ФМИ - СУ “Св. Климент Охридски”

Катедра “Информационни технологии”

Тема: VPN достъп до мрежата на СУ SUnet за индивидуални потребители и отдалечени звена

Дипломант: Цветомир Милчев Христов, Ф.№ 21110, специалност: ИКТ

Ръководител: доц. Красен Стефанов,ФМИ, Катедра “Информационни технологии”

Гр.София 15.07.2006 г.

Съдържание

Увод

1. Преглед на технологиите за VPN.

    1.1. Определение за Виртуална частна мрежа (VPN)

    1.2. Използване на VPN

    1.3. Основни изисквания към VPN

    1.4. Методи за криптиране, използвани при VPN

    1.5. Методи за удостоверяване, използвани при VPN

    1.6. Атаки върху мрежовата сигурност при VPN

    1.7. Практически решения за VPN

2. Описание на протокола IPSec.

    2.1. Общо описание и основни термини, свързани с IPSec.

    2.2. Протокол Encapsulating Security Payload (ESP)

    2.3. Протокол Authentication Header (AH)

    2.4. Протокол Internet Key Exchange (IKE)

3. Описание и конфигуриране на Openswan VPN сървър и клиент.

    3.1. Конфигурационен файл ipsec.conf. Описание на параметрите на Openswan

    3.2. Конфигурационен файл ipsec.secrets.

4. Реализация на достъпа на астрономическата обсерватория на SUNet през ADSL мрежата на БТК. Проблеми и възможности за включване на други звена.

    4.1. Инсталиране на Openswan под Линукс

    4.2. Конфигуриране на Openswan на Линукс сървъра vpn.uni-sofia.bg

    4.3. Конфигуриране на L2TP и PPP на сървъра vpn.uni-sofia.bg

    4.4. Конфигуриране на Openswan на отдалечени Линукс клиенти

    4.5. Конфигуриране на Windows XP клиент за свързване към Openswan по протокола l2tp/ipsec

Заключение

Използвана литература

Увод

Масовото разпространение на Интернет и непрестанното увеличение на скоростта му променя телекомуникациите и общуването между хората, организациите и държавите. Виртуалните частни мрежи (Virtual Private Network - VPN) са един етап от развитието на Интернет и се характеризират с предоставянето на сигурни комуникации между цели мрежи или отделни компютри през публична и незащитена Глобална мрежа.

Потенциални ползватели на VPN са компании и организации с офиси и филиали, които се намират в географски отдалечени точки, в рамките на градове, държави, континенти и т.н., желаещи да имат сигурен достъп до основните информационни ресурси на компанията или организацията, например обща база от данни между различните филиали, обмяна на информация по защитен канал или достъп на индивидуални потребители до ресурсите на компанията.

Основна цел в настоящата дипломна работа е анализиране на проблемите, свързани с реализацията на VPN в една голяма корпоративна мрежа, и в частност в мрежата на СУ, избор на подходящо технологично решение, както и практическа реализация на това решение.

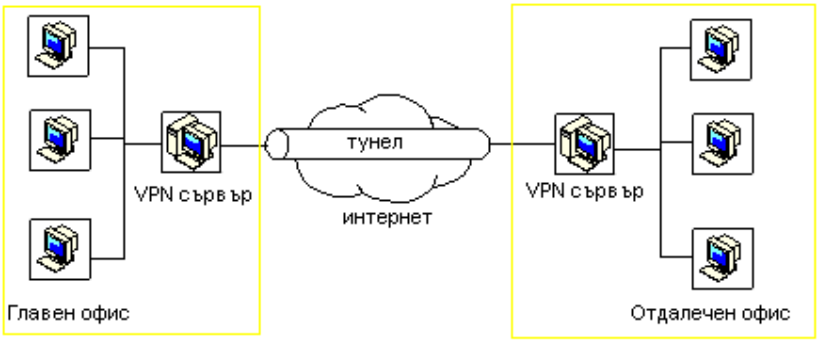
Дипломната работа се състои от четири части, Увод и Заключение. В първата част се представя определение за виртуална частна мрежа, изискванията и използваните технологии във VPN и най-разпространените решения, известни към момента. Във втората част се описват подробно протоколът IPSec и съставните му протоколи. В третата част се представя продуктът Openswan, който се използва в дипломната работа за реализация на VPN – конфигурационните му файлове и параметри. В последната част е изложена реализацията на VPN достъпа до мрежата на СУ – SUNet, конфигурирането на Openswan сървъра за различни клиенти, както и конфигурирането на клиентите за свързването им във виртуалната частна мрежа.

1. Преглед на технологиите за VPN: SSL и IPSec.

1.1. Определение за Виртуална частна мрежа (VPN).

VPN (Virtual Private Network - виртуална частна мрежа) е разширение на обхвата на една частна мрежа (на търговско дружество, университет или др. организация) чрез връзки през споделена или публична среда, каквато е Интернет. VPN позволява да се обменят данни между два компютъра през споделена или публична мрежа по начин, който наподобява свойствата на частна мрежа от типа "точка към точка" (Point-to Point). На практика тя "виртуално" свързва отдалечения компютър, като го прави част от частната мрежа, създавайки криптиран тунел през публичната мрежа. Процесът на конфигуриране и създаване на такава мрежа се нарича "virtual private networking".

За да наподоби връзка от типа "точка към точка", данните са енкапсулирани (обвити) със заглавна част (header – хедър), която съдържа информация за маршрута, осигуряваща преминаването на данните през публичната мрежа и достигането им до крайната точка (дестинацията). За да се постигне подобие на частна мрежа и гарантирана поверителност, обменяните данни се криптират. Пакетите, които са прихванати от публичната мрежа, са нечетими без съответния ключ за декодиране. Частта от връзката, в която частните данни са енкапсулирани, се нарича "тунел". Връзката, в която частната информация е криптирана, се нарича "виртуална частна мрежа". На фигура 1.1 е показана виртуална частна мрежа.



Фигура 1.1.

VPN позволява на потребителите да работят вкъщи или на път, като се свързват по сигурен начин към отдалечения сървър на организацията, използвайки структурата на публична мрежа (например Интернет). За потребителя VPN връзката е от тип

“точка към точка” между неговия компютър и сървъра на организацията. Естеството на междинната мрежа не е от значение, от негова гледна точка той е в частната си мрежа.

VPN технологията дава възможност на фирмите да се свързват със свои филиали или други компании през публични мрежи (като Интернет), осигурявайки сигурна комуникация. В този случай VPN връзката през Интернет логически оперира като глобална мрежа (wide area network - WAN) между страните.

С разработването на виртуалните частни мрежи се посрещат изискванията, произтичащи от разрастването на телекомуникациите и разпространението на глобалните операции, което позволява на служителите да се свързват към централните ресурси и да общуват помежду си.

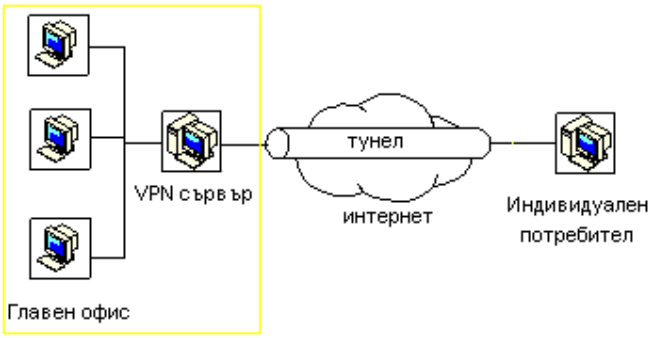
За да предостави на служителите възможността да се свързват към компютърните ресурси на организацията без значение от тяхното местонахождение, фирмата трябва да осигури гъвкаво решение за отдалечен достъп. Обикновено се избира едно от следните решения: отдел, който се занимава със закупуването, инсталирането и поддръжката на набор от модеми и инфраструктура за частната мрежа, или наемане на външна компания, която да осъществява тези задачи.

Нито едно от изброените решения не предлага необходимите способности от гледна точка на разходи, гъвкаво управление и възможност за връзки. Следователно е целесъобразна замяната на базата от модеми и инфраструктура на частната мрежа с по-евтино решение, базирано на Интернет технологията, така че бизнесът да се съсредоточи върху своята основна дейност. Чрез Интернет решението няколко Интернет връзки през доставчик на Интернет и VPN сървър компютър могат да обслужват отдалечените мрежови потребности на стотици или хиляди отдалечени клиенти и офиси.

1.2. Използване на VPN.

Отдалечен достъп през Интернет.

VPN позволява отдалечен достъп до ресурсите на организация през Интернет, осигурявайки поверителност на информацията.



Фигура 1.2.

Вместо да осъществява свързване чрез телефон по модем към сървъра на организацията, потребителят използва Интернет връзка към своя Интернет доставчик и създава VPN връзка през Интернет към сървъра.

Свързване на мрежи през Интернет.

Двата традиционни метода за свързване на отдалечени офиси към един главен корпоративен офис са чрез dial-up връзка през публичната мрежа на телефонна компания или чрез наета линия. Тези методи изискват огромен труд за поддръжка и струват стотици или хиляди левове на месец.

Чрез VPN от тип “мрежа към мрежа” компаниите имат възможност да намалят разходите за скъпи високоскоростни линии. Чрез използване на връзка към Интернет доставчик при отдалечените офиси и една връзка при главния офис компаниите имат възможност да елиминират използването на много високоскоростни връзки, управлението на Frame Relay, поддръжката на WAN архитектура и значителните финансови и административни разходи, свързани с тях.

Съществуват два метода за използване на VPN в зависимост от продължителността на връзката:

- **Постоянна VPN връзка.** Двата офиса са свързани постоянно през Интернет като всеки офис е свързан към локален Интернет доставчик, а VPN връзката през Интернет се осъществява чрез VPN софтуер.
- **VPN при поискване (Demand-Dial VPN Networking).** Отдаленият офис се свързва чрез dial-up връзка към Интернет и чрез VPN софтуер - към главния офис при необходимост, вместо да използва телефонна линия до главния офис, който може да е разположен на много голямо разстояние.

Свързване на компютри през Интранет.

В някои организационни структури определена част от вътрешната информация може да бъде поверителна, поради което е необходимо отделянето на LAN мрежите им от останалите мрежи на организацията. Въпреки че това предпазва поверителната информация на отдела, то едновременно създава и проблеми с достъпа на тези потребители, които не са във физически контакт с отделната LAN мрежа.

1.3. Основни изисквания към VPN.

При реализация на VPN трябва да се спазят няколко основни изисквания, за да се осигури секретност, цялост на данните (*data integrity*) и лесно управление на връзките за отдалечен достъп към ресурсите и информацията на организацията. За тази цел е необходимо да се използват заедно няколко технологии. Те трябва да осигурят връзката на отдалечени потребители към LAN ресурсите, както и на отдалечените офиси един към друг.

