултата остварених овим пројектом је портовање различитих корисничких алата на Линукс платформу у оквиру *IPsec-Tools* пројекта [26]. У оквиру

ових алата се налазе:

1. **libipsec** - Библиотека са PF_KEY имплементацијом.
2. **setkey** - Алат за управљање базом безбедносних полиса (*Security Policy Database, SPD*) и базом безбедносних асоцијација (*Security Association Database, SAD*).
3. **racoон** – програм за аутоматско размењивање кључева за везе.
4. **racoонctl** – програм за контролу *racoон*-а.

2.3.3.1.4.2.Имплементација у језгру Линукс оперативног система

Почев од верзије 2.5.47 на Линуксу је могуће користити IPsec имплементацију интегрисану у језгро оперативног система. Аутори ове имплементације, Алексеј Кузнецов и Дејвид С. Милер, свој рад су започели из основа, а од краја 2002. године се *IPsec* налази као саставни део кернела Линукс оперативног система верзије 2.6.



Слика 22. Организација компонената значајних за IPsec

Решење везано за безбедну комуникацију представљено у овом делу рада базирано је на *IPsec* имплементацији у језгру Линукс оперативног система. Неки од најважнијих разлога за давање предности овој имплементацији за интеграцију сопственог шифарског алгоритма су:

1. Доступност изворног кода имплементације,
2. перформансе (решење на нивоу језгра оперативног система) и
3. лиценца (*GPL*).

Додатно, ова имплементација користи хеш и шифарске алгоритме који су такође интегрисани у језгро оперативног система и чији је изворни код такође доступан (слика 22).

Сам Линукс оперативни систем је популаран као веома стабилан оперативни систем (постоје примери сервера који су под пуним оптерећењем радили и по више година без ресетовања) са добрим мрежним перформансама. Још једна значајна карактеристика Линукс оперативног система, односно његовог језгра, јесте и та да су изворни кодови свих компонената доступни јавности.

Језгро Линукс оперативног система садржи и друге значајне мрежне компоненте које се могу користити у комбинацији са решењем везаним за заштићену комуникацију. На пример, *netfilter* компонента језгра омогућава филтрирање пакета уз коришћење *iptables* корисничког алата.

У поређењу са осталим, раније описаним, имплементацијама *IPsec* безбедносног решења, имплементација у језгру Линукс оперативног система нуди могућност лакше интеграције сопственог шифарског алгоритма уз могућност постизања максималних перформанси.

Имплементација је, као и остатак језгра оперативног система, реализована у програмском језику *C*. Програмски језик *C*, поред добрих перформанси, нуди и могућност директног приступа хардверским ресурсима. Ова карактеристика решења може бити од значаја код будућих проширивања решења, пре свега повезивања са хардверским реализацијама шифарског алгоритма.

2.3.3.1.5. *Internet Key Exchange* протокол

Протокол *IKE* (енгл. *Internet Key Exchange*) је компонента *IPsec* протокола која се користи за међусобну проверу аутентичности учесника у комуникацији и за успостављање и управљање заштићеним везама (безбедносним асоцијацијама, енгл. *Security Associations, SA*). Тренутно актуелна верзија овог протокола је 2 (*IKEv2*).

Успостављање заштићених веза једна је од основних функција *IKE* протокола. Кроз успостављање заштићене везе дефинишу се безбедносни параметри неопходни за рад *ESP* и *AH* протокола, односно, шифарских алгоритама који обезбеђују комуникацију.

Све комуникације *IKE* протокола подразумевају две поруке - захтев и одговор. Овај пар порука назива се разменом (енгл. *exchange*). За обезбеђивање поузданости комуникације задужена је страна која шаље захтев - уколико се одговор не добије у одређеном временском периоду страна која шаље захтев дужна је да захтев понови или прекине

везу. Први пар порука *IKE* протокола при успостављању везе (*IKE_SA_INIT*) служи за утврђивање параметара заштићене везе, временску синхронизацију и размену *Diffie-Hellman* вредности. Други пар порука (*IKE_AUTH*) размењује идентитете учесника, потврђује познавање дељене тајне и успоставља заштићену везу.

IKE протокол подразумевано ради на порту 500 или 4500. За размену порука користе се датаграми *UDP* протокола транспортног нивоа. С обзиром да овај протокол не нуди поуздан пренос података сам *IKE* протокол садржи функције за препознавање и решавање губитка пакета, вишеструку испоруку истог пакета и фалсификовање пакета.

Поруке *IKE* протокола су пројектоване да буду кратке. Међутим, горња граница порука није експлицитно ограничена (осим спецификацијом да имплементације морају да подрже дужину од 1280 бајтова а препоручено је да имају могућност обраде порука дужине до 3000 бајтова, или већих). Сам *IKE* протокол не поседује могућност фрагментовања порука.

При успостављању безбедне везе *IKE* протокол поред сета безбедносних протокола који ће бити коришћени (*ESP*, *AH*) утврђује и који ће шифарски алгоритми бити коришћени за те протоколе. Иницијално се листа подржаних шифарских алгоритама (трансформација) размењује у виду предлога. За сваки од алгоритама могу се јавити атрибути уколико сам назив алгорита није довољан за једнозначно одређивање жељене трансформације (нпр. дужина кључа код алгоритама који подржавају различите дужине).

Шифарски кључ који се уговори *IKE* протоколом може се користити ограничен период времена, за шифровање ограничене количине података. Ова ограничења уједно представљају и животни век заштићене везе за коју је кључ искоришћен. Уколико се достигне било које од поменута два ограничења заштићена веза се више не сме користити, а уколико постоји потреба за даљим преносом података мора се успоставити нова заштићена веза. За успостављање нове заштићене везе користи се термин *rekeying*. Нова заштићена веза се успоставља нешто раније у односу на сам истек текуће заштићене везе, у циљу елиминисања прекида у преносу података.

2.4. Стандардни шифарски алгоритми

Основна подела шифарских алгоритама дефинише два типа: симетричне и асиметричне шифарске алгоритме. Симетрични шифарски алгоритми (шифровање тајним кључем) резултат су развоја класичне криптографије. За успешно коришћење симетричног шифровања корисници пре тога морају разменити кључ који ће бити коришћен за шифровање и дешифровање садржаја. Основе безбедности коју нуде симетрични шифарски алгоритми леже у тајности кључа и немогућности добијања оригинала без његовог познавања. У најпопуларније симетричне шифарске алгоритме спадају *AES*, *DES*, *RC4* и *Blowfish*.

Основна предност симетричних шифарских алгоритама је њихова брзина. Са друге стране, основни проблем код њиховог коришћења јесте успостављање тајног кључа, односно неопходност да се тајни кључ безбедно достави до обе стране. У ту сврху се најчешће користе одвојени комуникациони канали. То може бити значајан проблем код већих комуникационих мрежа јер се за потребе безбедне комуникације мора извршити $n-1$ размена кључева (n је број чланова мреже). Једно од решења за овај проблем може бити централно тело које издаје кључеве за комуникационе сесије.

Концепт асиметричног шифровања представљен је 1976. године од стране Вајтфилда Дифија и Мартина Хелмана [27]. У свом првом раду као основну мотивацију ови аутори су навели „елиминисање потребе за одвојеним комуникационим каналима за размену кључева и стварање еквивалента писаном потпису“. Код асиметричног шифровања свака страна има пар кључева, e (јавни кључ) и d (тајни кључ). Јавни кључ је јавно доступан док се тајни кључ чува ван домета осталих корисника. Подаци шифровани јавним кључем могу се дешифровати само упареним тајним кључем. Са друге стране, тајни кључ се може искористити за потписивање дигиталног садржаја, а јавни кључ се у том случају користи за проверу аутентичности потписа. Подразумева се немогућност израчунавања тајног кључа на основу његовог јавног пара. Најпознатији представници асиметричне криптографије јесу Дифи-Хелман систем размене кључева и *RSA* алгоритам.

У поређењу са симетричним еквивалентима асиметрични системи за шифровање су углавном спорији и нуде нижи ниво безбедности за исту дужину кључева (односно,

захтевају коришћење кључева веће дужине). Са друге стране, они не захтевају одвојене канале за дистрибуцију кључева, а нуде и специфичности у погледу електронског потписивања докумената. У пракси се често среће мешовита употреба симетричног и асиметричног шифровања - асиметрично шифровање користи се за дистрибуцију кључева симетричних алгоритама.

2.4.1. *Advanced Encryption Standard, AES*

Шифарски алгоритам *Advanced Encryption Standard (AES)* један је од најпопуларнијих шифарских алгоритама данас. Овај алгоритам је одобрен од стране *NIST*-а као федерални стандард за процесирање информација [28]. У питању је симетрични блоковски алгоритам за шифровање и дешифровање података. У пракси постоје различите софтверске, фирмверске и хардверске реализације овог алгоритма.

AES алгоритам може користити кључеве дужине 128, 196 и 256 битова. Иницијална спецификација алгоритма омогућавала је и друге величине кључева али оне нису ушле у стандард. Основна јединица за обраду података код *AES* алгоритма је бајт, што значи да се улазни подаци, излазни подаци и кључ за шифровање обрађују у јединицама од једног бајта. Код *AES* алгоритма дужина улазног блока, излазног блока и стања је 128 битова. Интерно, операције *AES* алгоритма извршавају се над дводимензионалним низовима бајтова који се називају стања. Ови низови садрже по четири реда бајтова од којих сваки садржи N бајтова (N је величина блока подељена са 32).

Извршавање код *AES* алгоритма, односно број рунди (енгл. *number of rounds*), зависи од дужине кључа. За дужину кључа од 128 битова број рунди је 10, за дужину од 196 битова је 12, а за дужину кључа од 256 битова број рунди је 14. За процесе шифровања и дешифровања података *AES* алгоритам користи следеће трансформације:

1. замена бајтова на основу табеле,
2. смицање редова матрице стања по различитим величинама,
3. мешање података у оквиру колоне матрица стања и
4. додавање *Round Key* вредности (вредност израчуната на основу кључа) матрици стања.

На почетку процеса шифровања улазни низ се копира у матрицу стања. Након додавања

Round Key вредности матрица стања се трансформише коришћењем функције заокруживања 10, 12 или 14 пута, у зависности од дужине кључа. Након заокруживања, матрица стања се копира у излазни низ. Један од параметара функције заокруживања је једнодимензионални низ речи од по четири бајта, добијен на основу *Key Expansion* рутине. Рутина *Key Expansion* на основу главног кључа генерише тзв. распоред кључева (енгл. *key schedule*) дужине Nr речи (Nr је број циклуса) од по четири бајта.

```

KeyExpansion(byte key[4*Nk], word w[Nb*(Nr+1)], Nk)
begin
    word temp
    i = 0
    while (i < Nk)
        w[i] = word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])
        i = i+1
    end while
    i = Nk
while (i < Nb * (Nr+1))
    temp = w[i-1]
    if (i mod Nk = 0)
        temp = SubWord(RotWord(temp)) xor Rcon[i/Nk]
    else if (Nk > 6 and i mod Nk = 4)
        temp = SubWord(temp)
    end if
    w[i] = w[i-Nk] xor temp
    i = i + 1
end while
end

```

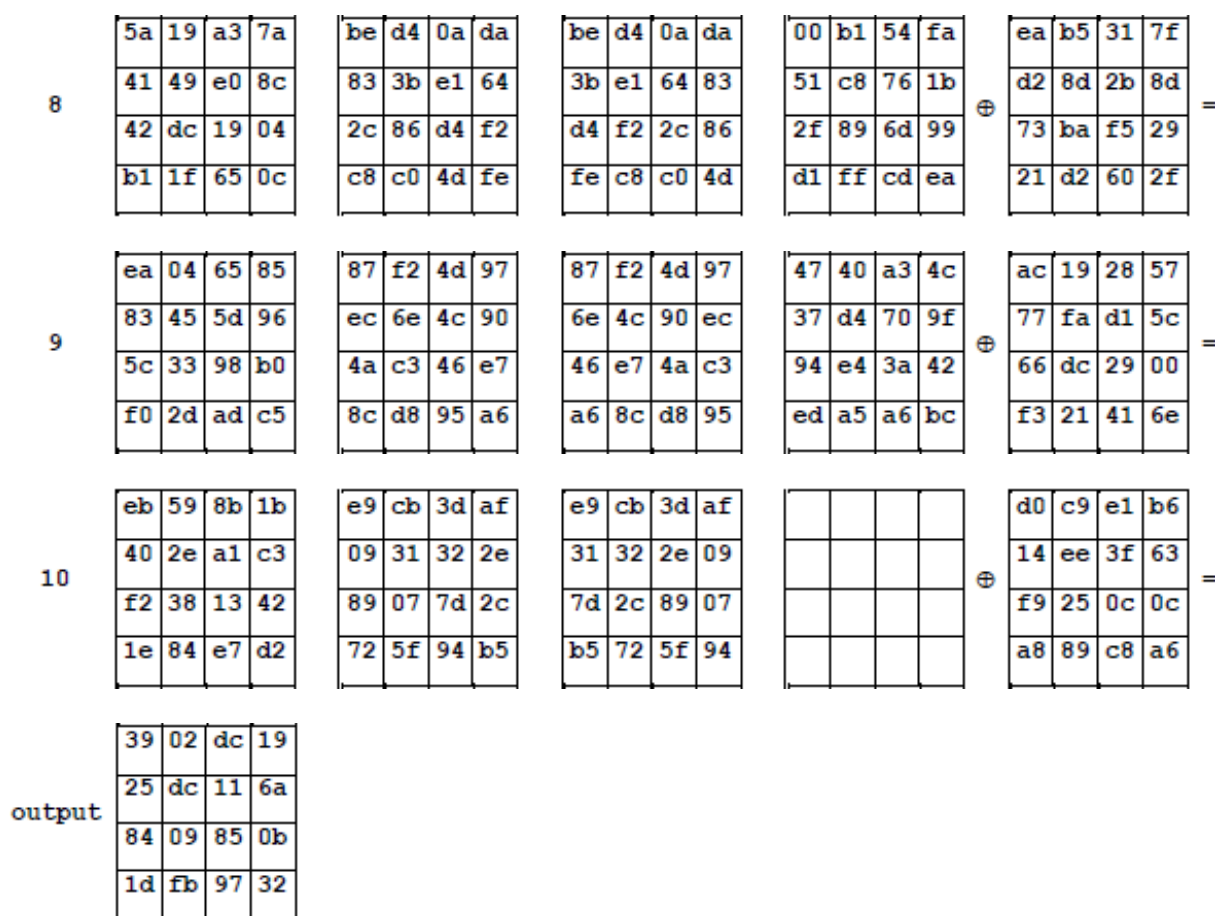
Листинг 5. Псеудо-код рутине *Key Expansion*

У наставку текста дат је процес шифровања за вредности:

садржај: 32 43 f6 a8 88 5a 30 8d 31 31 98 a2 e0 37 07 34
 кључ: 2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c