За да поддържа тези изисквания, VPN технологията трябва да осигури следното:

- **Идентификация на потребителите**. Решението трябва да провери идентичността на VPN клиента и да предостави VPN достъп само на оторизираните потребители. Също така трябва да има възможност да следи кой и колко време е свързан към VPN.
- **Управление на адресите.** Решението трябва да даде VPN адрес на всеки клиент в Интранет и да пази тези адреси и информацията за клиентите, необходима им за достъп до ресурсите на защитената мрежа като маршрутизираща информация, dns и да предостави филтриране на пакетите, за да осигури защита на вътрешната мрежа от неоторизиран достъп.
- **Криптиране на данните.** Данните, пренасяни през публичната мрежа, трябва да бъдат нечетими за всички освен VPN клиента и сървъра. За тази цел трябва да се използва технология за криптиране между клиента и сървъра.
- **Управление на криптиращите ключове.** За да използва криптиране, VPN решението трябва да предоставя механизъм, използващ криптиращ ключ, и да осигури тунела между клиента и сървъра, както и да генерира и да променя този ключ

за криптиране през определен период, така че сигурността на данните да бъде гарантирана.

1.4. Методи за криптиране, използвани при VPN.

Изборът на алгоритъм за криптиране е един от основните въпроси по сигурността, когато се изгражда криптографски базирано решение. В зависимост от конкретния случай могат да се използват различни криптографски алгоритми:

1. Симетрични криптографски алгоритми. Това са алгоритми, при които криптирането и декриптирането се извършва с един и същ ключ. Симетричното криптиране (с частен ключ) се базира на таен ключ, който се поделя между двете комуникиращи страни. Изпращащият съобщението използва тайния ключ като част от математическата операция за криптиране на чист (plain) текст в шифрован (cipher). Получателят използва същия ключ, за да декриптира cipher текста обратно в plain текст.

За сигурни към настоящия момент се считат блоковите симетрични алгоритми:

- **3DES** (Triple DES) – базира се на широко известния и стандартизиран отдавна алгоритъм за шифриране на данни DES (Data Encryption Standard). Triple DES използва схемата за трикратно шифриране с DES с помощта на два или три различни 56-битови ключа.
- **BlowFish, TwoFish, Goldfish.** BlowFish е 64-битов алгоритъм, който може да използва ключове с променлива дължина, от 32 бита до 448 бита. Той е бърз и неговия сорс код е достъпен. Goldfish и Twofish са наследници на Blowfish.
- **AES** (Advanced Encryption Standard) - AES е най-новият стандарт, базиран на блоковия алгоритъм Rijndael, и е завършен в края на 2000 г. При него дължината на ключа започва от 128 бита, минава през 192 бита и може да достигне до 1024 бита.

2. Асиметрични криптографски алгоритми. Това са алгоритми за криптиране, които използват двойка ключове за криптиране и за декриптиране. В сравнение със симетричните асиметричните криптографски алгоритми са по-удобни за шифриране на съобщения, тъй като не изискват допълнителен защитен канал за обмен на ключовете. Поради по-ниската им производителност и бързо действие обаче, те се използват основно в хибридни криптографски системи за защита на секретните ключове на симетричните алгоритми, с които най-често се шифрират предаваните съобщения. Най-разпространените алгоритми са следните:

- **RSA (**Rivest Shamir Adleman) е най-популярният асиметричен алгоритъм, който може да се използва за шифриране на съобщения, за обмен на сесийни ключове и за цифрови подписи.
- **DSA** (Digital Signature Algorithm) алгоритъмът се отнася към групата на асиметричните алгоритми за цифрови подписи и се използва за контрол за целостта на съобщенията и за удостоверяване на техните източници.
- **Diffie-Hellman** алгоритъмът позволява сигурен обмен на споделен ключ без изпращане на самия ключ по мрежата.

Освен за криптиране асиметричните алгоритми се използват и за удостоверяване чрез цифрово подписване на съобщенията, а също така и за размяна на сесийни симетрични ключове. Алгоритъмът RSA притежава всички изброени по-горе възможности, DSA се използва за удостоверяване между страните в една VPN връзка, а Diffie-Hellman се използва за обмен на сесиен ключ за симетричен криптографски алгоритъм. Затова най-често DSA и Diffie-Hellman се използват заедно.

Дължината на използваните ключове определя сигурността на една криптографска система. Препоръчителната дължина на ключа за съответните алгоритми е:

- За симетрични алгоритми: трябва да се използва минимум 112-битов ключ за осигуряване на добра дълготрайна защита.
- За асиметрични алгоритми: 1024-битов ключ за кратковременна защита и поне 2048-битов ключ за дълготрайна защита.

1.5. Методи за удостоверяване, използвани при VPN.

1.5.1. Методи за удостоверяване целостта на данните. За удостоверяване на целостта на данните (удостоверяване, че данните не са променени в процеса на предаване) се използват еднопосочни хеш функции, които служат за изчисляване на съпровождащи изпращани съобщения **цифрови сигнатури** *(message digests)*. Хеширането е процес, при който математически се преобразува наредена последователност от данни до поле с фиксирана големина. При несанкционирана промяна на съдържанието на съобщението неговата нова цифрова сигнатура ще бъде различна от първоначално определената такава. Обикновено цифровите сигнатури

трябва да бъдат допълнително защитени чрез шифриране, подписване или използване на хеш-функции с ключ, тъй като в противен случай нарушителят може да измени съобщението и да добави изчислената от него нова сигнатура.

Използвани алгоритми за хеширане:

- **MD5** (Message Digest) е наследник на алгоритмите MD2 и MD4, които генерират 128-битова хеш стойност.
- **SHA1 (**Secure Hash Algorithm) е алгоритъм за хеширане, който обработва входящите съобщения на блокове с дължина 512 бита и генерира 160-битова хеш стойност.
- **MAC** (Message Authentication Codes) представлява алгоритъм, при който хеширането се комбинира със симетрично криптиране, така че получената хеш стойност е криптирана и може да бъде прочетена от притежателя на същия симетричен ключ.
- **HMAC** (Hash Message Authentication Codes) представлява алгоритъм, при който хеширането се комбинира със симетрично криптиране. Алгоритъмът е проектиран така, че да работи с различни хеш функции и най-често се използва с MD5 и SHA.

1.5.2. Методи, използвани за удостоверяване между две страни при иницииране на VPN връзка.

- Удостоверяване чрез предварително споделена парола (Pre shared secret).
- Удостоверяване чрез асиметрично криптиране с публичен ключ.
- Удостоверяване чрез цифрови сертификати.

За да се осигури валидността на публичния ключ, той се публикува със сертификат. Това е структура от данни, която е цифрово подписана от доставчика на удостоверителни услуги (certification authority -CA), на когото потребителите могат да се доверят. Сертификатът съдържа поредица от стойности като име на сертификата, информация, определяща собственика на публичния ключ, самият ключ, дата на изтичане на валидността и име на CA. CA използва своя частен ключ, за да подпише сертификата. Ако получателят знае публичния ключ на това CA, той може да се увери, че сертификатът е точно от него и че съдържа достоверна информация и

валиден публичен ключ. Така сертификатите с публичен ключ осигуряват удобен и сигурен метод за удостоверяване на изпращащия съобщението.

1.6. Атаки върху мрежовата сигурност при VPN.

Атаки върху криптографските алгоритми.

Целта на тези атаки е атакуващият да се опита да предположи ключа, използван за криптирането на съобщенията, за да може да декриптира други съобщения, направени със същите ключове.

За увеличаване на сигурността на една криптографска система нейните сесийни ключове трябва често се подменят по време на операцията. Сесийните ключове имат ограничено време на съществуване и ако един такъв ключ е компрометиран, само данните, които той защитава, са също компрометирани, при положение че криптографската система използва PFS (Perfect Forward Secrecy). PFS определя, че последователните сесийни ключове не са свързани по никакъв начин и новите ключове се генерират независимо от старите.

"**Denial of service"** ("Отказ от услуги"). DoS е атака, целяща спирането на определена машина или мрежа, чрез претоварване на капацитета й. Атаките могат да се разделят на два вида:

- **Атаки върху реализацията.** Това са най-често използваният вид атаки. Те се основават на известни уязвимости и слабости, дължащи се на човешки грешки или небрежност. Редица известни уязвимости на използвания софтуер също могат да доведат до компрометиране на виртуалната частна мрежа. Уязвимост в реализацията на PPTP може да причини Denial-of-Service (DoS) при използване на ОС Windows NT, OpenBSD уязвимост, свързана с неправилната обработка на пакети AH/ESP (специален режим на IPSec), също може да доведе до DoS, известна е и уязвимост за Windows 2000 в реализацията на протокола за обмен на ключове IKE за IPSec.
- **Атаки върху операционната система.**

Нерядко реализацията на VPN се основа и надгражда операционната система, какъвто е примерът с операционните системи Windows, където VPN е част от операционната система. В този случай сигурността на VPN се определя и зависи от сигурността на самата операционна система.

“Man in the middle” е основна атака върху мрежовата сигурност. При нея атакуващият има възможността да застане на комуникационния път между двете страни и да spoof-ва другата машина за всяка от комуникиращите страни (той стои по средата и приема съобщения, променя данни и изпраща отново съобщение към получателя). Основният Diffie-Hellman алгоритъм е податлив на тази атака. Решението е с този алгоритъм да се използва допълнителен удостоверяващ механизъм (например HMAC или електронен подпис).

“Replay” атака – атакуващият прихваща част от криптираните данни и по-късно ги вкарва отново в потока от данни. Това може да доведе до повтаряне на валидни транзакции по мрежата, което да нанесе голяма вреда.

1.7. Практически решения за VPN.

От технологична гледна точка VPN могат да бъдат класифицирани в зависимост от това дали функционират чрез мрежа от слой 2 или слой 3 от Open System Interconnection (OSI) модела – фигура 1.3. Протоколите, функциониращи в слой 2, са L2TP и PPTP. VPN, работещи в слой 3, обикновено използват IP протокола като протокол от мрежовия слой. Пример за протокол от слой 3 е IPSec, но L2TP също може да се използва за транспортиране на протоколи от слой 3.

Номер	OSI	TCP/IP
Слой 7	Приложен слой	
Слой 6	Презентационен слой	
Слой 5	Сесиен слой	TCP,UDP
Слой 4	Транспортен слой	
Слой 3	Мрежови слой	IP , ICMP
Слой 2	Канален слой	Драйвери и ОС
Слой 1	Физически слой	

Фигура 1.3.

1.7.1. PPTP VPN протокол.

Point-to-Point Tunneling Protocol (PPTP) е протокол, разработен от Cisco и по-късно включен в операционната система Windows, поради което е получил своята популярност. Той представлява разширение на протокола Point-to-Point Protocol (PPP), което служи за осъществяване на отдалечен достъп. PPP пакетите се

енкапсулират във вътрешността на Generic Routing Encapsulation (GRE) хедър, дефиниран в RFC 1701 и 1702. Полученият пакет се трансформира в IP пакет, който осигурява необходимата маршрутизираща информация. PPP може да се използва за пренос на IP, IPX и NetBEUI пакети.

IP Header	GRE Header	PPP Header	PPP Payload

Фигура 1.4.

Удостоверяването при PPTP връзките се осъществява по два начина: чрез базирания на парола протокол MSCHAP-v2 или чрез използване на EAP-TLS, базиран на цифрови сертификати. Протоколът на Microsoft MSCHAP-v2 установява PPP сесия посредством потребителско име и парола и е наследник на MSCHAP-v1 и Challenge Handshake Authentication Protocol (CHAP). Удостоверяването чрез MSCHAP може да бъде уязвимо поради използването на слаби пароли.

Криптирането в PPTP се извършва чрез Microsoft Point-to-Point Encryption (MPPE). MPPE служи за криптиране на PPP пакети и използва RSA RC4 алгоритъм, като дължината на ключа може да бъде 40, 56 и 128-битова.

Предимства на PPTP.