Round Number	Start of Round	After SubBytes	After ShiftRows	After MixColumns	Round Key Value																																																																																
input	<table><tr><td>32</td><td>88</td><td>31</td><td>e0</td></tr><tr><td>43</td><td>5a</td><td>31</td><td>37</td></tr><tr><td>f6</td><td>30</td><td>98</td><td>07</td></tr><tr><td>a8</td><td>8d</td><td>a2</td><td>34</td></tr></table>	32	88	31	e0	43	5a	31	37	f6	30	98	07	a8	8d	a2	34	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																	<table><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td></tr></table>																	$\oplus$ <table><tr><td>2b</td><td>28</td><td>ab</td><td>09</td></tr><tr><td>7e</td><td>ae</td><td>f7</td><td>cf</td></tr><tr><td>15</td><td>d2</td><td>15</td><td>4f</td></tr><tr><td>16</td><td>a6</td><td>88</td><td>3c</td></tr></table> =	2b	28	ab	09	7e	ae	f7	cf	15	d2	15	4f	16	a6	88	3c
	32	88	31	e0																																																																																	
	43	5a	31	37																																																																																	
	f6	30	98	07																																																																																	
a8	8d	a2	34																																																																																		
2b	28	ab	09																																																																																		
7e	ae	f7	cf																																																																																		
15	d2	15	4f																																																																																		
16	a6	88	3c																																																																																		

$$\begin{array}{l}
 1 \quad \begin{array}{|c|c|c|c|} \hline 19 & a0 & 9a & e9 \\ \hline 3d & f4 & c6 & f8 \\ \hline e3 & e2 & 8d & 48 \\ \hline be & 2b & 2a & 08 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline d4 & e0 & b8 & 1e \\ \hline 27 & bf & b4 & 41 \\ \hline 11 & 98 & 5d & 52 \\ \hline ae & f1 & e5 & 30 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline d4 & e0 & b8 & 1e \\ \hline bf & b4 & 41 & 27 \\ \hline 5d & 52 & 11 & 98 \\ \hline 30 & ae & f1 & e5 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline 04 & e0 & 48 & 28 \\ \hline 66 & cb & f8 & 06 \\ \hline 81 & 19 & d3 & 26 \\ \hline e5 & 9a & 7a & 4c \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline a0 & 88 & 23 & 2a \\ \hline fa & 54 & a3 & 6c \\ \hline fe & 2c & 39 & 76 \\ \hline 17 & b1 & 39 & 05 \\ \hline \end{array} = \\
 \\
 2 \quad \begin{array}{|c|c|c|c|} \hline a4 & 68 & 6b & 02 \\ \hline 9c & 9f & 5b & 6a \\ \hline 7f & 35 & ea & 50 \\ \hline f2 & 2b & 43 & 49 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline 49 & 45 & 7f & 77 \\ \hline de & db & 39 & 02 \\ \hline d2 & 96 & 87 & 53 \\ \hline 89 & f1 & 1a & 3b \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline 49 & 45 & 7f & 77 \\ \hline db & 39 & 02 & de \\ \hline 87 & 53 & d2 & 96 \\ \hline 3b & 89 & f1 & 1a \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline 58 & 1b & db & 1b \\ \hline 4d & 4b & e7 & 6b \\ \hline ca & 5a & ca & b0 \\ \hline f1 & ac & a8 & e5 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline f2 & 7a & 59 & 73 \\ \hline c2 & 96 & 35 & 59 \\ \hline 95 & b9 & 80 & f6 \\ \hline f2 & 43 & 7a & 7f \\ \hline \end{array} = \\
 \\
 3 \quad \begin{array}{|c|c|c|c|} \hline aa & 61 & 82 & 68 \\ \hline 8f & dd & d2 & 32 \\ \hline 5f & e3 & 4a & 46 \\ \hline 03 & ef & d2 & 9a \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline ac & ef & 13 & 45 \\ \hline 73 & c1 & b5 & 23 \\ \hline cf & 11 & d6 & 5a \\ \hline 7b & df & b5 & b8 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline ac & ef & 13 & 45 \\ \hline c1 & b5 & 23 & 73 \\ \hline d6 & 5a & cf & 11 \\ \hline b8 & 7b & df & b5 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline 75 & 20 & 53 & bb \\ \hline ec & 0b & c0 & 25 \\ \hline 09 & 63 & cf & d0 \\ \hline 93 & 33 & 7c & dc \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline 3d & 47 & 1e & 6d \\ \hline 80 & 16 & 23 & 7a \\ \hline 47 & fe & 7e & 88 \\ \hline 7d & 3e & 44 & 3b \\ \hline \end{array} = \\
 \\
 4 \quad \begin{array}{|c|c|c|c|} \hline 48 & 67 & 4d & d6 \\ \hline 6c & 1d & e3 & 5f \\ \hline 4e & 9d & b1 & 58 \\ \hline ee & 0d & 38 & e7 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline 52 & 85 & e3 & f6 \\ \hline 50 & a4 & 11 & cf \\ \hline 2f & 5e & c8 & 6a \\ \hline 28 & d7 & 07 & 94 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline 52 & 85 & e3 & f6 \\ \hline a4 & 11 & cf & 50 \\ \hline c8 & 6a & 2f & 5e \\ \hline 94 & 28 & d7 & 07 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline 0f & 60 & 6f & 5e \\ \hline d6 & 31 & c0 & b3 \\ \hline da & 38 & 10 & 13 \\ \hline a9 & bf & 6b & 01 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline ef & a8 & b6 & db \\ \hline 44 & 52 & 71 & 0b \\ \hline a5 & 5b & 25 & ad \\ \hline 41 & 7f & 3b & 00 \\ \hline \end{array} = \\
 \\
 5 \quad \begin{array}{|c|c|c|c|} \hline e0 & c8 & d9 & 85 \\ \hline 92 & 63 & b1 & b8 \\ \hline 7f & 63 & 35 & be \\ \hline e8 & c0 & 50 & 01 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline e1 & e8 & 35 & 97 \\ \hline 4f & fb & c8 & 6c \\ \hline d2 & fb & 96 & ae \\ \hline 9b & ba & 53 & 7c \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline e1 & e8 & 35 & 97 \\ \hline fb & c8 & 6c & 4f \\ \hline 96 & ae & d2 & fb \\ \hline 7c & 9b & ba & 53 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline 25 & bd & b6 & 4c \\ \hline d1 & 11 & 3a & 4c \\ \hline a9 & d1 & 33 & c0 \\ \hline ad & 68 & 8e & b0 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline d4 & 7c & ca & 11 \\ \hline d1 & 83 & f2 & f9 \\ \hline c6 & 9d & b8 & 15 \\ \hline f8 & 87 & bc & bc \\ \hline \end{array} = \\
 \\
 6 \quad \begin{array}{|c|c|c|c|} \hline f1 & c1 & 7c & 5d \\ \hline 00 & 92 & c8 & b5 \\ \hline 6f & 4c & 8b & d5 \\ \hline 55 & ef & 32 & 0c \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline a1 & 78 & 10 & 4c \\ \hline 63 & 4f & e8 & d5 \\ \hline a8 & 29 & 3d & 03 \\ \hline fc & df & 23 & fe \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline a1 & 78 & 10 & 4c \\ \hline 4f & e8 & d5 & 63 \\ \hline 3d & 03 & a8 & 29 \\ \hline fe & fc & df & 23 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline 4b & 2c & 33 & 37 \\ \hline 86 & 4a & 9d & d2 \\ \hline 8d & 89 & f4 & 18 \\ \hline 6d & 80 & e8 & d8 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline 6d & 11 & db & ca \\ \hline 88 & 0b & f9 & 00 \\ \hline a3 & 3e & 86 & 93 \\ \hline 7a & fd & 41 & fd \\ \hline \end{array} = \\
 \\
 7 \quad \begin{array}{|c|c|c|c|} \hline 26 & 3d & e8 & fd \\ \hline 0e & 41 & 64 & d2 \\ \hline 2e & b7 & 72 & 8b \\ \hline 17 & 7d & a9 & 25 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline f7 & 27 & 9b & 54 \\ \hline ab & 83 & 43 & b5 \\ \hline 31 & a9 & 40 & 3d \\ \hline f0 & ff & d3 & 3f \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline f7 & 27 & 9b & 54 \\ \hline 83 & 43 & b5 & ab \\ \hline 40 & 3d & 31 & a9 \\ \hline 3f & f0 & ff & d3 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline 14 & 46 & 27 & 34 \\ \hline 15 & 16 & 46 & 2a \\ \hline b5 & 15 & 56 & d8 \\ \hline bf & ec & d7 & 43 \\ \hline \end{array} \oplus \begin{array}{|c|c|c|c|} \hline 4e & 5f & 84 & 4e \\ \hline 54 & 5f & a6 & a6 \\ \hline f7 & c9 & 4f & dc \\ \hline 0e & f3 & b2 & 4f \\ \hline \end{array} =
 \end{array}$$

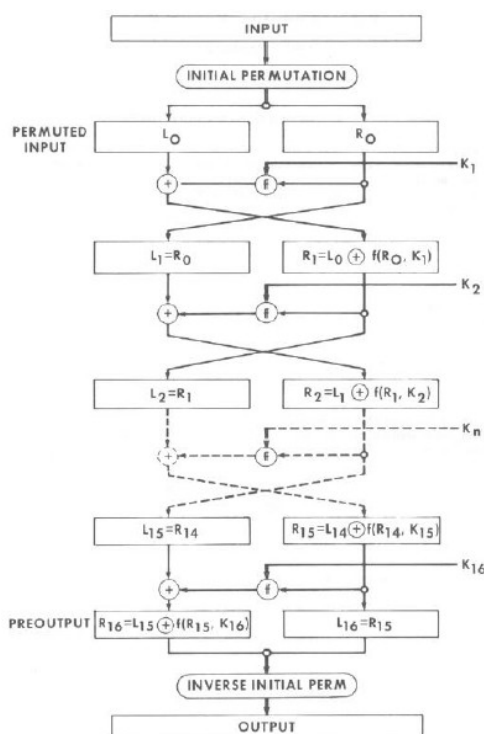
Слика 23. Итерације у процесу шифровања код *AES* алгоритма

2.4.2. *Data Encryption Standard (DES, 3DES)*

Шифарски алгоритам *Data Encryption Standard (DES)* је симетрични блоковски шифарски алгоритам, одобрен од стране *NIST*-а као један од федералних стандарда за процесирање информација [29]. Овај алгоритам је пројектован за шифровање и дешифровање података у блоковима дужине 64 бита. Кључеви који се користе за рад са *DES* алгоритмом такође имају дужину од 64 бита од чега се 56 користе за шифровање и дешифровање док се преосталих 8 бита користе за препознавање грешака.

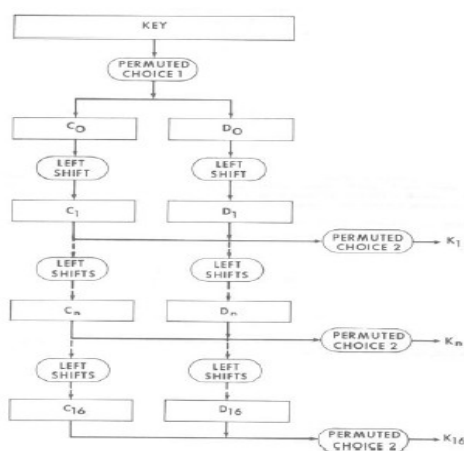
Први корак у шифровању података код *DES* алгоритма подразумева иницијалну пермутацију (I_p). Над резултатом иницијалне пермутације се, затим, врше сложене рачунске операције (ознака f) у којима као параметар учествује и кључ (над којим је извршена функција за израчунавање под-кључа, *key scheduling*, KS) којим се врши шифровање. Над резултатом ових операција се извршава процес реверзан иницијалној пермутацији (I_p^{-1}). Резултатом ових операција добија се шифрат, а реверзним

операцијама врши се његово дешифровање, односно, добија се оригинал.



Слика 24. Алгоритам процеса шифровања *DES* алгоритма

Само увођење *DES* алгоритма као стандарда за шифровање пратили су различити проблем и нејасноће. Развој овог алгоритма иницирао је *NIST* (тадашњи *NBS, National Bureau of Standards*) почетком седамдесетих година прошлога века. Међутим, испоставило се да ни једна од предложених реализација није испоштовала постављене високе стандарде. Након консултација са *NSA* агенцијом прихваћено је решење које је понудила компанија *IBM* а базирано је на претходно развијеном *Lucifer* алгоритму аутора Хорса Фистела.



Слика 25. Алгоритам израчунавања кључева

DES алгоритам је критикован и од стране Мартина Хелмана и Вајтфилда Дифија који су указали на скаћивање кључева и мистериозне *S* боксове, као последице утицаја *NSA* агенције на развој алгоритма. Претпоставља се да је било у питању намено ослабљивање алгоритма у циљу омогућавања разбијања шифрата од стране владиних агенција. И поред свих наведених проблема са овим алгоритмом он се и данас у пракси често среће у својој *3DES* варијанти.

2.4.3. *RSA*

Шифарски алгоритам *RSA* добио је име по својим ауторима Рону Рајвесту, Адију Шамиру и Леонарду Адлеману (*Rivest-Shamir-Adleman*). У питању је асиметрични шифарски алгоритам који се, осим за шифровање података, може користити и за потписивање електронских докумената.

Асиметрични рад *RSA* алгоритма подразумева постојање два кључа - јавног и тајног. Јавни кључ се састоји од два позитивна цела броја - модула и јавног експонента. Приватни кључ *RSA* алгоритма може се представити на два начина: преко модула и тајног експонента (позитивни цели бројеви), или преко седам параметара - први фактор, други фактор, први фактор *CRT* (*Chinese Remainder Theorem*) експонента, други фактор *CRT* експонента, први *CRT* коефицијент, *i*-ти фактор, *i*-ти фактор *CRT* експонента, *i*-ти фактор *CRT* коефицијента.

Шифарски алгоритам *RSA* представља групу асиметричних алгоритама. Основна предност ових алгоритама, у односу на симетричне шифарске алгоритме, јесте могућност да се између две стране изврши размена шифрованих података а да није неопходна претходна размена кључа заштићеним каналом. Са друге стране, симетрични шифарски алгоритми обично нуде боље перформансе у раду, а њихова безбедност не зависи од протокола за иницијалну размену кључа.

2.5. Трендови

Трендови везани за безбедност су у директној вези са општим трендовима на пољу рачунарских и телекомуникационих мрежа, односно са вредностима реалног света које се преносе њиховим каналима. На основу различитих истраживања [30, 31, 32, 33] као најважније области за испитивање безбедносних трендова издвајају се:

1. неовлашћени приступ,
2. проблем инсајдера и
3. крађа идентитета.

Још једна значајна карактеристика издвојених области јесте њихово често преплитање, односно повезан утицај различитих фактора. На пример, неовлашћени приступ ресурсима често се дешава као последица сарадње са инсајдерима, или као последица крађе електронског идентитета особе која је овлашћена за приступ електронским ресурсима.

Анкета [33] коју је 2004. године водио *CSO* магазин у сарадњи са Америчком безбедносном службом и координационим центром *CERT*-а, показује да бивше или тренутно запослене или привремено ангазоване особе представљају другу по реду најзначајнију претњу пословним системима. Један од значајних фактора који утиче на овај проблем јесте све већа могућност остваривања материјалне користи путем неовлашћеног присвајања електронских података уместо материјалних добара. Починиоци криминалних радњи базираних на инсајдерима у великој већини случајева подразумевају инсајдере – запослене – као директне починиоце а у остатку случајева запослени најчешће омогућава спољном нападачу приступ одређеним информацијама или функцијама.

По речима безбедносног експерта Гарија Барнета из фирме Овум једна од карактеристика 2009. године ће бити и примена безбедносних лекција научених ранијих година у пословним системима. Са следећим Барнетовим ставом слаже се и Мајк Гилеспи, директор безбедности консултантске фирме Адвентим – у 2009. години ће се знатно порадити и на заштити података од случајног или намерног уништавања или тзв. „цурења“ у јавност.

Оливер Фридрих, један од менаџера компаније Симантек, указује и на безбедносне проблеме који се могу јавити као последица убрзане примене концепта виртуализације рачунара. „Брзина и лакоћа којом се могу уводити виртуални рачунари може успавати људе по питању разматрања правилног обезбеђивања виртуалних рачунара и окружења у коме се налазе“ наводи Фридрих.

Безбедносни концепти који се представљају у овом раду подразумевају укључивање малог броја људи у процес остваривања и одржавања безбедности. Додатно,

представљено решење се базира на нижим слојевима OSI и TCP/IP модела и потпуно је ван видљивости и управљања од стране крајњих корисника. Ове особине знатно смањују ризик који потиче од инсајдера. Такође, мали број људи укључених у процес остваривања и одржавања безбедности повољно утиче и на смањење ризика од крађе идентитета комуникационих страна.

2.6. Стандарди, референтна тела и организације

Данас, доминантну улогу на пољу рачунарске безбедности и њене стандардизације имају САД. Водеће организације на пољу стандардизације безбедносних система функционишу као агенције владе САД. Такође, и значајне организације које имају утицаја на глобалне трендове у развоју информационих технологија лоциране су на северноамеричком континенту.

2.6.1. Међународне организације

Интернет корпорација за додељивање назива и бројева (енгл. *Internet Corporation for Assigned Names and Numbers, ICANN*) задужена је за управљање адресама на Интернету, одређивање аутономних система, администрацију кореног система доменских имена, одређивање бројева за протоколе и сл. Пре формирања ове корпорације при америчком министарству трговине, њене функције су вршене на Институту за информатичке науке при Универзитету Јужна Калифорнија, на основу уговора са америчким министарством одбране. При овој корпорацији функционише и *IANA (Internet Assigned Numbers Authority)*.

Специјална комисија за развој интернета (енгл. *Internet Engineering Task Force, IETF*) задужена је за развој и промовисање Интернет стандарда. Учесници ове организације су волонтери, с тим да је њихов рад плаћен од стране њихових послодаваца или спонзора - тренутног председавајућег ове организације спонзоришу компанија *VeriSign* и америчка Национална безбедносна агенција.

Међународна телекомуникациона унија (енгл. *International Telecommunication Union ITU*) је агенција при Уједињеним нацијама, са седиштем у Женеви. Основни циљ ове агенције је развој телекомуникационе инфраструктуре и успостављање међународних стандарда.

Европски институт за телекомуникационе стандарде (енгл. *European Telecommunications Standards Institute, ETSI*) је независна, непрофитна организација, призната од стране европске комисије, а основана 1988. године од стране европске конференције за поштанску и телекомуникациону администрацију. Основни допринос ове организације је у развоју стандарда за мобилну комуникацију.

Фондација за слободан софтвер (енгл. *Free Software Foundation, FSF*) непрофитна је организација основана са циљем да промовише слободу у коришћењу рачунарског софтвера, као и да заштити права корисника тог софтвера. Један од главних доприноса ове фондације је развој *GNU* оперативног система и допринос развоју софтвера са отвореним изворним кодом, првенствено Линукс оперативног система.

2.6.2. Националне организације

Једна од најзначајнијих организација за стандардизацију на пољу рачунарске безбедности је америчка Национална безбедносна агенција (енгл. *National Security Agency, NSA*). Ова агенција је основана 1952. године са циљем да пружи услуге надгледања комуникација страних обавештајних служби и заштиту комуникација владе САД. Од 2008. године ова агенција је задужена и за заштиту владиних рачунарских система и мрежа. Ова агенција је чланица обавештајне заједнице САД (енгл. *U.S. Intelligence Community*). Координацију између Националне безбедносне агенције и војске САД обавља Централна безбедносна служба (енгл. *Central security service*). Национална безбедносна агенција је имала значајан утицај на развој актуелних шифарских алгоритама, *AES* и *DES*, Клипер криптографског чипа и сл. Ова агенција је инвестирала милионе долара у академски развој на пољу безбедности, али су познати и примери у којима је она утицала на забрану објављивања одређених резултата (нпр. шифарски алгоритми *Khufi* и *Khafre*).

Амерички Национални институт за стандардизацију и технологију (енгл. *National Institute of Standards and Technology, NIST*), раније познат под именом Национални биро за стандардизацију (енгл. *National Bureau of Standards, NBS*), основан је 1901. године као агенција департмана за трговину владе САД. Као званична мисија института наведена је промоција иновативности и индустријске конкурентности САД путем унапређивања науке, стандарда и технологије. Овај институт је учествовао у стандардизацији популарних шифарских алгоритама, *AES* и *DES*.

3. Развој сопственог система заштите на мрежном слоју

У овом делу рада дата су уводна разматрања везана за различите аспекте развоја сопственог система заштите комуникација, као и процес његовог развоја и интеграције у стандардну комуникациону инфраструктуру. У поглављу су дати и процеси одабира комуникационе инфраструктуре у коју ће систем заштите бити интегрисан, као и избор носећих технологија (хардверске платформе, оперативног система и комуникационих протокола), рађених на основу анализе тренутног стања на пољу телекомуникационих технологија и безбедности, дате у претходном делу рада.

3.1. Избор комуникационе инфраструктуре

У делу рада који се бави анализом популарних комуникационих инфраструктура Интернет мрежа и њене технологије идентификоване су као циљна инфраструктура у коју би био интегрисан сопствени систем заштите. У аргументе овакве одлуке спадају глобална доступност, отвореност технологија, велики број корисничких сервиса, и т.д. а они су прецизније обрађени у наведеном делу рада. У складу са изабраном комуникационом инфраструктуром вршен је и избор носећих технологија решења - комуникационих протокола, оперативних система и хардверске платформе.

3.2. Избор носећих технологија

Комплетан развој свих компонената система, хардверских и софтверских, за заштиту комуникација на изабраној инфраструктури, представља огромну инвестицију са неизвесним резултатима. Алтернативу овом приступу чини коришћење већ развијених технологија у циљу обезбеђивања општих функција система, уз активно ангажовање на развоју специфичних безбедносних функција. Избор већ развијених носећих технологија система мора се вршити у складу са критеријумом компатибилности са постављеним циљем - развој сопственог система заштите у циљу постизања вишег безбедносног нивоа и елиминисања познатих, или претпостављених рањивости популарних решења. У основне технологије на које би се ослањао сопствени систем заштите спадају хардверска платформа, носећи оперативни систем и комуникациони протоколи.

3.2.1. Избор комуникационих протокола

Као основни комуникациони протокол који ће бити коришћен за интеграцију сопственог система заштите, а на основу анализе дате у претходном делу рада, изабран је *IPsec* протокол. У складу са тим обезбеђена је компатибилност са *IP* протоколом на мрежном ниову и са *IP*-компатибилним протоколима на транспортном и апликативном ниову. Такође, овим избором остварена је и компатибилност са свим инфраструктурама чији су протоколи на физичком ниову и на ниову везе података компатибилни са *IP* протоколом.

3.2.2. Избор оперативног система

У основне критеријуме које треба да испуни оперативни систем на коме би се извршавало предложено решење спадају:

1. доступност изворног кода оперативног система,
2. доступност изворног кода свих компонената које ће бити коришћене,
3. доступност изворног кода *IPsec* имплементације,
4. подршка за актуелне мрежне интерфејсе и инфраструктуру,
5. високе перформансе мрежних функција и
6. могућност постизања високог нивоа безбедности система.

На основу наведених критеријума, као основни систем на коме ће бити реализовано решење одабран је Линукс оперативни систем. Комплетан изворни код језгра овог система јавно је доступан. Систем се дистрибуира у виду дистрибуција (које поред језгра садрже и додатне софтверске пакете, као и алате за инсталацију и администрацију), а за реализацију предложеног система одабрана је Слеквер (енгл. *Slackware*) дистрибуција. Услов да неки софтверски пакет уђе у ову дистрибуцију јесте да је његов изворни код јавно доступан.

Подршка Линукс оперативног система за актуелне мрежне интерфејсе и инфраструктуру може се сматрати релативно добром. Са безбедносног аспекта је посебно значајна чињеница да су мрежни интерфејси популарних произвођача углавном подржани драјверима чији је изворни код јавно доступан (и уграђен у само језгро оперативног система).

Сам *IPsec* систем заштите директно је уграђен у језгро Линукс оперативног система. Самим тим, изворни код имплементације је доступан, могуће га је мењати, а уједно су омогућене високе перформансе у раду. Такође, перформансе општих мрежних функција Линукс оперативног система су релативно високе (у поређењу са сличним оперативним системима на тржишту) а уграђени фајервол систем *iptables* нуди могућност подизања безбедности система кроз филтрирање пакета на нивоу језгра оперативног система.

Конфигурабилност Линукс оперативног система омогућава подизање система на висок ниво кроз укључивање минималног сета неопходних функционалности. На тржишту постоји и велики број отворених и затворених допунских софтвера који се користе за подизање безбедносног нивоа Линукс оперативног система.

3.2.3. Избор хардверске платформе

Три основна критеријума која хардверска платформа мора да испуни, да би била коришћена за практичну употребу са предложеним системом, јесу:

1. компатибилност са изабраним оперативним системом,
2. компатибилност са планираном комуникационом инфраструктуром на којој ће се користити,
3. перформансе у складу са планираним обимом примене.

У складу са наведеним, за коришћење предложеног система заштите може се користити било која од хардверских архитектура коју подржава сам Линукс оперативни систем: *IA-32, MIPS, x86-64, SPARC, DEC Alpha, Itanium, PowerPC, ARM, m68k, PA-RISC, s390, SuperH, M32R* и друге. Конкретан избор првенствено зависи од планираног обима коришћења система (обима саобраћаја који ће бити шифрован), услова у којима ће се систем користити (да ли је у питању интегрисани систем који треба да издржи велике температурне осцилације, високу влажност и сл.) и компатибилности са комуникационом инфраструктуром (постојања адекватних физичких интерфејсова).

Са безбедносног аспекта, критеријум који свака од хардверских компонената система мора да испуни (да би била коришћена на развојним и оперативним рачунарима) јесте њена отворена архитектура и непостојање било каквих недокументованих функција. Овај критеријум је детаљније анализиран у наставку рада.

3.3. Успостављање чистог развојног и оперативног рачунарског система

У уводном делу рада појам „чистог рачунарског система“ дефинисан је (преко атрибута „чисти“) као систем у коме се не налази ни једна компонента чији детаљи функционисања нису у потпуности познати ауторима, односно не постоји ни једна функција у систему коју аутори нису развили или одобрили. У наставку текста биће анализирани компоненте које могу угрозити безбедност система (развојног или оперативног) кроз увођење недокументованих функција.

Чист рачунарски систем не сме да изврши ни једну функцију која кориснику није позната. Узимајући у обзир основну рачунарску архитектуру то значи да до централног процесора не сме доћи ни једна инструкција коју аутор система нија развио или одобрио. Инструкције потичу из следећих извора наведеним редоследом:

1. BIOS,
2. програм за учитавање оперативног система (енгл. *boot-loader*),
3. оперативни систем (са уграђени *IPsec*-ом и шифарским алгоритмом) и
4. кориснички софтвер

Да би се испунио наведени критеријум (да до централног процесора не сме доћи ни једна инструкција коју аутор система нија развио или одобрио) морају се испунити два услова:

1. изворни код свих наведених софтверских компонената мора бити написан или проверен од стране корисника и
2. изворни код свих наведених софтверских компонената мора бити преведен у извршни облик и повезан са захтеваним библиотекама коришћењем проверених преводиоца и линкера, и у чистом окружењу.

Из рекурзије у наведеном види се да се за успостављање иницијалног чистог рачунарског система не могу користити већ готови компајлери и окружења, већ се они морају самостално развити на асемблерском нивоу, или чак на нивоу писања бинарних инструкција за процесор на коме ће бити извршено иницијално превођење оперативног система, библиотека, компајлера и линкера написаних у програмском језику вишег нивоа.

Изворни код Линукс оперативног система и свог коришћеног софтвера (као и основне библиотеке *glibc* (*GNU C libraries*)) јавно је доступан или је самостално написан. Такође, изворни код програма за учитавање оперативног система *Linux Loader*, *LILO* јавно је доступан. Међутим, изворни код *BIOS* фирмвера који се испоручују уз матичне плоче није јавно доступан. Као замена (чији је изворни код доступан) за *BIOS* фирмвер може се користити *coreboot*.

Пројекат *coreboot* започет је 1999. године под називом *LinuxBIOS* са циљем омогућавања потпуне контроле над рачунаром од тренутка када се он укључи [35]. Изворни код *coreboot*-а јавно је доступан под *GPL* лиценцом. На званичном сајту пројекта налази се листа компатибилних матичних плоча. Превођењем изворног кода *coreboot*-а (на чистом систему) он би се могао искористити као адекватна замена *BIOS* фирмвера за успостављање чистог рачунарског система.

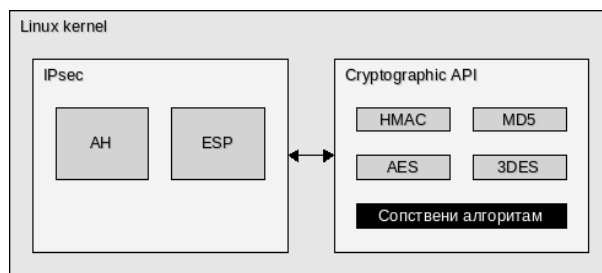
Са аспекта хардвера у чистом рачунарском систему не смеју постојати недокументоване функционалности ниједне од компонената које могу мењати или емитовати податке. Овде се првенствено мисли на централни процесор и на микро-процесоре у осталим хардверским компоненатама. За драјвере компонената важе иста правила као и за остали софтвер, првенствено оперативни систем.

3.4. Интеграција сопственог шифарског алгорита у *IPsec* систем заштите

У овом делу рада документован је комплетан процес интеграције сопственог шифарског алгорита у *IPsec* систем заштите комуникација. Документовање процеса интеграције шифарског алгорита засновано је на теоријским основама рада са Линукс оперативним системом [34] и на искуствима из практичне реализације. За све наведене кораке подразумева се коришћење раније дефинисаног чистог рачунарског система.

3.4.1. Модел и процес интегрисања сопственог шифарског алгорита

Први корак у процесу интегрисања сопственог шифарског алгорита у језгро Линукс оперативног система за потребе коришћења са *IPsec* системом заштите јесте реализација шифарског алгорита у програмском језику *C* уз примену правила *cryptographic API*-ја језгра.



Слика 26. Проширивање скупа шифарских алгоритама језгра сопственим алгоритмом

Алгоритму се мора доделити име уз поштовање ограничења која важе за именовање функција и променљивих у програмском језику C. За потребе ранијих радова у овој области развијен је алгоритам са радним називом *fbisca*. Реализација овог алгоритма у програмском језику C снимљена је у фајл *crypto/fbisca.c* у директоријуму са изворним кодом језгра Линукс оперативног система.

3.4.1.1. Повезивање алгоритма са језгром оперативног система

Повезивање са криптографским апликативним програмским интерфејсом Линукс оперативног система односи се, пре свега, на правила именовања функција алгоритма. У складу са тим, везне функције сопственог алгоритма назване су именима:

- *crypto_fbisca_set_key()*
- *fbisca_encrypt()*
- *fbisca_decrypt()*

Такође, уколико има потребе за иницијализацијом алгоритма пре употребе (што је код сопственог алгоритма био случај), иста се може позвати путем опште функције за иницијализацију модула језгра: *module_init(naziv_funkcije)*. За потребе сопственог алгоритма развијена је функција *fbisca_init()* која у основи има два задатка:

1. креирање матрице која ће се користити за шифровање,
2. регистрација алгоритма у скупу шифарских алгоритама језгра.

3.4.1.2. Регистрација шифарског алгоритма

Регистрација алгоритма у скупу шифарских алгоритама језгра важан је корак јер се њиме омогућава употреба алгоритма у осталим модулима (пре свега у *IPsec*-у у овом случају). За регистрацију алгоритма користи се системска функција

crypto_register_alg() којој се као аргумент прослеђује структура у којој је описан шифарски алгоритам. Уграђена функција *crypto_unregister_alg()* служи за ослобађање ресурса након краја рада алгоритма за шифровање. Аргумент ове функције је идентичан аргументу функције *crypto_register_alg()*.

Алгоритам се описује структуром путем које се, као што је речено, врши његова регистрација и ослобађање заузетих ресурса. У овој структури се дефинишу и параметри алгоритма као што су величина блока, најмања и највећа дужина кључа, итд. Једна од најважнијих информација која се чува у структури јесте и које функције ће се користити за шифровање и дешифровање података. У случају сопственог алгоритма, путем структуре је за потребе шифровања података предвиђена функција *fbisca_encrypt()* а за потребе дешифровања података функција *fbisca_decrypt()*.

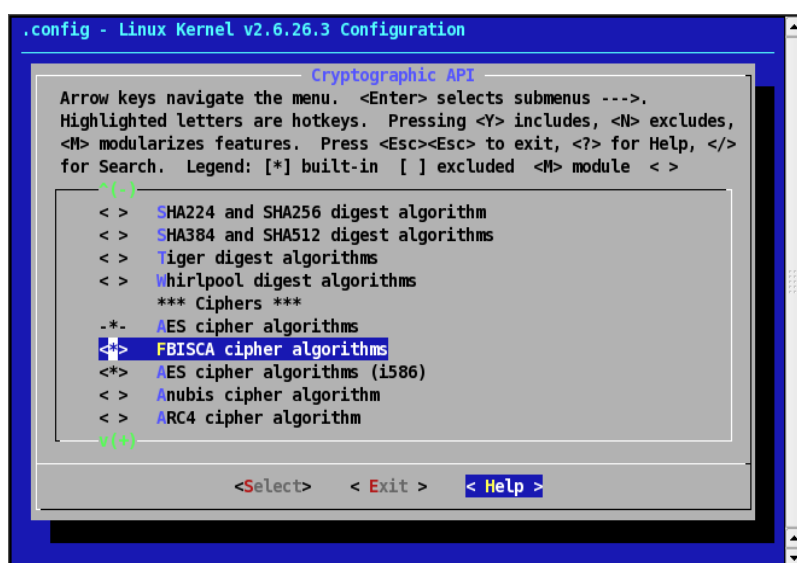
```
static struct crypto_alg fbisca_alg = {
    .cra_name      = "fbisca",
    .cra_driver_name = "fbisca",
    .cra_priority   = 100,
    .cra_flags      = CRYPTO_ALG_TYPE_CIPHER,
    .cra_blocksize  = FBISCA_BLOCK_SIZE,
    .cra_ctxsize    = sizeof(struct crypto_fbisca_ctx),
    .cra_alignmask  = 3,
    .cra_module     = THIS_MODULE,
    .cra_list       = LIST_HEAD_INIT(fbisca_alg.cra_list),
    .cra_u          = {
        .cipher = {
            .cia_min_keysize = FBISCA_MIN_KEY_SIZE,
            .cia_max_keysize = FBISCA_MAX_KEY_SIZE,
            .cia_setkey      = crypto_fbisca_set_key,
            .cia_encrypt     = fbisca_encrypt,
            .cia_decrypt     = fbisca_decrypt
        }
    }
};
```

Листинг 6. Структура за повезивање шифарског алгоритма са језгром

Следећи корак у процесу интеграције сопственог шифарског алгоритма јесте и повезивање са процесом конфигурисања језгра и његовог превођења у бинарни облик. Ово повезивање се врши путем фајлова *Kconfig* и *Makefile*. Оригиналан садржај ових фајлова за језгро верзије 2.6.26.3 дат је у наставку рада.

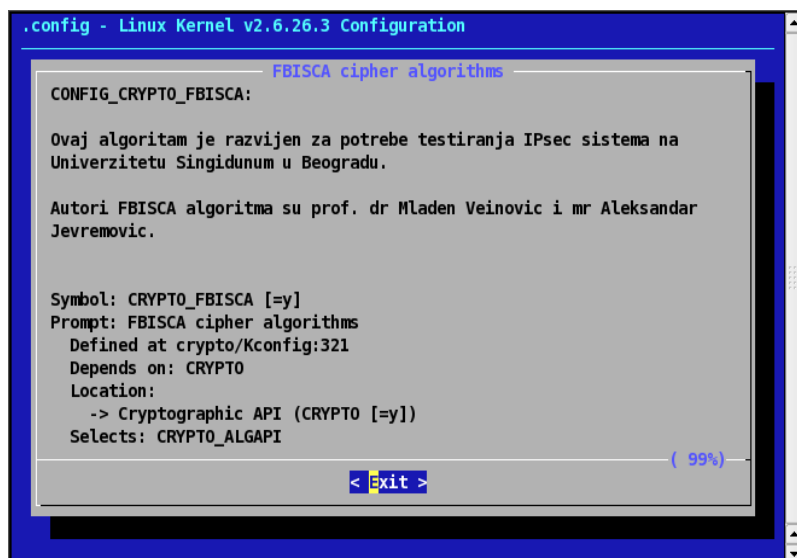
3.4.1.3. Конфигурисање и превођење измењеног оперативног система

Конфигурисање језгра се може обавити путем конзолних и графичких алата. За покретање програма за конзолно конфигурисање језгра користи се наредба *make meniconfig* док се за покретање графичког програма за конфигурисање користи наредба *make xconfig*. За извршавање обе наредбе неопходно је користити налог са администраторским привилегијама. За покретање графичког алата за конфигурисање језгра неопходно је имати инсталиран и покренут *X11* графички систем на рачунару.