PPTP е много популярен, защото лесно се конфигурира и е поддържан от всички версии на Microsoft Windows след Windows 95. Освен това съществуват клиенти за почти всички операционни системи – за Линукс, MacOS и различни PDA мобилни устройства. PPTP VPN осигурява защита на данните чрез криптиране и осигурява значителна сигурност на връзката при използване на сложни пароли или чрез метода EAP-TLS, който предлага възможност за използване и на смарт карти. Протоколът позволява свързване на клиенти зад NAT (Network Address Translation).

Недостатъците на VPN връзката, базирана на PPTP, се състоят в това, че тя не осигурява целостта на данните и не предлага удостоверяване на всеки пакет, а само първоначално удостоверяване на ниво потребител, което прави протокола уязвим на множество атаки, например “replay” атаки.

1.7.2. L2TP VPN протокол.

Layer 2 Tunneling Protocol (L2TP) функционира в каналния (datalink) слой на OSI модела и представлява протокол за тунелиране на мрежовия трафик между две точки през съществуваща мрежа (обикновено Интернет). В най-общия случай той пренася PPP сесия в L2TP тунел, подобно на PPTP. След създаване на тунела мрежовият трафик между двете точки е двупосочен. L2TP по IP интермрежи използва UDP и серия от L2TP-съобщения за управление на тунела. L2TP използва UDP, за да изпрати L2TP-капсулирания PPP пакет като тунелирани данни. Данните на капсулирания PPP пакет могат да бъдат криптирани и/ или компресирани, въпреки че реализацията на Microsoft на L2TP не използва MPPE за криптиране на PPP данни.

IP Header	UDP Header	L2TP Header	PPP Header	PPP Payload

Фигура 1.4.

L2TP сам по себе си не осигурява поверителност или удостоверяване и затова най-често се използва с IPSec, който предоставя поверителност, удостоверяване и интегритет. Комбинацията между двата протокола се нарича L2TP/IPSec и е описана в RFC3193.

L2TP предлага някои предимства в сравнение с PPTP, като например: L2TP има възможност да поддържа множество тунели между крайни точки; поддържа компресиране на хедъри, което спестява допълнителната информация; има възможност за тунелно удостоверяване и може да работи по мрежи, различни от IP, използващи ATM или Frame Relay виртуални вериги.

L2TP/IPSec се нуждае от сертификационна инфраструктура, за да прилага компютърни сертификати или споделени ключове. Поддържа се в Windows Server 2003, Windows XP, Windows 2000 и други L2TP клиенти, работещи с Microsoft L2TP/IPsec VPN Client.

1.7.3. IPSec VPN.

IPSec (Internet Protocol security) е утвърден стандарт за сигурност на протокола IP, разработен за неговата защита, тъй като Internet Protocol не предоставя такива възможности.

IPSec добавя допълнителни услуги, липсващи в IP, които предоставят следните възможности:

- Криптиране на трафика;
- Цялост на данните;
- Удостоверяване между участващите страни.

Тези услуги се осъществяват чрез трите протокола, от които се състои IPSec:

- AH (Authentication Header) - предоставя удостоверяване на IP пакетите.
- ESP (Encapsulating Security Payload) - предоставя криптиране и удостоверяване на IP пакетите.
- IKE (Internet Key Exchange) - договаря параметрите на връзката, включително ключовете.

IPSec е проектиран да работи, както в тунелен режим (за защита на комуникацията между отделни мрежи), така и в транспортен режим (за защита на комуникациите между двама участника), поради което се използва за изграждане на виртуални частни мрежи.

IPSec е единственият стандартизиран протокол, който отговаря на всички изисквания към виртуалните частни мрежи, затова е избран да бъде използван за реализирането на VPN в настоящата дипломна работа и ще бъде разгледан подробно в глава 2.

1.7.4. VPN протоколи, работещи в по-горен слой.

Съществуват протоколи, които работят над мрежовия слой 3 от OSI модела. Пример за това са решения, базирани на SSL, SSH и TLS, опериращи в слой 4-7 от OSI.

В настоящата дипломна работа ще бъде разгледан продуктът OpenVPN, който е представител на горните протоколи.

OpenVPN е VPN, базирана на индустриалния стандарт Secure Socket Layer/Transport Socket Layer (SSL/ TSL) протокол, който се използва при уеб-браузърите. Тя реализира множество конфигурации: отдалечен достъп, виртуални частни мрежи от типа “от сайт до сайт” (*site-to-site*), както и корпоративни решения за отдалечен достъп, включващи балансиране на натоварването, възстановяване след откази (*failover*) и прецизиране на контрола на достъпа спрямо даден потребител или

група. За целта използва правила, характерни за защитните стени (*firewall*), които се прилагат към виртуалния интерфейс на VPN-а.

Мрежовият интерфейс се реализира по проекта Universal/ Tap Device Driver. Устройствата Tun представляват виртуални мрежови устройства, емулиращи двойни връзки, докато устройствата Tap емулират Ethernet устройствата. Всеки изпращан през тях пакет се предава на програмата OpenVPN. Тя от своя страна използва програмите и библиотеките в проекта SSL за гарантиране на сигурността на пренасяните данни. Информацията се кодира и така виртуалната мрежа получава "частен" статут.

За критичната фаза на обмен на ключове OpenVPN използва RSA/DHE.

OpenVPN осигурява разширяване на мрежовата сигурност на OSI слой 2 или 3 и поддържа гъвкави методи за удостоверяване на клиенти чрез сертификати, смарт карти и др. Той позволява специфична за потребителя или групата политика за контрол на достъпа, използвайки правила на защитна стена, приложени към виртуалния интерфейс на VPN.

OpenVPN поддържа три метода за удостоверяване на клиента:

1. Най-ненадежден е методът ***"предварително споделен ключ" (pre-shared key)***, при който ключът е известен и на двата края още преди изграждането на тунела. Този метод е приложим само за проверки и обучения.

2. Вторият метод, който е най-сигурен, е чрез използване на ***X.509 сертификати.***

3. Третият метод представлява ***свързване на OpenVPN със стандартите за Линукс с включващи се модули за удостоверяване*** (Pluggable Authentification Module, PAM). Потребителят на клиента се удостоверява с помощта на комбинация от име и парола, които се проверяват на Линукс сървъра.

OpenVPN притежава следните предимства:

- използва изпитани криптографски алгоритми (SSL/ TLS), RSA сертификати и X509 PKI;
- позволява лесно инсталиране и конфигуриране;
- използва само един външен порт за много външни клиенти;
- поддържа динамични IP адреси и NAT;
- притежава висока съвместимост със защитните стени;

- работи с различни операционни системи - Linux, Solaris, OpenBSD, FreeBSD, NetBSD, Mac OS X, and Windows 2000/XP;
- поддържа компресиране на предаваните данни;
- не изисква лицензни такси - разпространява се GNU License и др.

Недостатъците му са свързани със слабата разпространеност и несъвместимостта с IPSec, IKE, PPTP или L2TP.

2.Описание на протокола IPSec.

2.1. Общо описание и основни термини, свързани с IPSec.

Целта на IPSec е да осигури стандартни криптографски механизми за сигурност между IPv4 и IPv6 обекти. Между основните услуги са контрол на достъпа, гарантиране на ненакърнимост на данните (integrity) по пътя им от източника до местоназначението, както и удостоверяване за произхода им, разпознаване и отхвърляне на подвеждащи атаки (replays), поверителност (чрез криптиране) и мерки за поверителност на потоците от трафик. Всички тези услуги се предоставят на ниво IP.

В този смисъл може да се каже, че IPSec включва минимални функционални характеристики на една "защитна стена" (firewall). Те се подсилват и от криптографски-базираните удостоверявания и проверки за ненакърнимост на данните, наложени на целия IPSec трафик.

IPSec се реализира върху хост компютър или като **"защитен шлюз"** (**security gateway - SG**) – защитна стена или маршрутизатор с IPSec възможности. Възможно е да се проектира самостоятелно устройство. Защитата, предлагана от IPSec, се базира на изискванията, дефинирани в базата от данни на политиката за сигурност (**Security Policy Database - SPD**) от страна на потребителя/администратора или приложна програма. Въз основа на тях IP пакетът се защитава (PROTECT) с помощта на IPSec услугите, отхвърля (DISCARD) или се пропуска без обработване от IPSec (BYPASS).

В основата на архитектурата на IPsec лежи **"Сдружение за налагане на политика за сигурност** (**Security Association - SA**). Протоколите, чрез които се реализират услугите на IPSec VPN - AH (добавяне на удостоверително заглавие, хедър, към IP пакета) и ESP (опаковане на полезните данни в IP пакета), използват SA. Същото важи и за протокола за обмен на ключове Internet Key Exchange (IKE).

“Сдруженията” възникват в резултат от прилагането на политика, която дефинира криптирането, създаването на ключове, удостоверяването и всички процеси, които ще бъдат прилагани към данните.

SA представлява еднопосочно съединение, което предлага услуги за сигурност на трафика, който носи, чрез AH или ESP (например: DES , 3DES, AES протокол за криптиране и MD5 или SHA-1 протокол за удостоверяване). За всеки един от протоколите, който се поддържа, AH или ESP, или и двата, трябва да се създаде отделна SA. Ако искаме да имаме двупосочни комуникации между две IPSec системи, ще са ни необходими две SA (по една за всяка от посоките). В този смисъл IKE по подразбиране създава двойки SA. Така че в IPSec VPN съществуват две форми на SA:

ISAKMP (Internet Security Association Key Management Protocol), известни и като ***IKE.*** IKE SA е двупосочна и предоставя сигурен комуникационен канал между двете страни, който може да се използва за договаряне на по-нататъшните комуникации.

IPSec SAs. IPSec SA е еднопосочна и се използва за действителната комуникация между устройствата. За двупосочна комуникация трябва да има поне две IPSec SAs – по една за всяка посока предаване и приемане.

IPSec SA могат да се дефинират еднозначно чрез три компонента:

1. Security parameter index (SPI) – 32-битово число, служещо за идентификация на SA.

2. IP адреса на получателя.

3. Идентификатор на протокола за сигурност, който дефинира дали SA е AH или ESP.

Информацията за всички SA се съдържа в ***Security Association Database (SAD).***

Архитектура.

Базовата архитектура на IPSec се описва от стандарта RFC 4301. Върху нея се градят всички реализации. Там се дефинират услугите, които IPSec предлага, как и къде се използват и как пакетите се конструират и обработват.

Повечето от услугите за сигурност на IPSec се осигуряват чрез един от двата протокола за защита на трафика. Authentication Header (RFC 4302) защитава целия IP пакет, а Encapsulating Security Payload (RFC 4303) - само на полезните данни, които носи IP пакетът, т.е. на по-горните слоеве. В тази схема участват и процедурите и

протоколите за управление на криптографските ключове. Точно как и кои процедури/протоколи ще се използват е зависи от волята на администраторите.

Всеки един от протоколите, AH и ESP, поддържа два режима на работа – транспортен и тунелен. Транспортният осигурява защита главно за протоколите на по-горните слоеве. IPSec хедърът се вмъква между IP хедъра и горните слоеве. В тунелен режим целият IP пакет се вмъква в нова дейтаграма, като IPsec хедърът се вмъква между външния и вътрешния хедър.

Транспортен режим.