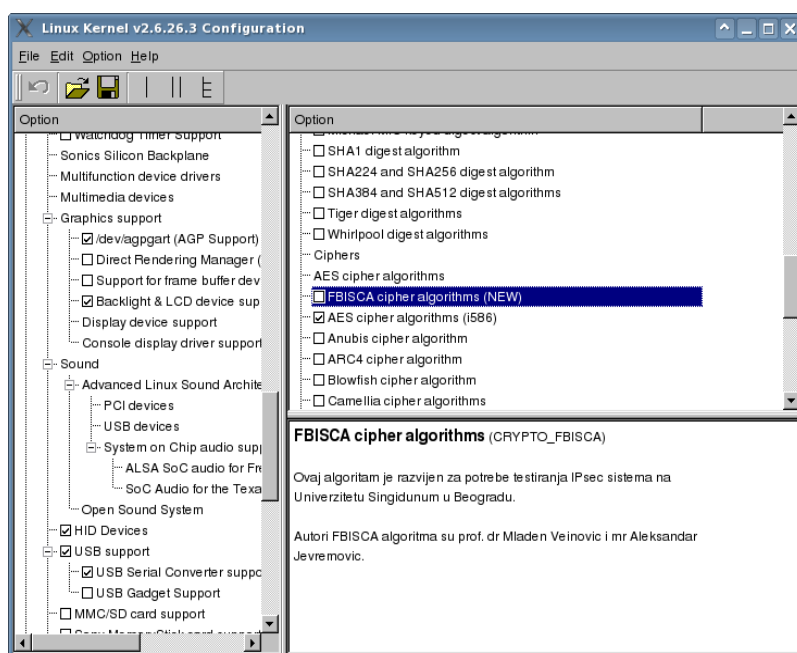
Слика 27. Избор сопственог алгоритма у конзолном конфигурационом програму

Конфигурациони програм, било да је конзолни или графички, након правилног повезивања кроз *Kconfig* фајл, у секцији везаној за криптографски програмабилни интерфејс језгра нуди могућност избора сопственог шифарског алгоритма. Алгоритам може бити одабран као модул или као саставни део језгра. Додатно, у помоћној документацији везаној за алгоритам налази се опис алгоритма (у овом случају кратак опис алгоритма и потпис аутора).



Слика 28. Опис сопственог алгоритма у конзолном конфигурационом програму

Помоћни екран у конзолном режиму рада такође приказује и зависности алгоритма (у случају сопственог алгоритма се једино захтева подршка за криптографску секцију језгра) и адресу *Kconfig* фајла на којој је алгоритам дефинисан.



Слика 29. Избор сопственог алгоритма у графичком конфигурационом програму

Фајл *Kconfig* је задужен за повезивање компонената језгра са програмима за његово конфигурисање. У оквиру листинга 7 дат је садржај којим се сопствени алгоритам повезује са системом за конфигурисање језгра.

```

config CRYPTO_FBISCA
    tristate "FBISCA cipher algorithms"
    select CRYPTO_ALGAPI
    help
        Ovaј algoritam je razvijen za potrebe testiranja IPsec sistema na
        Univerzitetu Singidunum u Beogradu.

        Autori FBISCA algoritma su prof. dr Mladen Veinovic i mr Aleksandar
        Jevremovic.

```

Листинг 7. Садржај *Kconfig* који се односи на сопствени алгоритам

При конфигурисању језгра је, осим сопственог алгоритма, неопходно укључити и стандардне компоненте *IPsec* система заштите. Реализација тих компонената се налази у оквиру следећих модула језгра:

1. *INET_AH*
2. *INET_ESP*
3. *INET_IPCOMP*
4. *INET_XFRM_MODE_TRANSPORT*
5. *INET_XFRM_MODE_TUNNEL*
6. *INET_XFRM_MODE_BEET*

Следећи корак, након конфигурисања језгра, јесте превођење иворног кода одабраних компонената у бинарни облик. Овај процес је аутоматизован и покреће се наредбом *make*. Ова наредба користи *Makefile* фајлове секција и на основу њих преводи и именује одабране компоненте. За потребе сопственог алгоритма *Makefile* у *crypto* секцији допуњен је кодом из листинга 8.

```
obj-$(CONFIG_CRYPTO_FBISCA) += fbisca.o
```

Листинг 8. Код за допуну *Makefile* фајла за укључивање сопственог алгоритма

Описаним скупом корака извршена је интеграција сопственог шифарског алгоритма у језгро Линукс оперативног система. Следећи корак јесте инсталација језгра и модула и њихово учитавање као оперативног система рачунара. Интегрисани сопствени алгоритам се, осим за потребе шифровања мрежних комуникација, може користити и за све друге примене код којих се користе алгоритми из криптографске секције језгра. На пример, овај алгоритам се може користити за шифровање фајл-система.

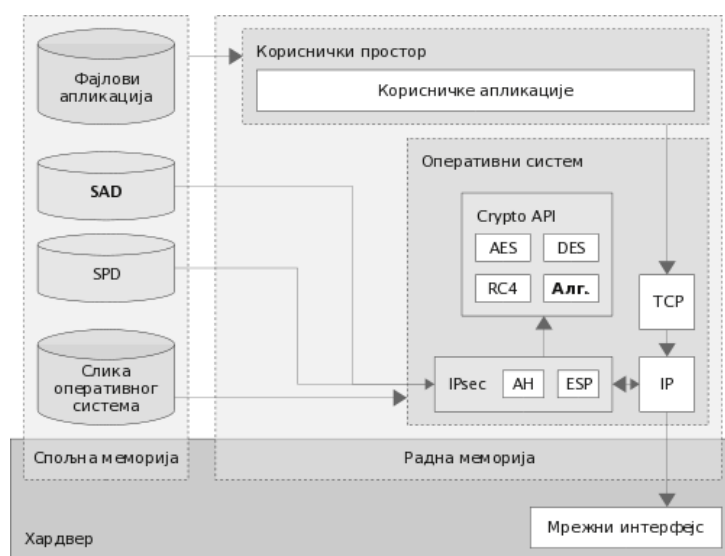
3.5. Модели управљања кључевима са криптоанализом

Предложени систем се заснива на сопственом шифарском алгоритму, имплементацији *IPsec* система за заштиту комуникација у Линукс оперативном систему, као и на осталим коришћеним компонентама Линукс оперативног система. Изворни код оперативног система и свих осталих коришћених софтверских компонената јавно је доступан. Систем је рањив кроз све актуелне безбедносне пропусте у реализацији коришћених софтверских компонената, као и кроз безбедносне пропусте који ће бити откривени у будућности. У том погледу систем нуди исти ниво заштите као и системи засновани на поменутих компонентама који користе стандардне шифарске алгоритме. Претпоставља се коришћење стандардних хардверских компонената.

Сопствени шифарски алгоритам је развијен на *C* програмском језику и у основном случају се налази на спољној или у *RAM* меморији као модул оперативног система, у бинарном облику. Изворни код сопственог шифарског алгоритма није присутан на систему који се користи. У случају остваривања приступа садржају спољне или *RAM* меморије система (могуће и удаљеним приступом), нападач може применити анализу бинарне варијанте шифарског алгоритма да би утврдио његов принцип рада. У случају хардверске реализације шифарског алгоритма нападач би морао да има физички приступ систему за анализу рада шифарског алгоритма. У поређењу са системима који користе стандардне шифарске алгоритме, предложени систем нуди виши ниво заштите с обзиром да нападачу није познат принцип рада шифарског алгоритма, као и да се користе кључеви вишеструко веће битске дужине. Међутим, у оквиру криптоанализа за различите предложене случајеве подразумеваће се, као додатна мера заштите при провери система, да је принцип рада сопственог шифарског алгоритма познат нападачу. Један од најважнијих корака у остваривању безбедне комуникационе платформе представља обезбеђивање адекватног система и полиса за размену кључева. У подразумеваном случају коришћења *IPsec* протокола под Линукс оперативним системом (Слика 30) унутрашњи кључеви се чувају у отвореном облику у фајлу који садржи базу безбедоносних асоцијација (енгл. *Security Association Database*).

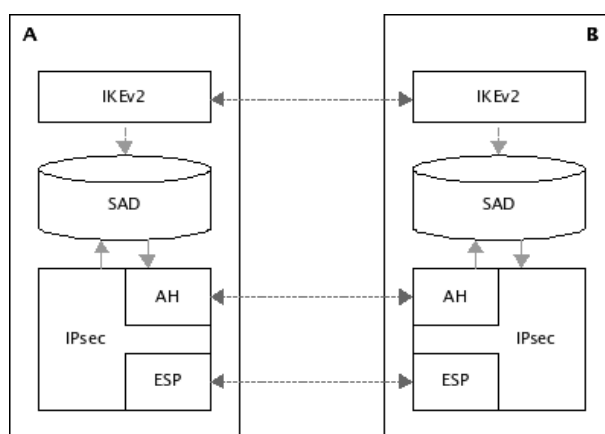
Код коришћења система у основном моделу критичну тачку система представља унутрашњи кључ, односно случај у коме нападач успе да оствари приступ унутрашњем кључу. Овај случај је могућ у случају да нападач искористи неки од потенцијалних

пропушта у дизајну и реализацији неке од коришћених компонената (оперативног система, имплементације протокола и сл.). С обзиром да се подразумева коришћење јавног сесијског кључа, приступ нападача унутрашњем кључу компромитовао би систем јер би нападачу омогућио ишчитавање садржаја свих будућих заштићених комуникација, као и претходних комуникација чији су садржај и сесијски кључ сачувани. Очигледно је да основни модел подразумева висок ниво ризика тако да у наставку рада предлажемо моделе управљања кључевима који нуде смањивање ризика кроз увођење нових компоненти и принципа за управљање кључевима.



Слика 30. Основна шема реализације предложеног система заштите

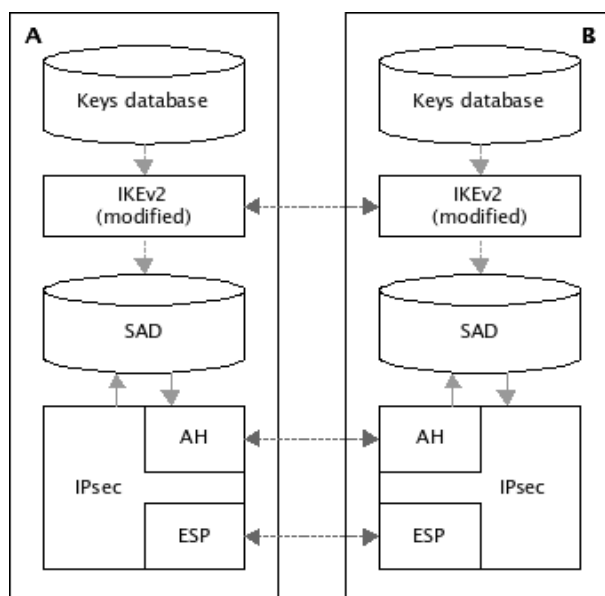
Основни протокол за динамичко успостављање тајних кључева за потребе *IPsec* протокола јесте *Internet Key Exchange* протокол (тренутно актуелна друга верзија, *IKEv2*). Веза *IKEv2* протокола са *IPsec*-ом представљена је на слици 31.



Слика 31. Оригинална *IPsec/IKEv2* архитектура

У оригиналној поставци *IKEv2* протокол користи *PKI* инфраструктуру и Дифи-Хелман успостављање кључева [6]. Дифи-Хелман је добро позната процедура за аутоматско успостављање тајних симетричних кључева. Ова процедура је компликована за имплементацију, посебно када су у питању кључеви велике битске дужине, а у потпуности се уклапа у основни симетрични шифарски систем. С обзиром да би у случају коришћења изворне варијанте *IKEv2* протокола ниво заштите који систем нуди у потпуности зависио од Дифи-Хелман процедуре (која није прихватљива за професионалне системе), у наставку рада је предложено неколико модела измена *IKEv2* протокола.

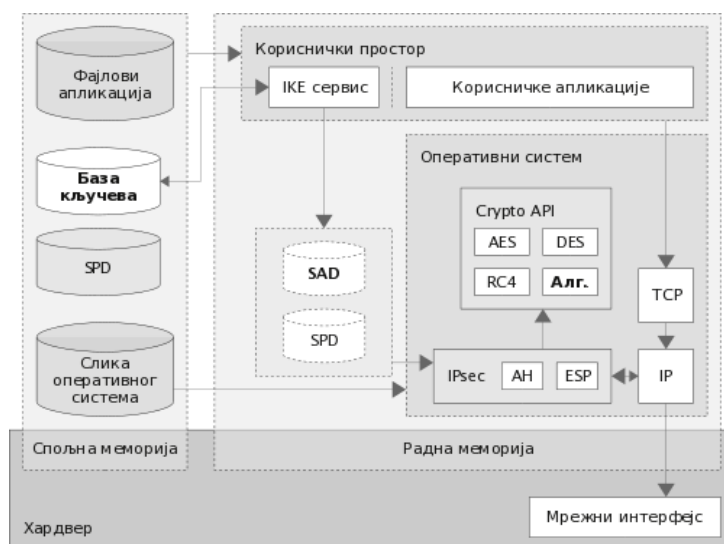
Први модел подразумева модификовану *IKEv2* имплементацију (слика 32) која укључује постојање базе симетричних кључева (размењене путем безбедних комуникационих канала). Кључ који је установљен преко *IKEv2* протокола користи се као основа за одређивање индекса тајног кључа у бази кључева који ће бити употребљен у шифарском алогиртму (слика 33). Сесијски кључ је јавно доступан. У предложеном решењу, тајни симетрични кључеви се генеришу путем независних уређаја који гарантују квалитет [36]. У складу са тим, кључеви нису само тајни већ и гарантованог криптолошког квалитета (у случају коришћења исправног генератора случајних бројева).



Слика 32. Измењена *IPsec/IKEv2* архитектура

Претпоставља се да у бази кључева постоји велики број кључева (на пример 2^{32}). Уколико би се користили кључеви дужине 1024 бита 32-битни индекс би омогућио

4.294.967.296 кључева у бази података, са укупном индивидуалном дужином од 1066 битова. Укупна величина базе кључева износила би 560 гигабајтова. У случају да се таква база кључева редовно користи у току једне године, 11.767.033 комуникације дневно могле би бити покривене јединственим кључем.



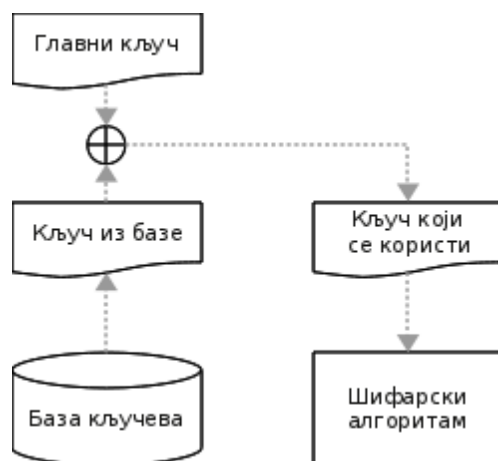
Слика 33. Модел система се претходно размењеном базом кључева

Криптоанализом предложеног решења долази се до закључка да се описаном изменом постиже отпорност система на *MITM* напад (што није подразумевана карактеристика основног *IKEv2* протокола). Додатно, омогућава се коришћење квалитетних кључева генерисаних на основу случајног извора. Међутим, уочава се и могућност (са релативно малом вероватноћом) да се за шифровање садржаја две одвојене комуникације употребе исти сесијски и тајни кључ. Из тог разлога систем треба проширити системом који онемогућава вишеструко коришћење истог тајног кључа из базе (на пример, означавањем искоришћених кључева или њиховим уклањањем). Уколико се утврди да кључ добијен *IKEv2* протоколом упућује на већ искоришћени кључ, целу процедуру треба вратити у иницијално стање (поново разменити кључ *IKEv2* протоколом).

Кроз анализу сценарија у коме нападач успе да дође у посед кључева уочава се потреба за увођењем додатних мера сигурности. Чак и непознавање алгоритма за добијање индекса кључа у бази на основу кључа добијеног *IKEv2* протоколом оставља нападачу могућност да разбије шифрат коришћењем познатог сесијског кључа и испробавањем свих кључева из базе података.