Концепцията на VPN решенията се основава главно на тунелите. Тунел възниква, в случаите когато оригиналните данни са енкапсулирани в нов пакет и препратени въз основа на атрибутите на протокола на новия пакет. Целта е да се осигури прозрачна връзка за основния протокол. Транспортният режим на IPSec е уникален с това, че не се осъществява енкапсулиране в нов пакет. Оригиналният IP хедър се използва и данните се препращат въз основа на оригиналните атрибути, заложени от протокола.

Тъй като дейтаграмите се предават от TCP/IP протокола към мрежовия слой, те са снабдени със съответната хедър информация от всеки слой. В края на операциите в мрежовия слой, IPSec премахва оригиналния IP хедър и криптира хедъра и данните на горния протокол. След това IPSec добавя избрания хедър на протокола за сигурност преди отново да приложи оригиналния хедър.

В транспортния режим оригиналният пакет остава същият с изключение на две важни промени: данните са криптирани и в пакета е добавен удостоверяващ протокол. По този начин пакетът получава интегритет, за да се гарантира, че данните не са променяни по време на пренос.

Главното ограничение се състои в това, че в рамките на транспортния режим не могат да се предоставят шлюзови услуги. Ако две системи комуникират в транспортен режим, комуникацията не може да бъде осъществена отвъд крайните точки на VPN. Транспортният режим е запазен за комуникация от типа “точка към точка”. Ако при неговото използване е установен VPN с VPN шлюз, отдалечената система има възможност да комуникира само с шлюза, като не може да обменя данни с вътрешната мрежа.

Съществуват няколко случая, при които използването на транспортния режим е необходимо. VPN чрез транспортен режим пряко към шлюз може да осигури защитено администриране без да съществува безпокойство относно достъпа до вътрешната мрежа. В ситуациите, при които VPN шлюз се управлява отдалечено от трета страна, достъпът й до вътрешната мрежа може да бъде нежелан. Способността за вмъкване на транспортен режим SA в тунелен режим SA осигурява разширена защита отвъд шлюза във вътрешната мрежа.

Тунелен режим.

Тунелният режим е най-разпространен при VPN и широко използван в IPSec имплементациите. Той се използва главно за шлюзови услуги, защото енкапсулирането осигурява способността за предаване на няколко сесии чрез една точка. Това позволява VPN шлюз да деенкапсулира данните и да ги препрати към крайната вътрешна точка.

Тунелният режим енкапсулира пакет, предназначен да комуникира в мрежа, и го енкапсулира в друг пакет за предаване към отдалечена мрежа. Този процес позволява протокол, който е възприет за входна и изходна точка, да пренася друг комуникационен протокол, който обикновено не би могъл да се препрати през мрежата.

В тунелния режим целият оригинален пакет е енкапсулиран и кодиран, като са добавени нов IP хедър и хедър на удостоверителния протокол. Резултатът е нов пакет, който съдържа два IP хедъра. Те се използват за комуникационни услуги в съответните мрежи:

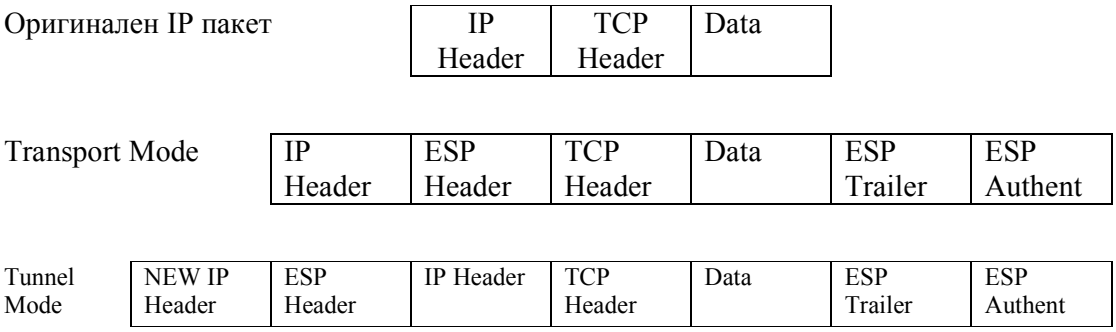
- Вътрешен хедър
- Външен хедър.

Вътрешният IP хедър се прилага към оригиналните дейтаграми, които се предават в мрежовия слой, като в края на краищата се енкапсулира в нов пакет. Новият пакет получава външен IP хедър, който съдържа информация, необходима за комуникирането във VPN.

Протоколите за сигурност представляват същността на IPSec и тяхната реализация определя конкретно как да се приложи транспортен или тунелен режим. Те могат да бъдат използвани поотделно или единият да бъде “вмъкнат” в другия. Така се разширяват възможностите на VPN и тяхното функциониране.

2.2. Протокол Encapsulating Security Payload (ESP).

Протоколът е описан в RFC 4303. ESP предоставя няколко услуги за сигурност - поверителност на данните, интегритет, удостоверяване на източника, услуги срещу повторни атаки (anti-replay) и ограничена поверителност на трафика. Разширяването на поверителността и интегритета на комуникацията са свързани с режима. В тунелен режим вътрешният IP хедър е добре защитен, докато външният не е. В транспортен режим няма вътрешен IP хедър, поради което защитата на мрежовия слой е ограничена. Наборът от предоставяните услуги зависи от избраните опции по време на установяването на Security Association (SA) и конкретната имплементация.



Фигура 2.1. ESP в тунелен и транспортен режим.

ESP осигурява поверителност посредством криптиране и интегритет на данните с удостоверяване. Използваните за ESP алгоритми се определят от атрибутите за създаване на SA. ESP сам по себе си не е управляван от специфични алгоритми, а представлява отворен стандарт за прилагане на различни такива (например: DES, 3DES, AES). Стандартът дефинира процедури и необходими действия за криптиращия процес, но не дефинира какво може и не може да бъде използвано за криптографската услуга. Прилагането на поверителност не е задължително, но ако е необходимо само удостоверяване, се използва протоколът AH.

Въпреки че поверителността и удостоверяването са основните услуги, предоставяни от ESP, те не са задължителни. Една от двете обаче трябва да бъде използвана. Основната идея е да се използва ESP при необходимост от удостоверяване и криптиране, а AH – когато е необходимо разширено удостоверяване без криптиране.

Дефиниция на ESP хедър.

Положението на ESP хедъра е непосредствено след IP хедъра и опциите без значение от режима на работа на VPN. Номерът на протокола в IP хедъра е 50, тъй като ESP е дефиниран като протокол 50 от Асоциацията за присвояване на имена в Интернет (IANA).

Както е показано на фигура 2.2. ESP хедърът съдържа няколко полета за създаване и поддържане на SA.

<table>
<tr><td colspan="3">Security Parameters Index (SPI)</td></tr>
<tr><td colspan="3">Sequence Number Field</td></tr>
<tr><td colspan="3">Payload Data (Variable)</td></tr>
<tr><td></td><td colspan="2">Optional Padding (0-255)</td></tr>
<tr><td></td><td>Pad Len.</td><td>Next Header</td></tr>
<tr><td colspan="3">Authentication Data (Variable)</td></tr>
</table>

Фигура 2.2. ESP хедър

“Полето SPI” е първата 32-битова стойност в хедъра. Стойностите от 1 до 255 са запазени, а нула се използва само при договаряне в ISAKMP на SA.

Следващото поле се нарича *“Sequence Number Field” (пореден номер)* и се използва за защита срещу повторни атаки (replay) и за удостоверяване на източника. Ако защитата срещу повторни атаки е активирана, поредният номер трябва да бъде уникален за времето на действие на SA. Ако номерът достигне максималната стойност, SA се изтрива и се установява нова SA.

“Payload data” е поле с променлива дължина, което е свързано с размера на данните, предавани от по-горните мрежови слоеве. Различни криптиращи алгоритми се използват за инициализиране на декриптиращия процес. В случай че конкретният криптиращ алгоритъм изисква инициализиращ вектор (initialization vector-IV), той се вмъква пред данните.

“*Padding*” осигурява ограничена поверителност на комуникационния поток, но това е повече вторичен продукт от други изисквания, свързани с криптирането. Някои криптиращи алгоритми изискват оригиналните данни да съответстват на размера на блока, използван от алгоритъма. Padding се използва и за да гарантира, че дължината на pad и следващите полета на хедъра са дясно подравнени към 4-байтовата граница.

Полето *“Pad Length”* е стойност, която дава на получателя информация относно дължината или броя на padding, които са добавени към шифрования текст. Padding е помощно поле, но използването му не е задължително. Полето “Pad Length” обаче трябва да съществува, дори когато не се използва pad.

Полето *“Next Header”* е 8-битово поле, което показва типа от данни, съдържащи се в полето Payload data. То е необходимо за декриптирането на данните, което да позволи те да бъдат обработени правилно.

“Authentication Data” е поле с променлива дължина, което съдържа Integrity Check Value (ICV), изчислена от ESP пакета без Authentication Data. Дължината му се определя от избраната удостоверителна функция. Полето Authentication Data не е задължително и се включва, само когато по време на осъществяване на SA е избрана удостоверителната услуга.

За правилното прилагане на ESP са необходими няколко последователни стъпки. Процедурите са пряко свързани с посоката на данните: предаване или приемане.

Процес на предаване.

Съществува последователност от пет стъпки, които са необходими за прилагането на ESP. Те предполагат, че са използвани поверителност и удостоверяване.

1. Енкапсулиране на горния слой данни в полето Payload. В транспортен режим това се прилага само за горния слой протоколи, а ако комуникацията е в тунелен режим – за горния слой и оригиналния IP хедър.

2. Добавяне на необходимия padding. Попълване на полетата Pad Length и Next Header.

3. Криптиране на данните, което включва горния слой, pad, pad length и next header. Ако криптиращият алгоритъм изисква инициализиращ вектор (IV), такъв се създава и прилага в съответствие с използвания криптиращ стандарт.

4. Създаване на ESP хедър. Вмъкване на пореден номер. Ако това е първият пакет, му се присвоява стойност единица.

5. След завършване на криптирането, удостоверяването се прилага към информацията в ESP хедъра (SPI и Sequence number) и Payload, което включва

данните от горния слой, информацията от padding и полето Next Header. Полученият резултат се добавя в края на пакета след шифрования текст.

Процес на приемане.

Ако по време на пренасянето възникне фрагментация, преди обработването на ESP се осъществява дефрагментация. След нейното приключване се извършват няколко стъпки:

1. Използване на полето SPI в ESP хедъра и IP адреса на получателя с цел намиране на SA в базата данни SA. Ако този процес се окаже неуспешен, пакетът се отхвърля.

2. Проверка на поредния номер, в случай че е активирана защита срещу повторни атаки.

3. Използване на SA, установена в предходния етап, за удостоверяване и декриптиране на данните в ESP.

4. След декриптирането - използване на вътрешния IP хедър за определяне на селектор в SA.

5. Намиране на стойност в SPD , която съвпада с данните от вътрешния пакет с цел установяване валидността на пакета за съответната SA.

6. Препращане на данните към крайната точка въз основа на информация от вътрешния IP хедър.

Удостоверяване на ESP и защита срещу “replay” атаки.

В IPSec протокола съществуват механизми за защита от атаки, които се основават на събирането на данни и повторното им връщане в по-късен етап. Поредните номера в хедърите на протоколите за сигурност са предназначени да осигуряват защита срещу повторни атаки. Този, който изпраща пакета, попълва и управлява поредния номер, а получателят следи за неговата проверка.