Следећи предложени модел проширује претходни кроз коришћење додатног, **главног**

кључа као додатка кључевима у бази кључева. Битска дужина главног кључа мора бити иста или већа од дужине кључева у бази. Овај кључ би поседовала само особа која се у одређеном периоду овлашћује за руковање системом. Кључ би се складиштио на одвојеном медију (нпр. *SmartCard*, *USB* меморија и сл.). Принцип коришћења главног кључа приказан је на слици 34.



Слика 34. Коришћење базе кључева са додатком главног кључа

Увођењем главног кључа расте сложеност целокупног система а уводи се и питање генерисања и дистрибуције тог кључа. Међутим, ова допуна уводи две главне предности:

1. онемогућава се коришћење система без присуства овлашћеног лица које поседује главни кључ и
2. проблем компромитовања базе кључева решава се заменом главног кључа.

Као подразумевана операција за спаривање кључа из базе и главног кључа наведена је искључива дисјункција. Она се може сматрати довољном јер нападач до главног кључа не може да дође познавањем кључа из (компромитоване) базе и шифрата. Наравно, наведена операција се у случају потребе може заменити неком сложенијом операцијом.

Трећи модел управљања кључевима подразумева коришћење главног кључа као тајног кључа за шифарски алгоритам, а коришћење кључева из базе података (добитених на основу индекса/кључева добијених *IKEv2* протоколом) као сесијских кључева (слика 35). Коришћењем оваквог модела нападач се ставља у позицију да му нису познати нити тајни, нити сесијски кључ. У случају да дође до компромитовања базе кључева систем се своди на модел у коме се сесијски кључ јавно преноси и подразумева се да је

нападач упознат са њим.



Слика 35. Модел са коришћењем базе кључева као сесијских кључева

У првом предложеном моделу подразумева се једноструко коришћење кључева из базе, односно шифровање сваке од сесија јединственим тајним кључем. У овом моделу постоји исто ограничење броја могућих комуникација уколико се подразумева коришћење јединственог сесијског кључа из базе кључева. Међутим, систем се може проширити коришћењем кључа из базе у комбинацији са делом претходно послатог шифрата. У том случају се мора строго водити рачуна о квалитету алгоритма којим се добија сесијски кључ с обзиром на то да се подразумева да су раније послати шифрати познат нападачу.

4. Експериментална анализа перформанси решења

Један од атрибута развијеног решења, критичан за употребу у пракси, јесу његове перформансе. О безбедносним аспектима и критеријумима решења дискутовано је у ранијим деловима рада. У овом делу рада даће се коментар перформанси решења у погледу постигнуте брзине шифровања и дешифровања података, односно коју је количину података, у којој јединици времена и на којој хардверској платформи могуће обрадити. Анализа перформанси решења базирана је на резултатима експеримента, чији је модел и структура такође дат у овом поглављу.

4.1. Циљеви експерименталне анализе

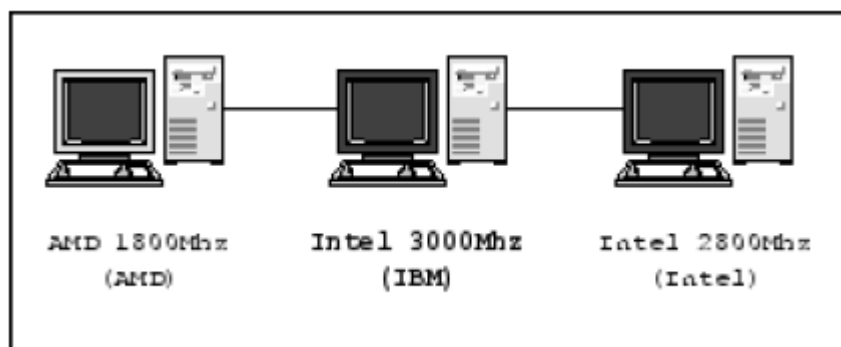
Као основни циљ експерименталне анализе перформанси развијеног система за заштиту комуникација постављено је утврђивање оптималног односа између перформанси процесора (који се користе за шифровање и дешифровање саобраћаја) и пропусне моћи комуникационих канала. С обзиром да се описано решење може користити и на комуникационим пролазима које користи велики број чланова интерне мреже, количина саобраћаја коју је потребно шифровати у јединици времена може бити веома велика. Такође, савремене комуникационе технологије нуде све веће брзине мрежног преноса података, а савремени сервиси и захтевају високе пропусне моћи комуникационих канала.

За основни циљ развоја постављен је виши ниво заштите информација у односу на постојеће шифарске алгоритме, а не боље перформансе по питању количине шифрованих информација у јединици времена. Међутим, у оквиру дате експерименталне анализе извршени су и тестови популарних *AES* и *3DES* алгоритама, а перформансе сопственог шифарског алгоритма упоређене су са измереним перформансама тих алгоритама. Додатно, поређење перформанси је дато и у односу на отворен пренос података, без коришћења шифарских алгоритама.

Важна напомена везана за поређење перформанси сопственог шифарског алгоритма са *AES* и *3DES* алгоритмима јесте та да су добијени слабији резултати сопственог алгоритма очекивани као последица веће рачунске сложености. Сама замена већ уграђених јавних алгоритама не изазива пад перформанси система.

4.2. Модел експерименталног окружења

За потребе вршења експерименталне анализе перформанси развијеног решења постављено је наменско експериментално окружење. Ово окружење је формирано путем 3 рачунара 32-битне *x86* архитектуре који су редно повезани *Ethernet* технологијом пропусне моћи 1Gb/s (слика 36).



Слика 36. Шема рачуарске мреже коришћене у експерименту

Битне карактеристике рачунара коришћених у експерименту дате су у табели 13:

AMD	
Централни процесор:	<i>AMD Sempron(TM) 2200+</i> <i>1800MHz</i> <i>256KB</i>
Радна меморија:	<i>256MB</i>
Мрежни интерфејс:	<i>Realtek RTL-8169 Gigabit Ethernet</i>
Intel	
Централни процесор:	<i>Intel Pentium D</i> <i>2,8GHz</i> <i>1024KB</i>
Радна меморија:	<i>4096MB</i>
Мрежни интерфејс:	<i>NetLink BCM5789 Gigabit Ethernet</i>
IBM	
Централни процесор:	<i>2 x Intel Xeon</i> <i>2 x 3GHz</i> <i>2 x 2048KB</i>
Радна меморија:	<i>4096MB</i>
Мрежни интерфејс:	<i>2 x NetXtreme BCM5721 Gigabit Ethernet</i>

Табела 13. Карактеристике рачунара коришћених у експерименту

На све рачунаре инсталиран је Линукс оперативни систем верзије 2.6.17.8 (Слеквер дистрибуција верзије 10.2). На *AMD* и *Intel* рачунарима укључена је подршка за *IPsec* протокол:

- *CONFIG_NET_KEY = Y*
- *CONFIG_INET_AH = Y*
- *CONFIG_INET_ESP = Y*

као и за одговарајуће шифарске алгоритме:

- *CONFIG_CRYPTOHMAC = Y*
- *CONFIG_CRYPTOMD5 = Y*
- *CONFIG_CRYPTOSHAI = Y*
- *CONFIG_CRYPTODES = Y*
- *CONFIG_CRYPTOAES = Y*
- *CONFIG_CRYPTOFBISCA = Y*

док је на *IBM* рачунару укључена подршка за функцију премошћавања:

- *CONFIG_BRIDGE = Y*

Од додатног софтвера на *AMD* и *Intel* рачунарима инсталиран је програм *netperf* који омогућава извршавање различитих мрежних тестова. Једна од главних особина овог програма је реализација на клијент-сервер архитектури. На тај начин је омогућено синхронизовано праћење ратличитих параметара на обе стране. Перформансе свих преноса податка датих у овом раду (отворених, шифрованих јавним алгоритмима и шифрованих сопственим алгоритмом) добијене су путем овог програма.

На *IBM* рачунару инсталиран је програм *ethereal* који омогућава снимање мрежног саобраћаја кроз рачунарски систем. Снимљени саобраћај се може накнадно анализирати. У експерименту је овај софтвер коришћен да би се проверило да ли се пренос врши у шифрованом облику, односно, да ли се мерење врши над исправним током података.

4.3. Структура експеримента

Након успостављања описаног експерименталног окружења извршено је мерење перформанси решења. Мерење је извршено са следећим карактеристикама:

- коришћен је десетоминутни *TCP* стрим, генерисан помоћу *netperf* софтвера,
- за сваки тип преноса (отворени пренос, шифровање *DES* алгоритмом, шифровање *AES* алгоритмом, шифровање сопственим алгоритмом) вршена су по три мерења,
- сваки тип преноса вршен је у оба смера,
- код свих шифрованих комуникација коришћени су и *AH* и *ESP* протоколи,
- за сва мерења је коришћена иста платформа, у истој конфигурацији.

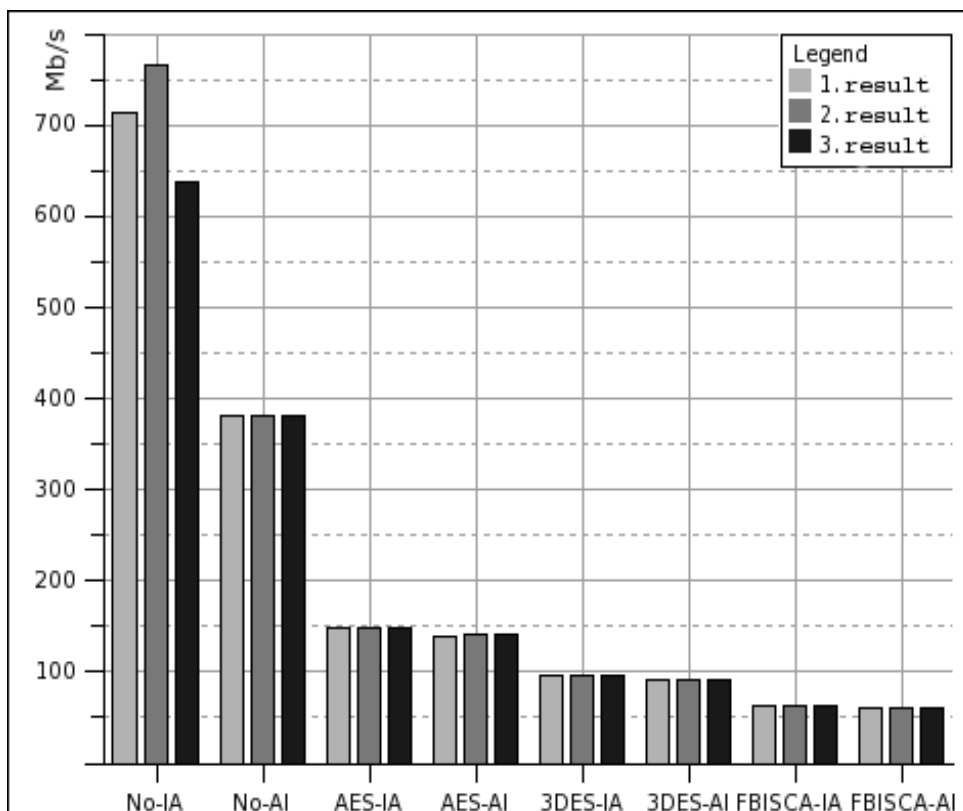
При сваком од мерења бележени су следећи параметри:

- остварена брзина протока података,
- оптерећење централног процесора пошиљаоца,
- оптерећење централног процесора примаоца.

Резултати описане експерименталне анализе дати су у наредном делу рада.

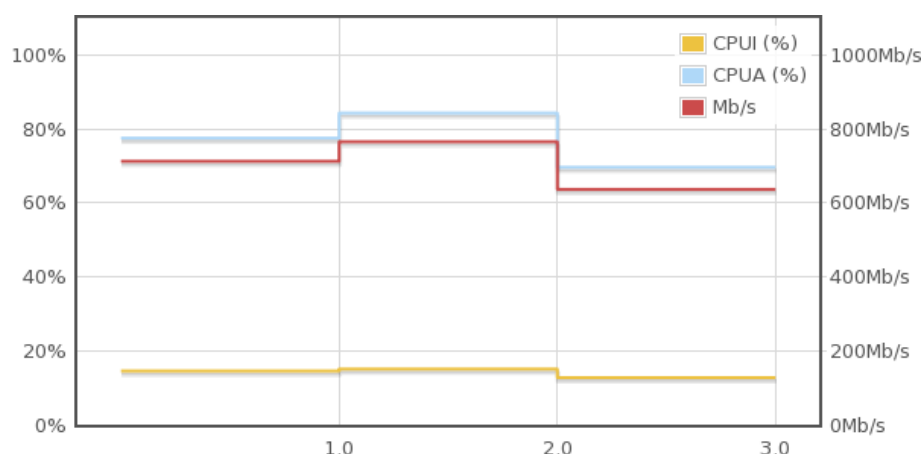
4.4. Резултати

Резултати мерења су приказани на слици 37 (дијаграм остварених брзина протока) и у табели 14 која садржи бројчане вредности.



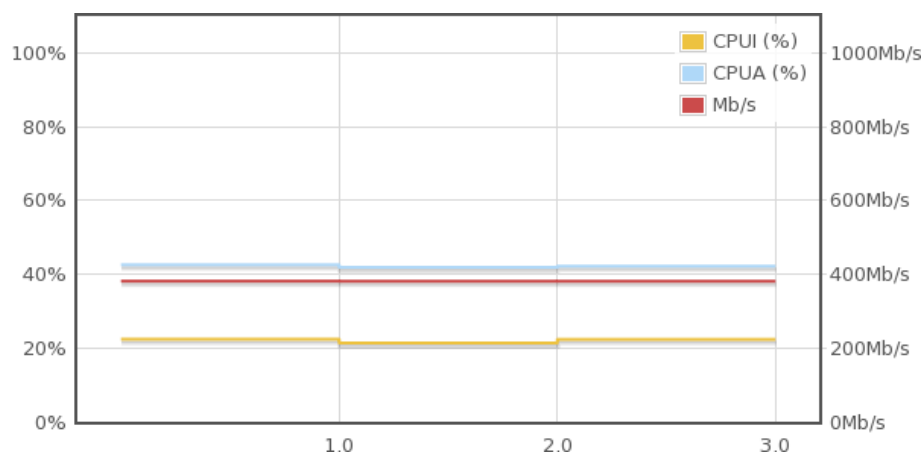
Слика 37. Дијаграм са оствареним брзинама протока по алгоритму

На слици 37 види се пад брзине преноса података при коришћењу захтевнијих алгоритама за шифровање. Имајући у виду да је за сва мерења коришћена иста хардверска платформа и исти комуникациони канал (исте пропусне моћи) јасно је да разлог успорења лежи у оптерећењу процесора учесника у комуникацији тј. њиховој могућности да у јединици времена шифрују/дешифрују одређену количину података.

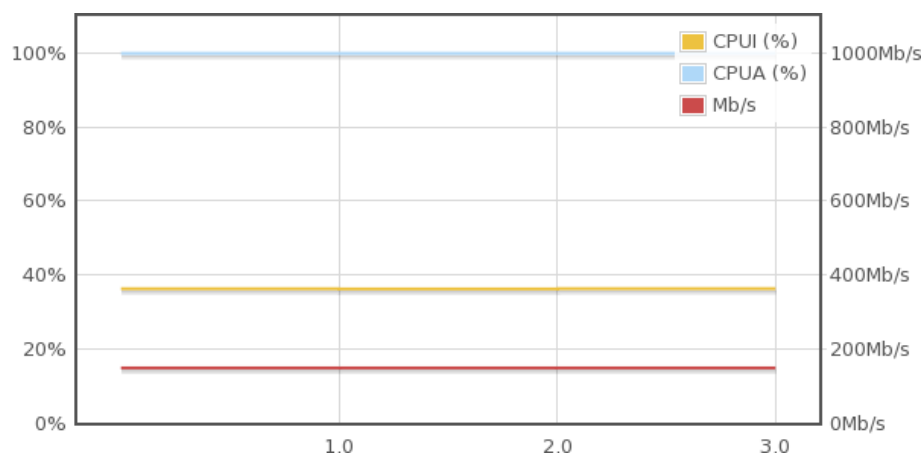
Слика 38. Перформансе *Intel->AMD*, без шифровања

На слици 38 приказани су резултати три мерења остварене брзине преноса података и оптерећења централних процесора комуникационих страна. При овом преносу рачунар

Intel имао је улогу генератора података и пошиљаоца, док је рачунар *AMD* имао улогу примаоца. Подаци су пренесени у отвореном облику, односно није коришћен ни један алгоритам за шифровање. У првом мерењу забележена је просечна остварена брзина преноса од $712,41\text{Mb/s}$, централни процесор пошиљаоца био је $14,52\%$ оптерећен, а централни процесор примаоца имао је оптерећење од $77,52$ процената. У другом мерењу забележена је просечна остварена брзина преноса од $765,25\text{Mb/s}$, централни процесор пошиљаоца био је $15,06\%$ оптерећен, а централни процесор примаоца имао је оптерећење од $84,31$ процената. У трећем мерењу забележена је просечна остварена брзина преноса од $636,59\text{Mb/s}$, централни процесор пошиљаоца био је $12,73\%$ оптерећен, а централни процесор примаоца имао је оптерећење од $69,63$ процената.