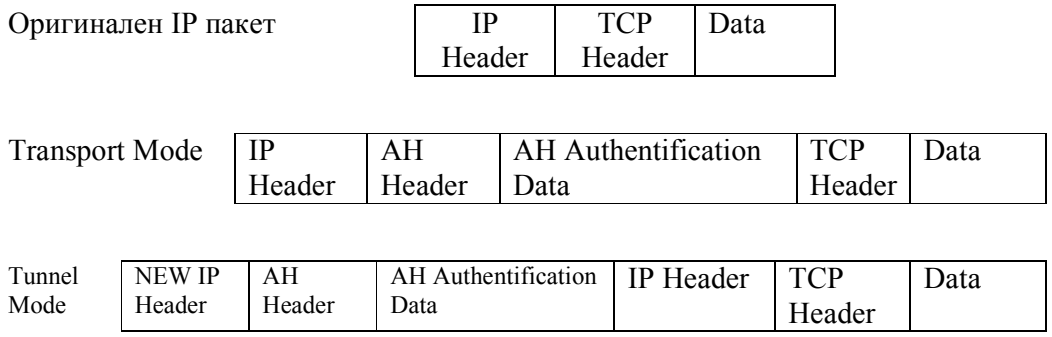
Според стандарта RFC (RFC2406) защитата срещу повторни атаки може да бъде активирана, само ако е избрано удостоверяване на източника на данните и това е предоставено на преценката на получателя. Ако ESP е имплементиран без удостоверяване и е приложена защита срещу повторни атаки, “поредният номер” не е удостоверен. В противен случай поредният номер се включва в полето authentication payload.

Ако някой прихване пакет и направи опит да промени поредния номер при използвано удостоверяване, получателят ще открие промяната и ще отхвърли дейтаграмата. Почти всички имплементации поддържат ESP без удостоверяване, тъй като това е позволено в RFC. Неизползването на удостоверяване в ESP обаче разрива няколко уязвимости, поради което строго се препоръчва неговото прилагане.

2.3. Протокол Authentication Header (AH).

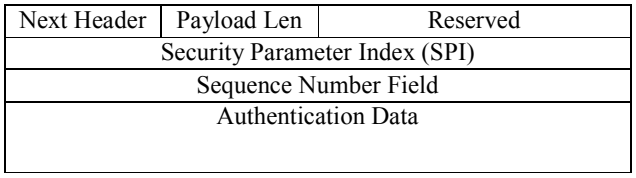
Протоколът, който добавя удостоверително заглавие (хедър) - Authentication Header (AH), е описан в RFC 4302. AH осигурява интегритет на данните, удостоверяване на източника и възможност за използване на защита срещу повторни атаки. AH обаче не предоставя поверителност. Неговата основна функция е да осигури удостоверителни услуги при комуникацията. Тъй като при AH липсва поверителност, не е необходимо да се дефинира криптиращ алгоритъм.

Както при ESP, положението на AH хедъра зависи от режима на комуникация. В транспортен режим хедърът е вмъкнат след IP хедъра и опциите и преди който и да е протокол от горния слой, включително други IPSec хедъри. В тунелен режим оригиналният пакет е разположен зад хедъра, като новият IP хедър и опции предхождат AH хедъра. От гледна точка на външния IP хедър положението на AH е същото, както в транспортния режим.



Фигура 2.3.

На фигура 2.4. е показана структурата на AH хедъра, който съдържа следните полета:



Фигура 2.4. AH Header.

В IP хедъра 8-битовото поле на протокола е с номер 51, тъй като AH е дефиниран като протокол 51 от IANA.

За разлика от ESP полето Next Header дефинира следващия хедър в пакета, който следва непосредствено удостоверяващите данни. То е 8-битово поле, което определя типа на данни след полето Authentification Header.

Payload Length представлява 8-битово поле, което определя дължината на AH в 32-битови думи минус “2”.

Reserved представлява 16-битово поле, което е запазено за бъдещо използване. Стойността му трябва да бъде “нула”.

Следващото поле SPI е 32-битово число и има същите характеристики, както в протокола ESP.

Полето Authentication Data представлява удостоверяване, използвано за проверка на целия пакет. Размерът се контролира частично от избраната функция за удостоверяване на съобщението. Тъй като стойността обаче може да е по-голяма от 32 бита, е възможно да се изисква прилагане на padding.

Съществуват няколко стъпки, необходими за правилното прилагане на AH. По подобие на ESP процедурите са свързани пряко с посоката на данните: предаване или приемане.

Процесът на предаване включва четири основни стъпки:

1. Определяне на съответната SA чрез наличната в оригиналния пакет информация.

2. След като комуникацията за прилагане на AH е определена, се вмъква AH хедър след IP хедъра и се имплементира SPI.

3. Генериране на пореден номер и въвеждането му в хедъра.

4. Изчисляване и прилагане на интегритет на съобщението, попълване на получения резултат и добавянето му в края на АН и преди протоколите от горния слой.

Тъй като основната цел на АН е да осъществява удостоверяване, създаването на удостоверителни данни, което обхваща целия пакет, изисква използването на по-точни техники в сравнение с ESP. По време на удостоверителния процес трябва да бъдат отчетени няколко аспекта:

- Определяне на полетата на IP хедъра, които са постоянни и променливи.
- Задаване на стойност “нула” на променливите полета.
- Събиране на информацията за АН хедъра; на удостоверяващите данни се задава стойност нула.
- Събиране на информацията за протокола от горния слой, за която се предполага, че е постоянна.

Постоянно поле означава поле, което няма да се променя при преноса.

Променливо поле представлява стойност, която може да бъде променяна при преноса и поради тази причина не се отчита от получателя. Променливите полета не участват в удостоверителния процес. Ето защо след определянето им те се нулират преди извършване на удостоверителния процес.

Постоянните полета в IP хедъра са следните:

- version
- IP header length
- total length
- identification
- protocol (51 за АН)
- source address
- destination address.

Променливите полета са:

- type of service
- all flags
- fragment offset
- time to live (TTL)
- header checksum.

Процес на приемане.

Както при ESP, ако по време на пренасянето възникне фрагментация, преди обработването на AH от IPSec се осъществява дефрагментация. След нейното приключване са необходими няколко стъпки, които трябва да бъдат извършени:

1. Използване на полето SPI в AH хедъра и IP адреса на получателя с цел намиране на SA в базата данни SA. Ако този процес се окаже неуспешен, пакетът се отхвърля.

2. Проверка на поредния номер, в случай че е активирана защита срещу повторни атаки.

3. Използване на SA, установена в предходния етап, за удостоверяване на пакета.

4. Препращане на данните към крайната точка въз основа на информация от IP хедъра.

2.4. Протокол Internet Key Exchange (IKE).

Протоколът IKE е хибрид от протоколите Oakley и SKEME и действа в рамките, определени от Internet Security Association and Key Management Protocol (ISAKMP). ISAKMP определя формата на пакетите, таймерите за препредаване и изискванията за конструиране на съобщенията. Oakley и SKEME определят стъпките, които двете страни трябва да предприемат за установяване на споделен, удостоверен ключ. Освен това IKE протоколът се грижи да променя периодично ключовете (rekeying), за да осигури тяхната поверителност.

IKE използва концепцията на SA, но физическата конструкция на IKE SA е различна от IPSec SA. IKE SA определя начина, по който се осъществява комуникацията между две страни, например кой алгоритъм да се използва за криптиране на трафика на IKE, как да се удостоверят двете страни и т.н. IKE SA се използва, за да установи необходимите IPSec SA между страните.

Oakley определя “режимите”, а ISAKMP – “фазите”. Връзката между двете е пряка и IKE представя различните начини на размяна, като режими, които действат в една от двете фази.

Във фаза 1 двете страни в ISAKMP установяват сигурен и удостоверен канал, по който да комуникират. Това е т. нар. ISAKMP Security Association. Двата режима,

чрез които се осъществява обмяна във Фаза 1, са “Main Mode” и “Aggressive Mode” и се използват само в тази фаза.

Във Фаза 2 съществуват два режима – “Quick Mode” и “New Group Mode”. “Quick Mode” се използва за установяване на SA на базата на основния протокол за сигурност. “New Group Mode” е режим от Фаза 2, но услугите, които предоставя, ползват операциите във Фаза 1.

Във Фаза 2 SA се договарят за услуги като IPsec например, за които е необходим ключ и/или договаряне на параметри. “Quick Mode” завършва обмяната във Фаза 2 и се използва само в тази фаза.

“New Group Mode” следва Фаза 1 и служи за установяване на нова група, която може да бъде използвана при бъдещо договаряне.

ISAKMP SA е двупосочна. Поради това веднъж установена, всяка страна може да инициира Quick Mode, Informational и New Group Mode обмен. ISAKMP SA се идентифицира чрез “бисквитките” на инициатора, следвани от тези на отсрещната страна в комуникацията.

Основната разлика между режимите във Фаза 1 е броят на обменените съобщения за установяване на SA. В Main Mode те са шест, а в Aggressive Mode - три.

2.4.1. ISAKMP хедър.

Всички операции във Фаза 1 и 2 изискват ISAKMP хедър, който може да бъде следван от различни като дължина payload (товар с данни). ISAKMP хедърът е необходим на ISAKMP протокола, за да поддържа връзката по време на комуникацията.

Съществуват 13 дефинирани payload-и, които могат ясно да се разграничат в един ISAKMP хедър. Всеки от тях може да бъде използван за генериране на информация за създаване, управление или изтриване на SA.

Както е показано на фиг. 2.5. ISAKMP хедърът е 28-байтов и съдържа основната информация за връзката.

“Бисквитките” на инициатора и отсрещната страна са 8-байтови случайни числа, които се използват, за да идентифицират SA. “Бисквитките” са необходми на системите за бързо идентифициране на валидността на комуникациите. Това намалява излагането на атаки от типа “отказ от услуга”(denial-of-service), като позволява на

получателя да провери основния номер, споделен в рамките на предходна комуникация, преди обработването на цялостния пакет.

<table>
<tr><td colspan="4">Initiator Cookie</td></tr>
<tr><td colspan="4">Responder Cookie</td></tr>
<tr><td>Next Pay</td><td>MJ Ver.</td><td>MN Ver.</td><td>Flags</td></tr>
<tr><td colspan="4">Message ID</td></tr>
<tr><td colspan="4">Length</td></tr>
</table>

Фигура 2.5. ISAKMP хедър.

Полето “Next Payload” се изполва в ISAKMP хедъра и в Generic хедър за идентифициране на съдържанието на следващия Payload. Това е 8-битово поле, което предоставя 255 възможности, 13 от които имат значение за ISAKMP/ IKE.

Полетата “Major” и “Minor” определят версията на ISAKMP, която се използва.

“Exchange type” определя типа на предаваното съобщение. Съществуват няколко съобщения, които могат да се използват в ISAKMP:

NONE	0
Base	1
Identity Protection	2
Authentication Only	3
Aggressive	4
Informational	5
ISAKMP Future Use	6–31
DOI Specific Use	32–239
Private Use	240–255

Полето “Flag” позволява комбинация от три състояния на съобщения.

Полето “Message ID” се използва, за да идентифицира уникално състоянието на протокола във Фаза 2.

Полето “Length” определя дължината на payloads и на ISAKMP хедъра в байтове.