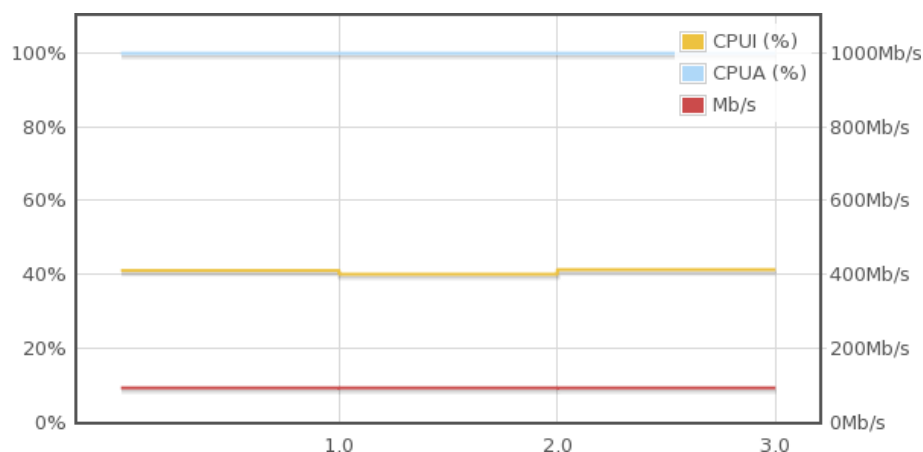
Слика 39. Перформансе *AMD->Intel*, без шифровања

На слици 39 приказани су резултати три мерења остварене брзине преноса података и оптерећења централних процесора комуникационих страна. При овом преносу рачунар *AMD* имао је улогу генератора података и пошиљаоца, док је рачунар *Intel* имао улогу примаоца. Подаци су пренешени у отвореном облику, односно није коришћен ни један алгоритам за шифровање. У првом мерењу забележена је просечна остварена брзина преноса од $381,11\text{Mb/s}$, централни процесор пошиљаоца био је $42,56\%$ оптерећен, а централни процесор примаоца имао је оптерећење од $22,38$ процената. У другом мерењу забележена је просечна остварена брзина преноса од $380,91\text{Mb/s}$, централни процесор пошиљаоца био је $41,86\%$ оптерећен, а централни процесор примаоца имао је оптерећење од $21,33$ процената. У трећем мерењу забележена је просечна остварена брзина преноса од $380,96\text{Mb/s}$, централни процесор пошиљаоца био је $42,17\%$ оптерећен, а централни процесор примаоца имао је оптерећење од $22,34$ процената.



Слика 40. Перформансе Intel->AMD, AES алгоритам

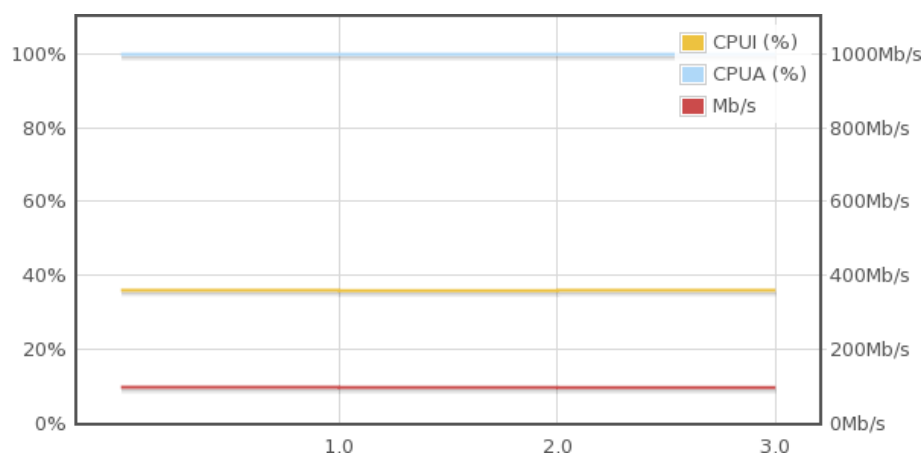
На слици 40 приказани су резултати три мерења остварене брзине преноса података и оптерећења централних процесора комуникационих страна. При овом преносу рачунар *Intel* имао је улогу генератора података и пошиљаоца, док је рачунар *AMD* имао улогу примаоца. При преносу података коришћен је *AES* алгоритам за шифровање. У првом мерењу забележена је просечна остварена брзина преноса од $147,99\text{Mb/s}$, централни процесор пошиљаоца био је $36,22\%$ оптерећен, а централни процесор примаоца имао је оптерећење од $99,9$ процената. У другом мерењу забележена је просечна остварена брзина преноса од $148,15\text{Mb/s}$, централни процесор пошиљаоца био је $36,18\%$ оптерећен, а централни процесор примаоца имао је оптерећење од $99,9$ процената. У трећем мерењу забележена је просечна остварена брзина преноса од $148,03\text{Mb/s}$, централни процесор пошиљаоца био је $36,25\%$ оптерећен, а централни процесор примаоца имао је оптерећење од $99,9$ процената.



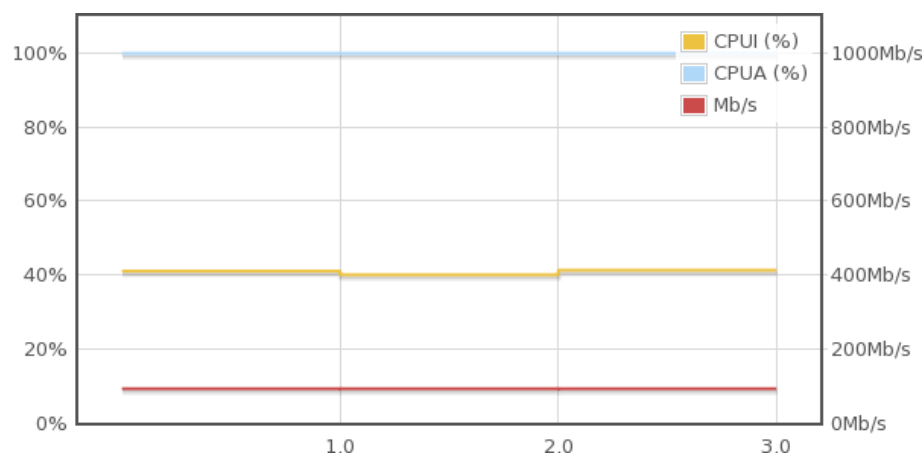
Слика 41. Перформансе AMD->Intel, AES алгоритам

На слици 41 приказани су резултати три мерења остварене брзине преноса података и

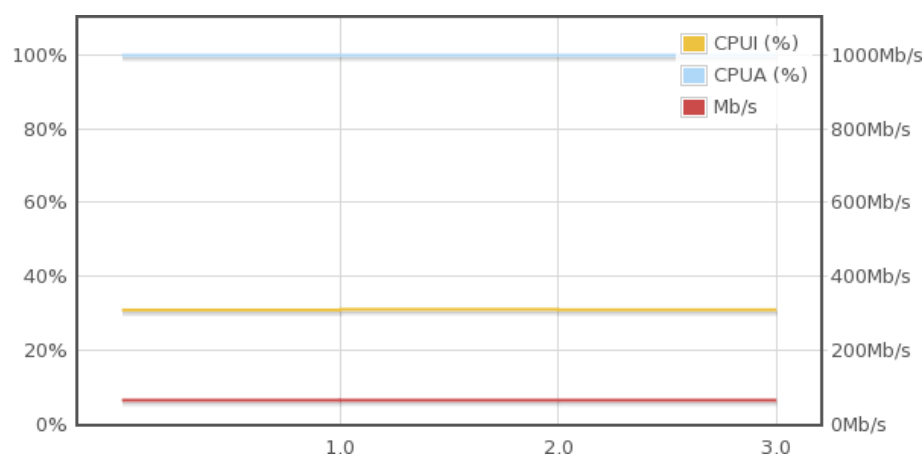
оптерећења централних процесора комуникационих страна. При овом преносу рачунар *AMD* имао је улогу генератора података и пошиљаоца, док је рачунар *Intel* имао улогу примаоца. При преносу података коришћен је *AES* алгоритам за шифровање. У првом мерењу забележена је просечна остварена брзина преноса од $140,05\text{Mb/s}$, централни процесор пошиљаоца био је 99,9% оптерећен, а централни процесор примаоца имао је оптерећење од 47,93 процената. У другом мерењу забележена је просечна остварена брзина преноса од $140,47\text{Mb/s}$, централни процесор пошиљаоца био је 99,9% оптерећен, а централни процесор примаоца имао је оптерећење од 47,89 процената. У трећем мерењу забележена је просечна остварена брзина преноса од $140,25\text{Mb/s}$, централни процесор пошиљаоца био је 99,9% оптерећен, а централни процесор примаоца имао је оптерећење од 47,82 процената.

Слика 42. Перформансе *Intel->AMD*, *3DES* алгоритам

На слици 42 приказани су резултати три мерења остварене брзине преноса података и оптерећења централних процесора комуникационих страна. При овом преносу рачунар *Intel* имао је улогу генератора података и пошиљаоца, док је рачунар *AMD* имао улогу примаоца. При преносу података коришћен је *3DES* алгоритам за шифровање. У првом мерењу забележена је просечна остварена брзина преноса од $96,25\text{Mb/s}$, централни процесор пошиљаоца био је 35,93% оптерећен, а централни процесор примаоца имао је оптерећење од 99,9 процената. У другом мерењу забележена је просечна остварена брзина преноса од $95,7\text{Mb/s}$, централни процесор пошиљаоца био је 35,82% оптерећен, а централни процесор примаоца имао је оптерећење од 99,9 процената. У трећем мерењу забележена је просечна остварена брзина преноса од $95,51\text{Mb/s}$, централни процесор пошиљаоца био је 35,92% оптерећен, а централни процесор примаоца имао је оптерећење од 99,9 процената.

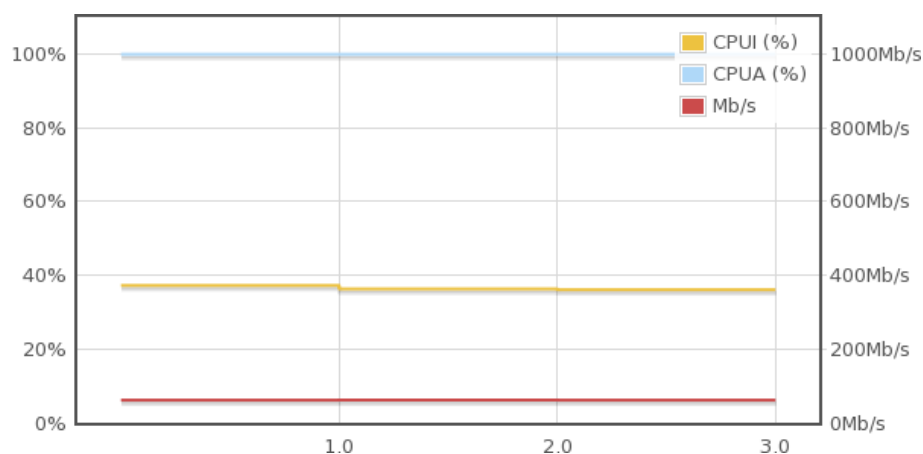
Слика 43. Перформансе *AMD->Intel*, *3DES* алгоритам

На слици 43 приказани су резултати три мерења остварене брзине преноса података и оптерећења централних процесора комуникационих страна. При овом преносу рачунар *AMD* имао је улогу генератора података и пошиљаоца, док је рачунар *Intel* имао улогу примаоца. При преносу података коришћен је *3DES* алгоритам за шифровање. У првом мерењу забележена је просечна остварена брзина преноса од $91,68\text{Mb/s}$, централни процесор пошиљаоца био је 99,9% оптерећен, а централни процесор примаоца имао је оптерећење од 40,99 процената. У другом мерењу забележена је просечна остварена брзина преноса од $91,55\text{Mb/s}$, централни процесор пошиљаоца био је 99,9% оптерећен, а централни процесор примаоца имао је оптерећење од 40,01 процената. У трећем мерењу забележена је просечна остварена брзина преноса од $91,57\text{Mb/s}$, централни процесор пошиљаоца био је 99,9% оптерећен, а централни процесор примаоца имао је оптерећење од 41,23 процента.

Слика 44. Перформансе *Intel->AMD*, *FBISCA* алгоритам

На слици 44 приказани су резултати три мерења остварене брзине преноса података и

оптерећења централних процесора комуникационих страна. При овом преносу рачунар *Intel* имао је улогу генератора података и пошиљаоца, док је рачунар *AMD* имао улогу примаоца. При преносу података коришћен је *FBISCA* алгоритам за шифровање. У првом мерењу забележена је просечна остварена брзина преноса од $63,72\text{Mb/s}$, централни процесор пошиљаоца био је $30,83\%$ оптерећен, а централни процесор примаоца имао је оптерећење од $99,9$ процената. У другом мерењу забележена је просечна остварена брзина преноса од $63,75\text{Mb/s}$, централни процесор пошиљаоца био је 31% оптерећен, а централни процесор примаоца имао је оптерећење од $99,9$ процената. У трећем мерењу забележена је просечна остварена брзина преноса од $63,74\text{Mb/s}$, централни процесор пошиљаоца био је $30,89\%$ оптерећен, а централни процесор примаоца имао је оптерећење од $99,9$ процената.



Слика 45. Перформансе *AMD->Intel*, *FBISCA* алгоритам

На слици 45 приказани су резултати три мерења остварене брзине преноса података и оптерећења централних процесора комуникационих страна. При овом преносу рачунар *AMD* имао је улогу генератора података и пошиљаоца, док је рачунар *Intel* имао улогу примаоца. При преносу података коришћен је *FBISCA* алгоритам за шифровање. У првом мерењу забележена је просечна остварена брзина преноса од $61,33\text{Mb/s}$, централни процесор пошиљаоца био је $99,9\%$ оптерећен, а централни процесор примаоца имао је оптерећење од $37,22$ процента. У другом мерењу забележена је просечна остварена брзина преноса од $61,44\text{Mb/s}$, централни процесор пошиљаоца био је $99,9\%$ оптерећен, а централни процесор примаоца имао је оптерећење од $36,28$ процената. У трећем мерењу забележена је просечна остварена брзина преноса од $61,4\text{Mb/s}$, централни процесор пошиљаоца био је $99,9\%$ оптерећен, а централни процесор примаоца имао је оптерећење од $36,07$ процената.

У табели 14 бројчано су дате вредности приказане на претходним дијаграмима. С обзиром на то да је оптерећење *AMD* процесора при коришћењу сваког од алгоритама било на максимуму ($99,9\% + 0,01\%$ за системске процесе) то јасно указује на уско грло ових комуникација. Такође, могуће је предвидети 2-3 пута већу брзину коришћењем снажнијег процесора какав је у *Intel* рачунару.

Остварена брзина (Mb/s)	CPU оптерећење пошиљаоца (%)	CPU оптерећење примаоца (%)
Без шифровања. (Intel -> AMD)		
712.41	14.52	77.52
765.25	15.06	84.31
636.59	12.73	69.63
Без шифровања (AMD -> Intel)		
381.11	42.56	22.38
380.91	41.86	21.33
380.96	42.17	22.34
AES (Intel -> AMD)		
147.99	36.22	99.90
148.15	36.18	99.90
148.03	36.25	99.90
AES (AMD -> Intel)		
140.05	99.90	47.93
140.47	99.90	47.89
140.25	99.90	47.82
3DES (Intel -> AMD)		
96.25	35.93	99.90
95.70	35.82	99.90
95.51	35.92	99.90
3DES (AMD -> Intel)		
91.68	99.90	40.99
91.55	99.90	40.01
91.57	99.90	41.23
FBISCA (Intel -> AMD)		
63.72	30.83	99.90
63.75	31.00	99.90
63.74	30.89	99.90
FBISCA (AMD -> Intel)		
61.33	99.90	37.22
61.44	99.90	36.28
61.40	99.90	36.07

Табела 14. Резултати мерења.

Напомена: У овом раду није вршено поређење могућности процесора компанија *Intel* и *AMD*. Скромније могућности *AMD* процесора су свакако очекиване јер је тај процесор знатно старији од коришћеног *Intel* процесора, ради на нижој фреквенцији и има мању количину кеш меморије. У експерименту су намерно коришћени системи различитих могућности да би се што боље проценило оптерећење шифровањем велике количине података у што краћем временском периоду.

4.5. Дискусија резултата

Представљени резултати експерименталне анализе перформанси предложеног система за заштићену комуникацију могу се разматрати са више аспеката. У оквиру критичке анализе треба размотрити забележене разлике у перформансама, употребљивост решења за практичну примену и коришћење евидентираних резултата у планирању хардверске платформе за одређен обим употребе.

Прво што се може приметити јесу знатно ниже перформансе сопственог алгоритма у односу на *AES* и *3DES* алгоритме. Међутим, претходно је већ истакнуто да је циљ овога рада развој система који нуди виши ниво безбедности у односу на јавно доступне системе заштите, а да је анализа перформанси и поређење са *AES* и *3DES* алгоритмима првенствено дата као оријентациона вредност у планирању хардверских ресурса за практичну примену решења у конкретном случају. Евидентиране разлике у перформансама су очекиване с обзиром на већу комплексност сопственог алгоритма у односу на алгоритме са којима је поређен. Такође, с обзиром да је остварен планиран ниво безбедности са употребљивим перформансама на актуелним хардверским ресурсима, остварени однос:

$$[\text{сложеност алгоритма}] - [\text{хардверска платформа}] - [\text{перформансе}]$$

може се кориговати у складу са конкретним могућностима и потребама. Првенствено, порастом могућности централних процесора може се додатно усложњавати шифарски алгоритам у циљу повећања нивоа безбедности.

Остварена брзина од око 60Mb/s на процесору старије генерације (*AMD* процесор у експерименталном окружењу), какав се данас може срести у интегрисаним системима са пасивним хлађењем, може се сматрати употребљивом у пракси. Кроз комуникациони канал те пропусне моћи може се истовремено преносити саобраћај више видео-конференција, сесија Интернет телефоније и сл. Са друге стране, резултати показују да би се коришћењем процесора новије генерације (са перформансама приближним *Intel* процесору у експерименталном окружењу) на обе комуникационе стране могла остварити брзина шифровања од 150Mb/s , што превазилази пропусну моћ већине комуникационих канала, чак и у локалним мрежама.

5. Области примене предложеног решења

Као основно технолошко и употребно окружење предложеног решења, у делу рада који се бави анализом инфраструктура погодних за интеграцију сопственог система заштите комуникације, наведена ја Интернет мрежа. У овом делу рада анализиране су могућности примене предложеног решења у другим областима и код других комуникационих инфраструктура. Анализиране су могућности примене у WAN и Интернет мрежама, у локалним рачунарским мрежама и у мобилној телефонији.

5.1. Примена решења на WAN и Интернет мрежама

Комуникациони канали WAN и Интернет мрежа подразумевано се сматрају небезбедним. Међутим, комуникациона инфраструктура ових мрежа омогућава једноставно и јефтино остваривање комуникације, чак и између комуникационих страна које се налазе на великим географским удаљеностима. У складу са тим, а како је и у ранијим деловима рада истицано, примена представљеног решења на WAN и Интернет мрежама не представља само могућу област примене, већ је и претпостављени циљ развоја.

С обзиром на реализацију решења на нивоу мрежног протокола, њиме се могу штитити комуникације WAN и Интернет каналима које користе различите транспортне и апликативне протоколе. У најзначајније апликативне протоколе који се могу штитити овим решењем спадају протоколи Интернет телефоније, видео-конференција, Веб апликације, електронска пошта, и т.д.. Скуп апликативних протокола, наравно, није ограничен на ове и остале протоколе.

5.2. Примена решења у локалној мрежи

Иако је основна претпоставка да су комуникациони канали локалних рачунарских мрежа безбедни (заштићени од прислушкивања), постоје одређене могућности за примену описаног решења и у њиховој комуникационој инфраструктури. Услед пораста потребе за коришћењем информационих технологија, као и пада цена рачунарских система и комуникационих компонената, данас се могу срести локалне рачунарске мреже са више хиљада корисника и мрежних чворова. У таквој ситуацији безбедност целокупног система подразумевано пада, односно, расте могућност инфилтрирања

нападача у комуникациони систем. Примена предложеног решења у описаној ситуацији може елиминисати, односно ублажити безбедносне претње на два начина:

1. нападач иако је оставрио приступ не може да прислушкује,
2. ко нема инсталиран систем не може да комуницира са остатком мреже

5.3. Примена решења у мобилној телефонији

Мобилна телефонија представља једно од циљних поља за заштиту комуникација, с обзиром на постојеће безбедносне проблеме, истакнуте раније у раду. За примену решења потребно је обезбедити компатибилну хардверско-софтверску платформу. Она подразумева присуство Линукс оперативног система као посредника између кориснички генерисаних података и хардверске инфраструктуре.

Једна од првих платформи мобилне телефоније базирана на Линуксу била је развијена у оквиру пројекта *OpenMoko*. Први телефон који је подржао ову платформу био је *Neo 1973* који је развила компанија *FIC*. Ова платформа била је намењена за коришћење у другој и трећој генерацији мобилне телефоније, а с обзиром на њено рано појављивање, прва разматрања везана за заштиту комуникација мобилне телефоније била су усмерена ка њој.

Једно од основних ограничења за коришћење развијеног система за заштиту комуникација на трећој и ранијим генерацијама мобилне телефоније јесте потреба за изменом шифарског алгорита и начина на који би се он користио. Међутим, објављивањем спецификације четврте генерације мобилне телефоније постало је јасно да се систем у њој може још ефикасније применити, и то у својој основној варијанти. Из тог разлога у овом делу рада разматрамо *Android* платформу мобилне телефоније као најперспективнију платформу за примену сопственог система заштите у четвртој генерацији мобилне телефоније.

Специфичност мобилне телефоније у односу на остала поља примене шифрованог преноса података огледа се у потреби да се у оквиру ограничених процесорских, меморијских и комуникационих ресурса изведе шифровање и дешифровање гласа у реалном времену. У противном, као последица кашњења може се јавити нестабилност везе или потпуна неупотребљивост сервиса.

5.3.1. Android платформа

Андроид платформа се најчешће доводи у везу са *Google* компанијом с тим да иза ње формално стоји *Open Handset Alliance*. Ова алијанса обухвата следеће чланове:

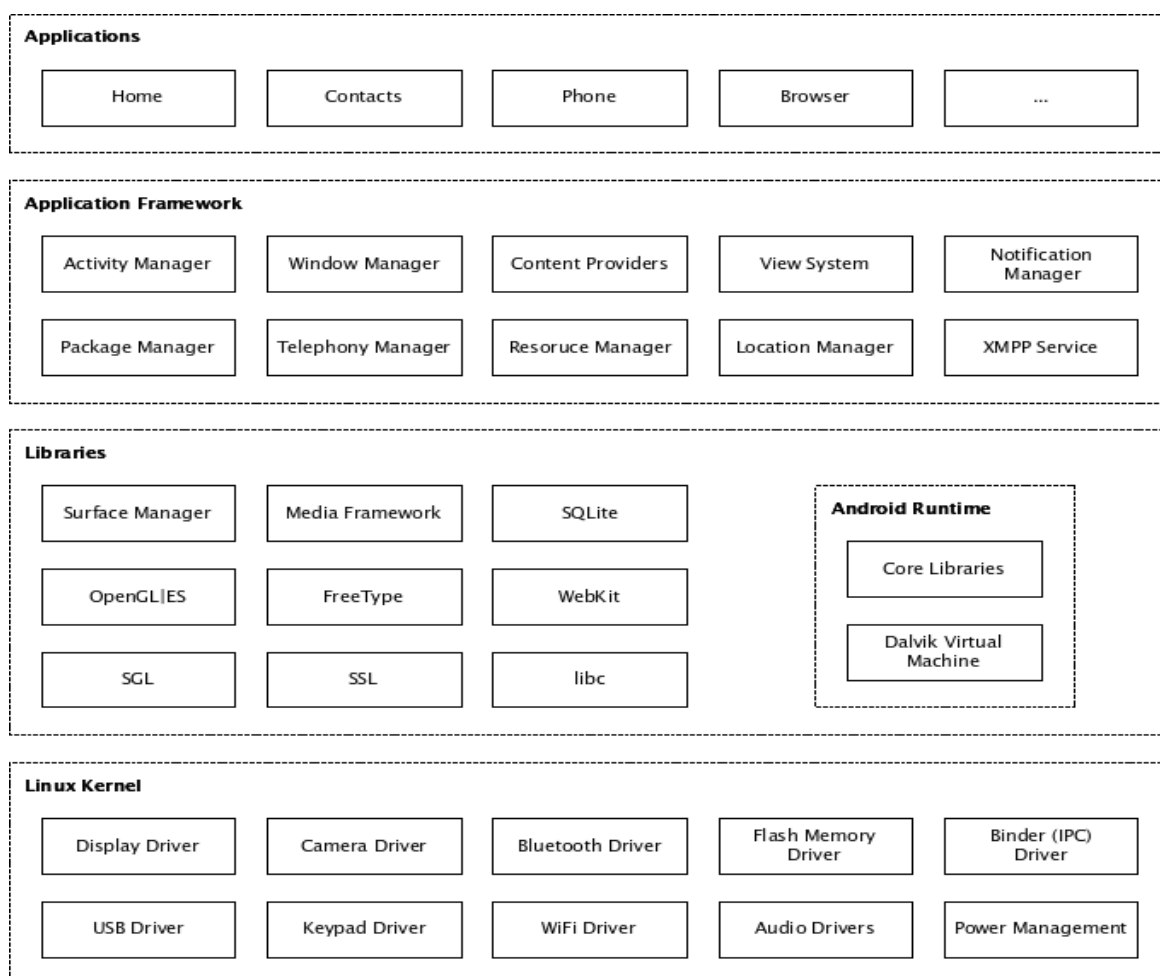
- Мобилни оператери: *Bouygues Telecom, China Mobile Communications, China Telecommunications Corporation, China United Network Communications, KDDI CORPORATION, NTT DOCOMO, INC., SOFTBANK MOBILE Corp., Sprint Nextel, T-Mobile, Telecom Italia, Telefónica, Vodafone*
- Произвођачи мобилних уређаја: *Acer Inc., Alcatel mobile phones, ASUSTeK Computer Inc., CCI, Dell, Foxconn International Holdings Limited, Garmin International, Inc., Haier Telecom (Qingdao) Co., Ltd., HTC Corporation, Huawei Technologies, Kyocera, Lenovo Mobile Communication Technology Ltd., LG Electronics, Motorola, Inc., NEC Corporation, Samsung Electronics, Sharp Corporation, Sony Ericsson, Toshiba Corporation, ZTE Corporation*
- Компаније за производњу полупроводника: *AKM Semiconductor Inc, Audience, ARM, Atheros Communications, Broadcom Corporation, CSR Plc., Cypress Semiconductor Corporation, Freescale Semiconductor, Gemalto, Intel Corporation, Marvell Semiconductor, Inc., MediaTek, Inc., MIPS Technologies, Inc., NVIDIA Corporation, Qualcomm Inc., Renesas Electronics Corporation, ST-Ericsson, Synaptics, Inc., Texas Instruments Incorporated, Via Telecom*
- Софтверске компаније: *ACCESS CO., LTD., Ascender Corp., Cooliris, Inc., eBay Inc., Google Inc., LivingImage LTD., Myriad, MOTOYA Co., Ltd., Nuance Communications, Inc., NXP Software, OMRON SOFTWARE Co, Ltd., PacketVideo (PV), SkyPop, SONiVOX, SVOX, VisualOn Inc.*
- Компаније за комерцијализацију: *Aplix Corporation, Borqs, L&T Infotech, Noser Engineering Inc., Sasken Communication Technologies Limited, SQLStar International Inc., TAT - The Astonishing Tribe AB, Teleca AB, Wind River, Wipro Technologies*

Почев од 21. октобра 2008. године изворни код *Android*-а доступан је јавности под *Apache 2.0* лиценцом. Ова лиценца омогућава појединцима и компанијама слободно коришћење платформе за развој сопствених решења. Такође, појединцима и

компанијама омогућено је и укључивање сопствених доприноса у званични пројекат након добијања адекватних лиценци (*Individual Contributor License Grant* и *Corporate Contributor License Grant* документи).

5.3.1.1. Основна безбедносна архитектура

Примарне безбедносне карактеристике Линукс оперативног система (систем привилегија базиран на јединственим идентификационим бројевима корисничких налога, група и процеса) искоришћене су и као носилац основних компонената безбедносне архитектуре *Android* платформе. Наиме, инсталациони систем за сваку од софтверских компонената (службени назив је *Android* пакети који се дистрибуирају у виду фајлова са екстензијом *.apk*) креира нови кориснички налог и на тај начин атомизује област додељених привилегија на ниво апликације, односно, власничког налога процеса и фајлова.



Слика 46. Архитектура *Android* платформе

У случају да је потребна међупроцесна комуникација процеса различитих апликација, таква комуникација је омогућена постављањем заједничког идентификационог броја налога за тај скуп апликација. У противном, било каква комуникација између процеса апликација (или приступ системским фајловима или фајловима осталих апликација) онемогућена је. Ограничења приступа су активна и на свим новокреираним фајловима апликације (путем припадности корисничком налогу апликације) док год апликација експлицитно не одобри јавни приступ сопственим фајловима (појединачно за сваки фајл са одвојеним режимима за читање и измену).

Аутори *Android* платформе начинили су и занимљиву листу карактеристика на основу којих би одређена софтверска компонента (апликација) на платформи могла бити окарактерисана као малициозни софтвер:

1. убрзано празни батерију уређаја,
2. кориснику приказује наметљиве поруке (посебно поруке које корисника упућују на одређену куповину),
3. одбија (или покушава да одбије) покушаје корисника да је уклони са система,
4. аутоматски покушава да се инсталира и на осталим уређајима мрежног окружења,
5. скрива своје фајлове и процесе,
6. прослеђује приватне податке трећој страни без знања или сагласности корисника,
7. уништава податке корисника, софтверске или хардверске компоненте без знања или сагласности корисника,
8. обавља одређене акције (слање порука, акције на Web-у и сл.) у име корисника без његовог знања и сагласности;

Иако ставке ове листе немају директних веза са тезом која се обрађује у овом раду (интеграцијом сопствених безбедносних система) оне су веома значајне јер обезбеђују стабилност, доступност и, у одређеној мери, безбедност окружења у коме егзистира сопствени криптолошки систем за заштиту комуникације.

5.3.1.2. Откривени безбедносни пропусти

Само неколико недеља након објављивања платформе (тачније, 6. новембра 2008. године) откривен је веома озбиљан безбедносни пропуст. Пропуст се манифестује у виду прослеђивања свих откуцаних карактера командној линији и њихово извршавање са администраторским (root) привилегијама. На пример, уколико се у било ком тренутку (при додељивању имена објекту, куцању SMS поруке, итд) јави следећи низ карактера:

enter – r – e – b – o – o – t – enter

оперативни систем телефона ће се рестартовати. Иако само рестартовање телефона не представља значајан проблем, јасно је да се овакав пропуст може искористити за потпуну манипулацију системским и корисничким фајловима и процесима а самим тим и за оштећивање софтвера телефона, инсталацију малициозног софтвера и сл. Као технички узрок проблема наводи се погрешна конфигурација `init.rc` фајла, највероватније као последица конфигурације која се користила при развоју платформе и отклањању грешака.

Пропуст је присутан у свим верзијама платформе закључно са верзијом 1.0-TC4-RC29. У верзији RC30 овај пропуст је отклоњен. Пропуст је иницијално објављен на форуму развојног тима платформе али се врло брзо појавио и на другим сајтовима који се баве сродним областима и темама:

1. <http://www.macworld.com/article/136596/2008/11/jailbreak.html>
2. <http://blogs.zdnet.com/Burnette/?p=676>
3. <http://blogs.zdnet.com/Burnette/?p=680>

Комплетна упутства како се овај пропуст може искористити за добијање администраторских привилегија налазе се на следећим адресама:

1. *How to get Root on the G1*
<http://forum.xda-developers.com/showthread.php?t=442480>
2. *ROOT FOUND!! How to get root on your G1*
http://android-dls.com/forum/index.php?f=15&t=151&rb_v=viewtopic
3. *Easiest way to root shell*
<http://android.jim.sh/index.php/ConsoleShell>

4. *Run commands on G1 without 3rd party app*

<http://forum.xda-developers.com/showthread.php?p=2870041>

Без обзира на ниво угрожености корисника и телефона који се може јавити као последица оваквих пропуста, чињеница је да је овај пропуст готово безопасан јер до масовније дистрибуције телефона (базираних на *Android* платформи) крајњим корисницима још увек није дошло. Такође, за отклањање поменутог пропуста развојном тиму платформе било је потребно свега неколико сати. Из тих разлога може се закључити да разлог за указану медијску пажњу пре свега лежи у значају компаније Google, као и тренутно најјачем конкуренту ове платформе – *Apple iPhone* платформи. За тезе које се обрађују у овом раду далеко је значајнија чињеница да је доступност изворног кода платформе јавности довела до тога да се пропуст у најкраћем могућем року открије, детаљно разради, утврде спорне компоненте и понуди решење.

5.3.1.3. Интеграција сопственог шифарског алгоритма

Интеграција сопственог система за заштиту података у инфраструктуру четврте генерације мобилне телефоније, коришћењем платформи базираних на Линукс оперативном систему, подразумева исти процес као и када је у питању стандардна рачунарска платформа. Детаљи процеса такве интеграције дати су у претходном делу рада.

Треба напоменути да су сва разматрања на тему интеграције система заштите у инфраструктуру мобилне телефоније чисто теоријског карактера, односно, да нема никаквих практична искуства, нити резултата експеримената коју су на овом пољу вршени. Међутим, разматрања су дата због значаја обезбеђивања професионалног система заштите на пољу стандардних мобилних комуникација. Додатно, описане значајне карактеристике четврте генерације мобилне телефоније обезбеђују такву компатибилност и указују на могућност примене решења описаног у овом раду.