2.4.2. Payload-и, използвани в IKE.

Generic Payload хедър. Този хедър свързва няколко payload-а в едно съобщение. Той е необходим почти при всяка размяна на IKE и позволява гъвкавост на протокола. Към всеки payload е прикрепен generic хедър.

Next payload	Reserved	Payload Length

Фигура 2.6. Generic Payload.

Полетата Next Payload и Length са идентични с тези от ISAKMP хедъра. Комбинацията от двете полета позволява гъвкаво създаване на съобщение и намалява сложността на ISAKMP хедъра.

Security Association Payload се използва за предаване на атрибутите за сигурност, които да бъдат използвани при установяването на SA.

Next Pay	Reserved	Payload Length
DOI		
Situation		

Фигура 2.7. Security Association Payload.

Proposal Payload съдържа обем информация, свързана със създаването на SA. По време на установяването на една сигурна връзка всяка система трябва да определи ниво на удостоверяване и поверителност, както и различна информация, свързана със сигурната връзка. Proposal payload е последван от един или няколко transform payload-и, които съдържат информация за SA.

Next Pay	Reserved	Payload Length	
Proposal#	Protocol ID	SPI Size	# of Trsfms
SPI			

Фигура 2.8. Proposal Payload.

Полета на Proposal payload:

Proposal number (Proposal #) представя номера на предложението в съобщението. В едно съобщение могат да съществуват няколко предложения с различни номера.

Protocol ID е протоколът за сигурност, който се установява по време на договарянето, например ESP или AH.

SPI size позволява интегриране на основни протоколи, различни от IPSec.

Transform Payload съдържа един единствен атрибут. Transform # се използва, ако съществуват повече от един transform payload. Например, ако се използва ESP с DES криптиране и MD5 удостоверяване, в предложението ще има два transform payload-а. *Transform ID* дефинира протокола, определен за трансформация. *SA Attributes* се съдържат вътре в специален *payload* - Information Attributes.

Next Pay	Reserved	Payload Length
Transforml#	Transform ID	Reserved
SA Attributes		

Фигура 2.9. Transform payload.

Identification Payload.

Този Payload се използва за размяна за идентифициращa информация за удостоверяване между двете страни.

Next Pay	Reserved	Payload Length
ID Type	Protocol ID	Port
Identification Data		

Фигура 2.10.

Чрез полето ID_Type се определя типът идентификация, който може да бъде IP v4 адрес, FQDN, IP v4 подмрежа, IPv6 адрес, IPv6 подмрежа и други.

Certificate Payload.

Този Payload се използва за предоставяне на информация за сертификата, използван за удостоверяване пред отсрещната страна.

Next Pay	Reserved	Payload Length
Cert encode	Cerificate	
	Data	

Фигура 2.11. Certificate Payload.

Полето certificate encoding определя формата на последващата сертифицираща информация, като различните типове се задават със следните стойности:

PKCS #7 wrapped X.509 certificate	1
PGP Certificate	2
DNS Signed Key	3
X.509 Certificate - Signature	4
X.509 Certificate - Key Exchange	5
Kerberos Tokens	6
Certificate Revocation List (CRL)	7
Authority Revocation List (ARL)	8
SPKI Certificate	9
X.509 Certificate - Attribute	10
RESERVED	11–255

Непосредствено след certificate encoding се намира полето, съдържащо сертификата.

Certificate Request Payload позволява една система да отправи запитване за сертификата на отсрещната страна в комуникацията. Отговарящият изпраща своя сертификат след получаване на запитване, в случай че той поддържа сертификати. Ако две системи се опитват да установят VPN и необходимото удостоверяване е чрез сертификати, то те трябва да бъдат предоставени под някаква форма.

Notification Payload се използва за предаване на информация относно статуса между двете страни. Основното му предназначение е за грешни съобщения в случай, че payload-ът е негоден (невалиден), деформиран или не бъде приет. Той се използва и за размяна на базова информация, като например информация относно известяване за продължителността на връзката, когато SA е към своя край.

Delete Payload съдържа информация за SA и специфична за протокола информация, което позволява на получателя да изтрие SA и да премахне съответстващата информация от SPD и SAD (SA Database).

По време на съобщенията във Фаза 1 и Фаза 2 на ISAKMP в payload-ите трябва да бъде споделяна различна информация. В някои payload-и, като Transform payload, атрибутите се съдържат в attribute payload, който представлява част от данни на пренасящия payload.

AF	Attribute Type	Attr.Length/Attr.Value
AF=0 Attribute Value AF=1 Not used		

Фигура 2.12. Дефиниция на атрибут.

Attribute payload се състои от следните полета: AF, Attribute Type – тип на атрибута, Attribute Length – дължина на атрибута, и Attribute Value – стойност на атрибута.

Ако стойността на бита AF е 0, полето за дължина е включено и стойността на атрибута е предадена в свое собствено поле. Ако обаче стойността на бита AF е 1, полето за дължина не е необходимо и е заменено от стойността на атрибута, като дължината на payload-а е редуцирана с 4 байта.

Атрибути на Фаза 1.

При IKE операции във Фаза 1 съществуват няколко специфични атрибута за осигуряване на защитата на IKE, за разлика от атрибутите от Фаза 2, които се използват за създаване на IPSec SA. За защита на IKE съобщенията IKE изисква различна информация – от криптиращи алгоритми до тип удостоверяване.

Атрибутите са дефинирани в IKE RFC, а не в IPSec DOI или ISAKMP RFC. IKE атрибутите и техните типове са следните:

Encryption Algorithm	1
Hash Algorithm	2
Authentication Method	3
Group Description	4
Group Type	5
Group Prime/Irreducible Polynomial	6
Group Generator One	7
Group Generator Two	8
Group Curve A	9
Group Curve B	10
Life Type	11
Life Duration	12
PRF	13
Key Length	14
Field Size	15
Group Order	16

Атрибут тип 1, криптиране, притежава осем стойности за използване в IKE съобщения. Дефинираният криптиращ алгоритъм ще бъде използван за защита на съобщенията във Фаза 1 и Фаза 2 или други съобщения в IKE, изискващи защита. Възможните алгоритми са следните:

DES-CBC	1
IDEA-CBC	2
Blowfish-CBC	3
RC5-R16-B64-CBC	4

3DES-CBC	5
CAST-CBC	6
AES-CBC	7
Camelia-CBC	8

Атрибут тип 2, хеш алгоритъм, определя удостоверяващия алгоритъм, който да бъде използван за всички IKE размени. Възможните алгоритми от този тип са следните:

MD5	1
SHA	2
Tiger	3
SHA2-256	4
SHA2-384	5
SHA2-512	6

Атрибут тип 3, удостоверяване, се използва за предаване на удостоверяващия тип, който ще бъде използван в IKE. Това е изключително важно, тъй като съобщава на отсрещната страна какъв формат и структура на съобщението ще последва:

Pre-shared key	1
DSS signatures	2
RSA signatures	3
Encryption with RSA	4
Revised encryption with RSA	5

Типове от 4 до 10 се използват за задаване на параметри на алгоритъм Diffie-Hellman.

Тип 11, life тип, осигурява две прости стойности, а тип 12, продължителност на връзката в секунди или килобайтове в зависимост от стойността на тип 11.

Тип 13, псевдо-случайна функция, обикновено не се използва. IPSec и ISAKMP не притежават псевдо-случайна функция и за операции при тях се използва HMAC частта на процеса на удостоверяване на съобщението.

Key length се използва за определяне на дължината на ключа на криптиращ алгоритъм, който поддържа променливи ключове. Тази възможност не се използва за криптиращи алгоритми, които имат предварително определена дължина на ключа.

Размерът на полето се отнася за алгоритъма Diffie-Hellman.

Атрибути на Фаза 2.

Съобщенията във Фаза 2 на IKE, като Quick Mode например, се използват за определяне на IPSec SA и по тази причина притежават специфични атрибути за създаването им в IKE. При операции във Фаза 2 се наблюдават SA payload-и, които

съдържат proposals and transform payload-и за създаване на IPSec SA. Това изисква атрибути, различни от използваните във Фаза 1 на IKE, и създаването на IKE SA. Proposal (предложение) ще съществува за всеки конфигуриран протокол за сигурност и със съответни transform payloads. В proposal payload-а се дефинира типът на IPSec протокола, а в transform payload-а се дефинира transform типът. Например при IPSec VPN с ESP протокол с DES криптиране и MD5 удостоверяване предложението ще има ID на протокола, зададено на IPSEC-ESP, а transform ID на ESP-DES и атрибут payload, дефиниращ останалите детайли от SA, като HMAC MD5 за удостоверяване на съобщение.

При съобщения във Фаза 2 за създаване на IPSec SA съществуват девет възможни стойности:

SA Life Type	1
SA Life Duration	2
Group Description	3
Encapsulation Mode	4
Authentication Algorithm	5
Key Length	6
Key Rounds	7
Compress Dictionary Size	8
Compress Private Algorithm	9

Типовете Life Type и Продължителност на връзката са идентични с тези във Фаза 1. Тези атрибути дефинират времето за съществуване на създадената SA.

Group description има за цел да осигури възможност на Фаза 2 за създаване на нова Diffie-Hellman връзка.

Encapsulation mode определя дали протоколът за сигурност да работи в транспортен или тунелен режим.

Tunnel	1
Transport	2

Authentification Algorithm определя използвания протокол за удостоверяване:

HMAC-MD5	1
HMAC-SHA	2
DES-MAC	3
KPDK	4

Следващите три типа – 6, 7 и 8, обикновено нямат асоциирани стойности и в IPSec имплементациите винаги са със стойност 0.

Compress private algorithm е алгоритъм, който може да бъде договорен между страните.

2.4.3. Фаза 1.

SA, които защитават съобщенията в ISAKMP, се договарят и създават по време на обмените във Фаза 1. Процесът във Фаза 1 чрез три стъпки създава SA за по-нататъшни ISAKMP съобщения, необходими за установяването на IPSec SA. Фаза 1 трябва да договори защитната обвивка, която да бъде използвана по време на комуникацията между двете системи. По време на Фаза 1 трябва да бъдат разменени параметрите на Diffie-Hellman, за да се създадат ключове за криптиране и удостоверяване на данните в защитените комуникации в последната част на операциите на Фаза 1 и Фаза 2. Фаза 1 също така трябва да осигури удостоверяване на отдалечената система или потребител.

Трите стъпки във Фаза 1 са Main Mode, Aggressive Mode или Base Mode. Най-широко използваните режими са Main Mode и Aggressive Mode. Причината за това е, че те са дефинирани в IKE и ISAKMP RFC, докато Base Mode понастоящем е в процес на планиране. Основната разлика между режимите е в броя на разменените съобщения за установяване на SA. В Main Mode те са шест, в Aggressive Mode – три, а в Base Mode – четири.

2.4.3.1. Main Mode.

За създаване на SA Main Mode използва шест съобщения в три обмена. Стъпките обикновено започват с договаряне на SA, следвано от Diffie-Hellman и “nonce” (случайна генерирана стойност) обмени и завършват с удостоверяване на страната. Ще бъдат разгледани подробно двата случая на удостоверяване с предварително споделен ключ и удостоверяване със сертификати са показани на фигури 2.13. и 2.14.