6. Закључак

У овом поглављу сумирани су резултати рада на интеграцији сопственог криптолошког система у стандардну рачунарско-телекомуникациону инфраструктуру. Резултати су дати у виду остварених резултата и доприноса. Додатно, наведена су и поља у којима се даљим истраживањем и развојем могу остварити додатни резултати и доприноси.

6.1. Сумирање циљева истраживања

На почетку овог истраживања као његов основни циљ постављен је развој сопственог система заштите који се може користити на јавној рачунарско-телекомуникационој инфраструктури. Један од мотива за постављање оваквог циља био је утисак (у круговима који се професионално баве безбедношћу у телекомуникацијама) да су одређени, јавно доступни системи и компоненте достигли ниво стабилности и безбедности неопходан за њихово коришћење у развоју професионалних система за заштиту комуникације. Са друге стране, пораст комплексности савремених комуникационих инфраструктура, као и потреба за преносом све већих количина података, чине развој сопствених система за заштиту изузетно сложеним и захтевним, посебно уколико би се све њихове компоненте развијале самостално. За успешно остваривање постављеног циља било је неопходно обезбедити потпуну контролу над свим компонентама система, као и тајност самостално реализованих криптолошких компонената. На основу постигнутих резултата и остварених доприноса може се закључити да је основни циљ истраживања у потпуности остварен, односно да остварени резултати и доприноси чине адекватну основу за потврђивање основне хипотезе.

6.2. Остварени резултати и допринос

У овом раду је представљено практично реализовано сопствено решење за заштиту комуникација путем стандардне јавне рачунарско-телекомуникационе инфраструктуре, које користи самостално развијени шифарски алгоритам. Овим радом је потврђено да се комплексан, скуп и неизван сопствени развој сигурног комуникационог окружења може постићи са знатно мање напора и средстава адекватним коришћењем јавно доступних технологија. Такав резултат је од посебног значаја за технолошки

неразвијене и сиромашне средине као што је Србија. Основне доприносе овог рада на пољу безбедности у телекомуникацијама чини следеће:

1. Теоријски и практично је доказана основна хипотеза рада да се сопствени криптолошки системи могу користити на савременим јавним рачунарско-телекомуникационим инфраструктурама. За остваривање тезе коришћена је стандардна комуникациона инфраструктура, као и постојеће отворене хардверске и софтверске компоненте. Потврђивање хипотезе заснива се на научним и технолошким сазнањима и достигнућима из области телекомуникација (првенствено рачунарских мрежа) и криптологије. Егзистенцијални доказ се заснива на практичној реализацији решења базираног на постигнутим сазнањима и осталим елементима теоријског потврђивања хипотезе.
2. Извршена је анализа погодних јавно доступних компонената за развој сопственог система за заштиту комуникација. Поред јавно доступних хардверских и софтверских компонената анализиране су и популарне јавне телекомуникационе инфраструктуре: јавна телефонска мрежа, Интернет мрежа и мобилна телефонија. Иако резултати ове анализе чине једну од основа целокупног рада, њихова примена се ту не завршава већ се они могу користити и у другим теоријским и практичним истраживањима.
3. Развијен је систем који путем коришћења сопственог алгоритма нуди виши ниво заштите у односу на стандардне алгоритме (AES, 3DES, и сл.). Примарни допринос овог резултата огледа се у могућности практичне заштите података на основу коришћења шифарских кључева вишестурко веће битске дужине. Такав приступ чини шифрате, добијене овим системом, практично имуним на разбијање применом „сирове“ силе. Додатну отежавајућу околност за нападача чини и непознавање принципа рада шифарског алгоритма.
4. Предложене су три конкретне процедуре за управљање кључевима, структуриране по нивоима сигурности. У раду је извршена модификација шифарских кључева из IKEv2 протокола. Размотрени су следећи концепти: примена базе кључева гарантованог криптолошког квалитета (кључеви генерисани генератором случајних бројева), комбиновање са мастер кључем и комбиновање са

секвенцама из заосталог шифрата. Додано, представљен је и модел коришћења јединственог кључа за сваку сесију без опасности од MITM напада. Коришћењем ових процедура у пракси може се подићи ниво заштите који систем нуди.

5. Дата је анализа трансфера предложеног решења у технолошко окружење четврте генерације мобилне телефоније. На основу јавно доступне спецификације четврте генерације мобилне телефоније, а првенствено на основу њене заснованости на шестој генерацији Интернет протокола, потврђена је компатибилност предложеног решења са наредном генерацијом стандарда у мобилној телефонији као једној од најраспрострањенијих јавних комуникационих инфраструктура.

6.3. Даљи правци рада

Решење представљено у овом раду у потпуности је одговорило на примарно постављени задатак – развити сопствени систем заштите података који се преносе незаштићеним јавним комуникационим каналима заснованим на стандардним и јавно доступним телекомуникационим технологијама. Решење је засновано на употреби сопственог шифарског алгорита са циљем потпуне контроле над дужином кључева и процесима шифровања и дешифровања података.

У току рада су уочене следеће могућности и правци даљег развоја решења:

1. Развој интегрисаног хардверско-софтверског решења,
2. Хардверска реализација шифарског алгорита,
3. Анализа погодних медија за пренос и складиштење кључева.

6.3.1. Развој интегрисаног хардверско-софтверског решења

Једна од главних карактеристика решења представљеног у овом делу рада јесте могућност његове интеграције у стандардну телекомуникациону инфраструктуру и системе базиране на Линукс оперативном систему. У циљу очувања могућности интеграције решења у постојеће комуникационе системе базиране на Линукс оперативном систему није пожељно радити на даљем осамостаљивању решења, односно, на развоју комплетног интегрисаног хардверско-софтверског решења.

Међутим, за одређене конкретне потребе, за један од даљих праваца развоја могао би се узети развој таквог решења.

За развој интегрисаног хардверско-софтверског решења искористиле би се софтверске компоненте описане у овом раду – Линукс оперативни систем са интегрисаном IPsec имплементацијом и систем за управљање кључевима. Основна идеја, везана за хардверску платформу таквог решења, јесте искоришћавање неког од индустријских рачунара базираних на некој од подржаних архитектура. Основне карактеристике индустријских рачунара јесу мале димензије, робусност, мала потрошња електричне енергије и дуг животног века.

Подразумева се присуство два мрежна интерфејса на рачунару – један ка интерној и један ка спољној мрежи. Избор процесора – односно његове брзине – треба извршити у складу са датом анализом перформанси решења и очекиваном количином саобраћаја у јединици времена. Подразумева се да коришћена хардверска платформа испуни раније наведене критеријуме за остваривање „чистог“ рачунарског окружења.

6.3.2. Хардверска реализација шифарског алгорита

Перформансе савремених рачунарских процесора могу задовољити одређене захтеве у комбинацији са фреквенцијом саобраћаја и комплексношћу алгорита. Међутим, део рада који се бави анализом перформанси решења показује да пораст комплексности шифарског алгорита може драстично утицати на перформансе система. У складу са тим, као један од даљих праваца развоја решења може се узети делимична или потпуна хардверска реализација шифарског алгорита.

Хардверска реализација шифарског алгорита омогућила би употребу хардвера прилагођеног одређеним карактеристикама шифарског алгорита (дужини кључа, регистара и сл.). Повезивање хардверски реализованог шифарског алгорита са рачунаром који обавља шифровање мрежног саобраћаја могло би се обавити путем стандардних магистрала – *PCI*, *USB* и сл.

Софтверско повезивање *IPsec* имплементације у језгру оперативног система могуће је одрадити путем интеграције сопственог алгорита у оквиру стандардне криптографске секције језгра, али уз замену функционалности шифровања функционалностима повезивања са хардверским уређајем за шифровање. Додатни атрибут решења, на који

треба обратити пажњу, јесте и брзина магистрале којом се хардверски уређај за шифровање повезује са магистралом рачунара. У противном, брзина коришћене магистрале би могла да представља ново „уско грло“ система. Реализација сопственог алгоритма на нивоу језгра оперативног система омогућава релативно лаку миграцију на хардверску реализацију алгоритма.

6.3.3. Анализа погодних медија за пренос и складиштење кључева

Као следећи од даљих праваца развоја може се узети омогућавање додатног нивоа безбедности система путем увођења заштићених медија за пренос и складиштење података. Основни задатак ових медија јесте да заштити систем од компромитовања, по могућству и у ситуацијама код којих нападачи успеју да остваре физички приступ уређајима (губитак, крађа и сл.).

Као основна асоцијација на овом пољу јављају се *SmartCard* картице за складиштење и рад са подацима. Главна предност ових картица јесте могућност интерне обраде података. Коришћењем *SmartCard* картица могло би се омогућити да кључеви никада не напусте картице, или, да не напусте картице ван периода у којем је њихова употреба одобрена.

7. Литература

- [1] Biryukov, Alex; Adi Shamir; David Wagner. "Real Time Cryptanalysis of A5/1 on a PC". Fast Software Encryption—FSE 2000: 1–18.
- [2] Golic, Jovan Dj. (1997). "Cryptanalysis of Alleged A5 Stream Cipher". EUROCRYPT 1997: 239–55
- [3] Biham, Eli; Orr Dunkelman (2000). "Cryptanalysis of the A5/1 GSM Stream Cipher". Indocrypt 2000: 43–51
- [4] Ekdahl, Patrik; Thomas Johansson (2003). ("Another attack on A5/1". IEEE Transactions on Information Theory 49 (1): 284–89
- [5] Kevin J. O'Brien (28 December 2009). "Cellphone Encryption Code Is Divulged". New York Times. <http://www.nytimes.com/2009/12/29/technology/29hack.html>.
- [6] Young Kyun, Kim; Prasad, Ramjee. 4G Roadmap and Emerging Communication Technologies. Artech House 2006. pp. 12–13. ISBN 1-58053-931-9.
- [7] "4G - Beyond 2.5G and 3G Wireless Networks". MobileInfo.com. <http://www.mobileinfo.com/3G/4GVision&Technologies.htm>. Retrieved on 2007-03-26.
- [8] Jawad Ibrahim (December 2002). "4G Features" (PDF). Bechtel Telecommunications Technical Journal. http://www.bechteltelecoms.com/docs/bttj_v1/Article2.pdf. Retrieved on 2007-03-26.
- [9] "Mobility Management Challenges and Issues in 4G Heterogeneous Networks". ACM Proceedings of the first international conference on Integrated internet ad hoc and sensor networks. May 30 - 31, 2006.
- [10] Werner Mohr (2002). "Mobile Communications Beyond 3G in the Global Context" (PDF). Siemens mobile. http://www.cu.ipv6tf.org/pdf/werner_mohr.pdf.
- [11] Noah Schmitz (March 2005). "The Path To 4G Will Take Many Turns". Wireless Systems Design. <http://www.wsdmag.com/Articles/ArticleID/10001/10001.html>.
- [12] Carlos Frederico Cid, „Cryptanalysis of RSA: A Survey“ SANS

- [13] <http://www.keylength.com/en/8/>
- [14] Brute force attacks on cryptographic keys — a survey by Richard Clayton,
<http://www.cl.cam.ac.uk/~rnc1/brute.html>
- [15] Gueneysu, Tim; Timo Kasper; Martin Novotný; Christof Paar; Andy Rupp (2008).
"Cryptanalysis with COPACOBANA". Transactions on Computers Nov. 2008 57:
1498–1513.
- [16] http://www.nytimes.com/2008/08/13/technology/13cyber.html?_r=1&em
- [17] <http://datatracker.ietf.org/wg/tls/>
- [18] <http://www.gnu.org/software/gnutls/>
- [19] <http://www.openssl.org/>
- [20] <http://www.mozilla.org/projects/security/pki/nss/>
- [21] <http://www.yassl.com/yaSSL/Home.html>
- [22] Kent, S., and R. Atkinson, "Security Architecture for the Internet Protocol", RFC
2401, November 1998.
- [23] Kent, S., and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406,
November 1998.
- [24] Steven M. Bellovin, "Problem Areas for the IP Security Protocols", Proceedings of
the Sixth Usenix Unix Security Symposium, July, 1996
- [25] <http://www.kame.net/>
- [26] <http://ipsec-tools.sourceforge.net/>
- [27] W. Diffie and M. Hellman. New Directions in Cryptography. IEEE Trans. Inform.
Theory IT-22 (1976), 644-654
- [28] Federal Information Processing Standards Publication 197, Announcing the
Advanced Encryption Standard (AES)
- [29] Federal Information Processing Standards Publication 46-3, Data Encryption
Standard (DES)

- [30] Arif Mohamed, „Security trends for 2009“, Arif Mohamed, Computer Weekly, January 2009.
- [31] Bob Sullivan, „Study: ID theft usually an inside job“, MSNBC, January 2004.
- [32] Matt Bishop, Deborah A. Frincke, „Combating the Insider Cyber Threat“, IEEE Securiry & Privacy, January 2008.
- [33] „E-Crime Watch Survey“, <http://www.cert.org/archive/pdf/2004eCrimeWatch>
- [34] TCP/IP and Linux Protocol Implementation, Jon Crowcroft, Iain Philips, Wiley Computer Pulbishing, 2002.
- [35] <http://www.coreboot.org/History>
- [36] Markus Dichtl, Norbert Janssen, „A High Quality Physical Random Number Generator“, Eurosmart Security Conference Proceedings, 2000.

7.1. RFC документи

- RFC 2246 - The TLS Protocol Version 1.0
- RFC 2712 - Addition of Kerberos Cipher Suites to Transport Layer Security (TLS)
- RFC 2817 - Upgrading to TLS Within HTTP/1.1
- RFC 2818 - HTTP Over TLS
- RFC 3268 - Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)
- RFC 3546 - Transport Layer Security (TLS) Extensions
- RFC 3749 - Transport Layer Security Protocol Compression Methods
- RFC 4132 - Addition of Camellia Cipher Suites to Transport Layer Security (TLS)
- RFC 4279 - Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)
- RFC 4346 - The Transport Layer Security (TLS) Protocol Version 1.1
- RFC 4366 - Transport Layer Security (TLS) Extensions
- RFC 4492 - Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)

RFC 4785 - Pre-Shared Key (PSK) Ciphersuites with NULL Encryption for Transport Layer Security (TLS)

RFC 4880 - OpenPGP Message Format

RFC 5054 - Using the Secure Remote Password (SRP) Protocol for TLS Authentication

RFC 5081 - Using OpenPGP Keys for Transport Layer Security (TLS) Authentication

RFC 5246 - The Transport Layer Security (TLS) Protocol Version 1.2

RFC 5288 - AES Galois Counter Mode (GCM) Cipher Suites for TLS

RFC 5289 - TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM)

RFC 5469 - DES and IDEA Cipher Suites for Transport Layer Security (TLS)

RFC 5487 - Pre-Shared Key Cipher Suites for TLS with SHA-256/384 and AES Galois Counter Mode

RFC 5489 - ECDHE_PSK Cipher Suites for Transport Layer Security (TLS)

RFC 5705 - Keying Material Exporters for Transport Layer Security (TLS)

RFC 5746 - Transport Layer Security (TLS) Renegotiation Indication Extension

7.2. Објављени радови у научним часописима и конференцијама

1. Veinović M., Jevremović A., Šimić G., „Analysis and implementation of custom cipher algorithm for IPsec under Linux OS“, International Journal of Computer Science and Network Security, Vol. 8 No. 7, July 2008;
2. Јевремовић А., Веиновић М., „Заштита података на IP нивоу под Линукс ОС“, Зборник радова 50. конференције Етран, Београд, 2005., стр. 114-117;
3. Јевремовић А., Веиновић М., „IPsec - анализа утицаја алгорита за шифровање на саобраћај у LAN мрежама“, 14. телекомуникациони форум Телфор 2006., Београд, 2006., CD издање;

4. Јевремовић А., Веиновић М., Шимић Г., „Примена сопственог шифарског алгоритма за заштиту AJAX позива код Web апликација“, Зборник радова 52. конференције за електронику, телекомуникације, рачунарство, аутоматику и нуклеарну технику - Етран, Београд, 2008., CD издање;
5. Веиновић М., Јевремовић А., Шимић Г., „Имплементација сопственог алгоритма заштите у IPsec-у под Линукс оперативним системом“, Сингидунум ревија, Vol 5 No 1, Београд, 2008., стр. 92-102;
6. Веиновић М., Јевремовић А., Шимић Г. „Модел примене сопствених криптографских и стеганографских алгоритама на примеру Web галерије слика“, 16. телекомуникациони форум Телфор 2008., Београд, 2008., CD издање;
7. Веиновић М., Јевремовић А., Шимић Г. „Утицај формата носећих података на ефикасност стеганографских метода“, I научни скуп Синергија 2009, Универзитет Синергија, Бијељина, 2009.;
8. Јевремовић А., Веиновић М., Шимић Г., „Модификација IKEv2 протокола у циљу избора радног тајног кључа симетричних шифарских система“, Зборник радова 53. конференције за електронику, телекомуникације, рачунарство, аутоматику и нуклеарну технику - Етран, Београд, 2009., CD издање;
9. Јевремовић А., Веиновић М., Шимић Г., „Заштита бежичних комуникација коришћењем сопственог шифарског алгоритма“, 17. телекомуникациони форум Телфор 2009., Београд, 2009., CD издање;