	Инициатор		**Отговарящ**
Съобщение 1	**ISAKMP Header & SA Proposal and Sets**	→	
Съобщение 2		←	**ISAKMP Header & Accepted SA Proposal and Set**
Съобщение 3	**ISAKMP Header, KE & Nonce**	→	
Съобщение 4		←	**ISAKMP Header, KE & Nonce**
Съобщение 5	**ISAKMP Header, ID_i & HASH**	→	
Съобщение 6		←	**ISAKMP Header, ID_r & HASH**

Фигура 2.13. Main mode – удостоверяване с предварително споделен ключ.

	Инициатор		**Отговарящ**
Съобщение 1	**ISAKMP Header & SA Proposal and Transformsets**	→	
Съобщение 2		←	**ISAKMP Header & Accepted SA Proposal and Transformset (only one)**
Съобщение 3	**ISAKMP Header, KE, Nonce, & optional Certificate Request**	→	
Съобщение 4		←	**ISAKMP Header, KE & NonceISAKMP Header, KE, Nonce, & optional Certificate Request**
Съобщение 5	**ISAKMP Header, ID_i, Signature, & optional Certificate**	→	
Съобщение 6		←	**ISAKMP Header, ID_i, Signature, & optional Certificate**

Фигура 2.14. Main mode – удостоверяване със сертификати.

Използването на предварително споделени ключове за удостоверяване налага всяка система да бъде конфигурирана с парола, която е асоциирана с IP адресите на удостоверяваните отдалечени системи.

Първи обмен.

Първият обмен започва със съобщение от инициатора до отсрещната страна, което съдържа ISAKMP хедър и SA хедър. В SA хедъра се съдържат съответните proposal и transform payload-и, които представят на отсрещната страна опциите за сигурност. В оригиналното съобщение от инициатора е включена “бисквитка”.

Второто съобщение е от отсрещната страна. То съдържа ISAKMP хедър и SA хедър, който от своя страна съдържа proposal и transform payload с параметрите, които ще се използват. Включени са “бисквитките” на отсрещната страна.

На този етап двете системи са се договорили за параметрите за сигурност, които ще бъдат използвани за защита на по-нататъшната комуникация, и са обменили “бискивитки”, които намаляват възможността атакуващ да вмъкне невалидни пакети по време на преговорите в IKE. След завършване на първия обмен всяка система генерира публичните стойности на Diffie-Hellman, които се споделят с отсрещната страна.

Втори обмен.

Следващата част от комуникацията позволява на страните да обменят публичните стойности на Diffie-Hellman и “nonce”. Nonce и “бисквитките” формират защитен слой и уверяват, че страната в комуникацията е активно участваща, а не само повтаряща прихванати по-рано съобщения или отправяща фалшиви запитвания. След завършване на втория обмен всяка система обработва стойностите на Diffie-Hellman с цел създаване на основен ключ. Всяка система създава четири ключа:

1. SKEYID, таен ключ, на който се основават всички следващи ключове. Ако обаче PFS (perfect forward secrecy- политика) е активирана, този ключ не се използва за създаване на бъдещи ключове, а се създава нов ключ в нова Фаза 1 и Фаза 2.

2. SKEYID_d се използва като ключов материал за създаване на ключове за SA в операции във Фаза 2 (т.е. за IPSec).

3. SKEYID_a е ключът, използван за удостоверяване и интегритет на данни.

4. SKEYID_e е ключът, използван за криптиране на IKE съобщения.

Ключовете, които фактически се използват при криптирането и удостоверяването, се базират на основния ключ, SKEYID. Атрибутите, които се използват за създаване на SKEYID, зависят от използвания удостоверяващ метод.

За изграждане на ключове се използва псевдо-случайна функция (prf) с ключ и различни данни, които се комбинират за създаване на фиксирана стойност. Prf е заключена хеш функция, която се използва за генериране на стойности, които изглеждат случайни, но се използват за валидиране на данните. Пример за този процес е:

Digest = prf (key/seed, data1 | data2 | data3)

Тази секция представлява детайлизиране на предварително споделено тайно удостоверяване, при което SKEYID се определя като:

SKEYID = prf (pre-shared key, Nonce_i | Nonce_r)

Информацията с която разполага всяка система за другата след първия обмен е:

Инициатор/ отсрещна страна (отговарящ)

1 – “Бисквитки” на инициатора

2 – “Бисквитки” на отсрещната страна (отговарящия)

3 – Предложения пакет за сигурност (ESP-DES и ESP MD5-HMAC).

При удостоверяване с цифрови сертификати ключът се определя, както следва:

SKEYID = prf (Nonce_i | Nonce_r, DH_key), където Nonce_i и Nonce_r са случайни стойности, а DH_key е е стойността, произлизаща от процеса Diffie-Hellman и е резултат от споделянето на публичните стойности на Diffie-Hellman. Тази стойност не е публична, а е създадена независимо от всяка страна чрез използване на публична стойност, предоставена от отсрещната страна.

SKEYID_d = prf (SKEYID, DH_key | Cookie_i | Cookie_r | 0)

SKEYID_a = prf (SKEYID, SKEYID_d | DH_key | Cookie_i | Cookie_r | 1)

SKEYID_e = prf (SKEYID, SKEYID_a | DH_key | Cookie_i | Cookie_r | 2)

Другите три ключа, създадени с използването на основния ключ SKEYID, са резултат от същия детайлизиран метод, независимо от избрания удостоверяващ метод.

При завършването на втория обмен всяка система разполага с достатъчно информация, за да започне криптирането на комуникацията. Важно е да се отбележи, че ключовете, които ще бъдат използвани за криптиране и удостоверяване, са създадени единствено на базата на IP адреса на отсрещната страна. Identification Payload предоставя възможност за удостоверяване на информационния обмен, като IP адреси, маски или “fully qualified domain name” (FQDN). ID payload-ът не се споделя по време на първия и втория обмен, което означава, че IP адресът на страната остава единствен уникален идентификатор.

Main Mode на Фаза 1 с предварително споделена тайна, като удостоверяващ процес, не предоставя защита и се основава изцяло на IP адреса на страната. Това е едно прието ограничение на Main Mode с предварително споделена тайна и в много случаи не представлява проблем. В случаи обаче, когато IP адресът е динамичен, отговарящият не може да поддържа предварително споделени тайни, базирани на IP адреси, които не познава. Решенията за отдалечен достъп са пример за това, че IP

адресът на инициатора може да бъде различен за всяка връзка. За да отговорят на тези случаи, се използват сертификати за удостоверяване на отдалечен достъп. Повечето решения за отдалечен достъп използват главно Aggressive Mode, защото ID payload-ът се осигурява по време на първия обмен.

Трети обмен.

Последният обмен се състои от ISAKMP хедър, ID payload и HASH payload за удостоверяване на произхода на данните. Той е криптиран с ключ SKEYID_e.

HASH обединява няколко компонента, които вече са известни на страните и един, който се осигурява при петото и шестото съобщение - ID payload. Той се състои от:

HASH = prf (SKEYID, Ya | Yb | Cookie_i | Cookie_r | SA offer | ID_i)

При използване на сертификати в третия обмен вместо HASH се предават съответните удостоверяващи сертификати.

2.4.3.2. Aggressive Mode.

Целта на Aggressive Mode е същата, както на Main Mode с тази разлика, че той завършва транзакцията за два пъти по-малко време от Main Mode. Въпреки че Aggressive Mode е по-бързият от двата режима, разходите се изразяват в невъзможността за договаряне на различни опции. Поради обстоятелството, че повечето payload-и се изпращат в първите две съобщения, не съществуват механизми за селекция или потвърждаване на отделни параметри. Например, публичният ключ на Diffie-Hellman се изпраща в същото съобщение, в което се изпраща и SA payload-ът.

Aggressive Mode не е предназначен за защита на идентичността, което се постига в Main Mode чрез криптиране на ID payload-а. Aggressive Mode не криптира ID payload-и и поради тази причина информацията е на публично разположение.

Aggressive Mode притежава няколко ограничения, но притежава също така и много предимства. На първо място, наблюдава се по-малък обмен на съобщения, който се отразява върху по-малката натовареност на мрежата и по-бързия отговор.

Aggressive Mode изпраща голяма част от необходимата информация за установяването на IKE SA в първите съобщения, които включват ID payload. Като не се предоставя защита на идентичността, ID payload може да бъде изпратен в чист вид.

Съдържащата се информация позволява на отговарящия да провери паролата, която се основава на характеристики, различни от IP адреса на инициатора (т.е. потребителското име). В случаи, при които IP адресът на инициатора не е постоянен, като при отдалечен достъп на потребителя, това осигурява средства за позволяване на удостоверяването. Поради тази причина Aggressive Mode се използва предимно за отдалечен достъп, за разлика от Main Mode.

Излагането на ID payload-а представлява заплаха за защитата на сесията.

За да се намали излагането на потребителското име в ID payload-а – използвайки типа ID_USER_FQDN, съществува идентификационен тип, наречен ID_KEY_ID. ID_KEY_ID, съгласно определението в IP Security Domain of Interpretation (RFC 2407), представлява неразбираема поредица от байтове, които се използват за предаване на специфична за производителя информация, необходима за определяне коя предварително споделена тайна ще се използва.

Отделна част от информацията може да бъде получена от пакета на клиента за последваща идентификация на индивида или на системата, инициираща VPN. Тъй като тази информация е крайно неопределена, на атакуващия ще са му необходими много повече стъпки, за да извлече информация с някаква стойност.

	Инициатор		**Отговарящ**
Съобщение 1	**ISAKMP Header, SA Proposal & Transformsets, KE, Nonce, and ID payload**	→	
Съобщение 2		←	**ISAKMP Header, SA Proposal & Transformsets, KE, Nonce, ID payload, and HASH_r and Set**
Съобщение 3	**ISAKMP Header and HASH_i**	→	

Фигура 2.15. Aggressive Mode - удостоверяване с предварително споделен ключ.

	Инициатор		**Отговарящ**
Съобщение 1	**ISAKMP Header, SA Proposal & Transformsets, KE, Nonce, and ID payload**	→	
Съобщение 2		←	**ISAKMP Header, SA Proposal & Transformsets,KE, Nonce, ID payload,Signature, and Optional Certificate**
Съобщение 3	**ISAKMP Header, optional Certificate, and Signature**	→	

Фигура 2.16. Aggressive Mode - удостоверяване със сертификати.

Използването на Aggressive Mode е ограничено в сравнение с Main Mode, но може да бъде много полезно при решения за отдалечен достъп, където IP адресът на инициатора не е известен и формата за удостоверяване е чрез предварително споделена тайна.

2.4.4. Фаза 2.

SA, създадена по време на Фаза 1, се използва за защита ISAKMP съобщенията от Фаза 2, които служат за създаване на множество IPSec SA. Операциите във втората фаза са подобни на обмяната на съобщения и на режимите от Фаза 1 с тази разлика, че на удостоверяването се отдава по-малко значение, като се приема, че съобщенията се обменят по удостоверена IKE SA. За създаване на SA за IPSec във Фаза 2 се използват два основни режима. SA се изграждат и имплементират чрез използване на протоколите за сигурност ESP и AH. Начините за имплементиране на протоколите за сигурност, дали в тунелен или транспортен режим, зависи от изискванията на VPN и политиката, дефинирана от администратора. Първата фаза от IKE ISAKMP се използва за създаване на една SA, която да позволи на по-нататъшната комуникация да бъде защитена. Фаза 2 от IKE използва структурата на съобщенията на ISAKMP, за да предостави ключовете и характеристиките за конкретна IPSec VPN.

2.4.4.1. Quick Mode.

Quick Mode се използва само във Фаза 2 и не съществува във Фаза 1, защото всички предавани данни трябва да бъдат под защитата на IKE SA. SKEYID_a, генериран в първата фаза и свързан с основния SKEYID, се използва за удостоверяване на всички съобщения в Quick Mode, а SKEYID_e се използва, за да ги криптира. Ако първата фаза бъде преодоляна, Фаза 2 ще бъде напълно изложена на атаки.

Quick Mode отговаря за имплементирането на пакета за сигурност в първите съобщения на Фаза 1 без значение от използвания режим. ISAKMP хедърът и съпътстващия го SA payload се използват за договаряне на детайлите на защитата - transformset. Договорените атрибути се използват не само във Фаза 1, но и в Quick Mode. По тази причина избраните удостоверяващи и криптиращи алгоритми ще бъдат използвани за Фаза 2. Съобщенията в Quick Mode са защитени чрез ключове,

дефинирани във Фаза 1, но се използват за дефиниране на ключове за защита на протоколите за сигурност на IPSec.

Quick Mode осигурява механизъм за създаване на SA за защита на специфична за политиката информация в IPSec. Например съгласно политиката на IPSec трафикът на определен тип може да бъде защитен с 3DES криптиране и SHA удостоверяване, докато на всички други данни е предоставено само DES криптиране. За да се изпълнят тези изисквания, трябва да бъдат създадени множество SA. Поради тази причина Quick Mode ще договаря нови SA от името на IPSec политиката. В предишния пример трябва да бъдат създадени минимум четири SA, всяка от които трябва да има една входяща и една изходяща SA. За да се гарантира, че установените SA са валидни и новите ключове са защитени, отново се използват „nonces", за да може страната да се увери в съществуването и активността на комуникацията.

Поради обстоятелството, че в Quick Mode могат да се изпълняват едновременно множество обмени на съобщения, ID на съобщението в ISAKMP хедъра се използва за мултиплексиране на съобщенията в Quick Mode. Всяко ID на ISAKMP хедъра се използва за идентифициране на асоцииираното договаряне в Quick Mode, а "бисквитките" на хедъра се използват за идентифициране на състоянието на съобщенията. Предполага се, че ID на страната в Quick Mode е IP адресът на страните във Фаза 1. Това е очевидно предположение, защото всички съобщения в Quick Mode са защитени от ключове, създадени между тези две страни. IKE обаче не притежава толкова ограничения и предлага на клиента ID payload-и, които не са нищо друго освен незадължителни ID payload-и, предавани между страните. ID payload-ите не служат за предоставяне на разширена удостоверяваща информация, както FQDM във Фаза 1. ID-тата на клиента се използват за осигуряване на информация, използвана от SPD и SAD в IPSec. Тази възможност може да бъде използвана за подобряване на филтриращите способности и приложението на пакетите за сигурност, дефинирани от политиката на шлюза за сигурност.

Селектор информация се предава във формата на ID payload на клиента в същото съобщение, което съдържа SA payload-и . Чрез комбиниране на SA предложенията със селектор информацията, се постига прилагане на политиката на комуникация. Отговарящият може да прегледа информацията в своята собствена SPD и да провери дали SA е договорено в подходяща и одобрена форма на комуникация,

базирана на конфигурираната политика. Ако SPD тестовете за проверка са успешни и заявката за комуникация е валидна, когато е отнесена към политиката, ID payload-ите на съобщението и приетото предложение трябва да бъдат предоставени на SAD. Следователно SAD и SPD ще гарантират, че идентифицираният от ID на съобщението трафик ще преминава само по конфигурирана SA и ще позволят само специфичен трафик, дефиниран от селекторите, идентифицирани в ID payload-а.

Едновременно с възможността да се използват ID payload-и на клиента, страните имат възможност да използват и PFS. Ако това се осъществява в съответствие с изискванията на политиката, публичните стойности на Diffie-Hellman трябва да бъдат разменени.

Основни обмени. Пример за обмяна на съобщения в Quick Mode е показан на фиг.2.17. Важно е да се отбележи, че всички съобщения са криптирани и удостоверени чрез използване съответно на SKEYID_e и SKEYID_a.

	Инициатор		**Отговарящ**
Съобщение 1	**ISAKMP Header, HASH(1), SA Proposal and Transformsets, Nonce_i, Optional KE, CID_i, and CID_r**	→	
Съобщение 2		←	**ISAKMP Header, HASH(2), SA Proposal and Transformsets, Nonce_r, Optional KE, CID_i, and CID_r**
Съобщение 3	**ISAKMP Header & HASH(3)**	→	

Фигура 2.17.

Съществуват няколко идентификатора, асоциирани с HASH payload-ите. Всяко съобщение съдържа различни характеристики, които да бъдат включени в хеша (HASH) и които зависят от това коя опция е избрана за използване. Пример за необходимостта от различно съдържание на хешовете (HASH) е имплементацията на PFS и включването в съобщението на KE. Така също, ако трябва да бъде създадена повече от една SA, може да бъде добавен SA payload. Следователно за всеки SA payload ще бъдат създадени две IPSec SA, които са базирани на характеристиките, договорени в оригиналните SA payload-и в Quick Mode.

Обяснението на различните HASH payload-и е следното (опциите са показани със знака “( )”). HASH(1)—prf (SKEYID_a, M-ID | SA offer | Nonce_I | (KE) | (CID_I) | (CID_r)).

Съдържанието на HASH(1) е точно такова, каквото е и в съобщението. Необходимо е само да се постави HASH(1) в хеша на пълното съобщение.

HASH(2)—prf (SKEYID_a, M-ID | Nonce_I | SA offer | Nonce_r | (KE) | (CID_I) | (CID_r)).

Подобно на HASH(1), HASH(2) покрива цялото съдържание на съобщението, само че е включено „nonce” на инициатора.

HASH(3)–prf (SKEYID_a, 0 | M-ID | Nonce_I | Nonce_r).

Третият тип HASH включва само „nonces” и ID на съобщението. Това е така, за да се гарантира съществуването на комуникацията преди започването на IPSec трафика. Без доказателство за активното участие на страната, атакуващият може да използва предварително получени пакети, идентифицирани като Quick Mode, и да приложи повторна атака (replay atack), консумирайки ресурси от отговарящия и накрая постигайки атака “отказ от услуга” (denial-of-service attack).

“M-ID” е ID на съобщението от ISAKMP хедъра, което идентифицира обмените на съобщения в Quick Mode за специфична SA. Както във Фаза 1, KE е публичният ключ на Diffie-Hellman, използван между страните за създаване на симетричен ключ за криптиране в протоколите за сигурност на IPSec. CID_I и CID_r са ID payload-и на клиента. Важно е да се отбележи използването на SKEYID_a в HASH за удостоверяване на данните.

Формиране на ключовете за IPSec.

SKEYID_d се създава във Фаза 1, за да се използва за генериране на ключове за криптиране на операциите във Фаза 2, които не са част от IKE. Ключовете за IPSec SA се състоят от няколко компонента от първите две фази. В Quick Mode се договаря SA, която се състои от две SA – по една за всяка посока. Те имат уникален SPI, а в резултат на това, и уникален ключ за всяка SA.

Информацията, използвана за създаване на ключове за IPSec, е зависима от опциите в първоначалната размяна на Quick Mode. Ако е конфигурирано използването на PFS, между двете страни трябва да се споделят стойностите на KE, поради което липсата или използването на публичен ключ на Diffie-Hellman ще

определи и ключовете за криптиране, използвани в IPSec, както е посочено във формулите:

KEY = prf (SKEYID_d, protocol | SPI | Nonce_i | Nonce_r) – без използване на PFS.

KEY = prf (SKEYID_d, DH_key(QM) | protocol | SPI | Nonce_i | Nonce_r) – с използване на PFS.

Ако PFS е активиран, двете страни ще трябва да изчислят нова стойност на Diffie-Hellman ключовете за операциите в Quick Mode.

По време на ISKAMP инициаторът предлага едно или няколко предложения чрез съпътстващи transformsets. Отговарящият трябва да избере само едно от предложенията. Затова към всяко предложение съществува поле SPI и protocol, което позволява на ключа да бъде свързан със SA, за която е създаден. Всяка страна определя SPI за своята входяща SA. Като резултат се договарят две SA и се създават ключове за всяка страна, които са необходими за работата на процесите за сигурност за двете SA.

Инициаторът на Quick Mode може да бъде всяка една от страните. Не е задължително това да е инициаторът на режима от Фаза 1.

Други обмени в Quick Mode.

Всички описани до този момент режими и обмени се отнасят до идентификацията, удостоверяването и договарянето на протоколи за сигурност за защитата на IKE съобщения и IPSec комуникации. След стартирането на IPSec операции и завършването на операции в Quick Mode, IKE SA повече не се използва за предаване на още допълнителни данни. След установяване на IKE SA и завършване на Quick Mode много малък трафик използва SA.

2.4.4.2. New Group Mode.

New Group Mode е предназначен да позволи на страните, участващи в IKE, да предоговорят Diffie-Hellman групата, установена във Фаза 1. Обменът се базира на заявка и отговор, подобно на договарянето, което използва SA payload. Инициаторът изпраща една или повече заявки за използване на “нова група” от параметри на Diffie-Hellman, а отсрещната страна отговаря с приетата версия.

	Инициатор		**Отговарящ**
Съобщение 1	**ISAKMP Header, HASH(NG1), and SA Proposal**	→	
Съобщение 2		←	**ISAKMP Header, HASH(NG2), and SA Proposal**

Фигура 2.18.

На фигура 2.18. са показани съобщенията, разменяни в режима New Group.Те се състоят от ISAKMP хедър, хеш и SA payload, съдържащ детайлите на Diffie-Hellman групата. HASH(NG1) и HASH(NG2) се базират на SKEYID_a, генериран във Фаза 1 и както беше казано по-рано се използват за удостоверяване на всички IKE съобщения

HASH(NG1) = prf (SKEYID_a, Message ID | Entire SA proposal, header and payload)

HASH(NG2) = prf (SKEYID_a, Message ID | SA reply)

Недостатъци на IPSec.

IPSec е предназначен да осигури IP връзки между устройства. Той изпълнява функцията си много добре, но съществуват някои ограничения, най-важните от които са:

- IPsec не може да бъде сигурен, ако системата не е сигурна. Сигурността на системата, на която е инсталиран IPsec и която служи за шлюз, е задължително изискване. От друга страна IPsec може да бъде средство за подобряване на сигурността на системата и мрежата.
- IPsec не е от типа „край към край”(end-to-end). Той не може да предостави същата сигурност, както системите работещи на по-високо ниво. IPsec криптира една IP връзка между две машини, което е нещо различно в сравнение с криптирането на съобщения между потребителите или приложенията.Например IPsec криптира пакети на машина, използвана за шлюз за сигурност, когато те напускат машината на подателя, и ги декриптира при пристигане при шлюза на машината на получателя. Това не предоставя полезна услуга за сигурност – само криптирани данни се предават през Интернет – но дори не се доближава до осигуряването на услуга от типа „край

към край”. По-специално всеки, който притежава привилегии на някоя от страните на LAN може да засече съобщение в некриптирана форма.

- IPsec не може да прави всичко. Той не предоставя всички функции на системите, работещи на по-високо ниво. Ако е необходимо един документ да бъде подписан електронно от определено лице, тогава се използва цифров подпис и криптографска система на публичен ключ, с който подписът да бъде проверен.

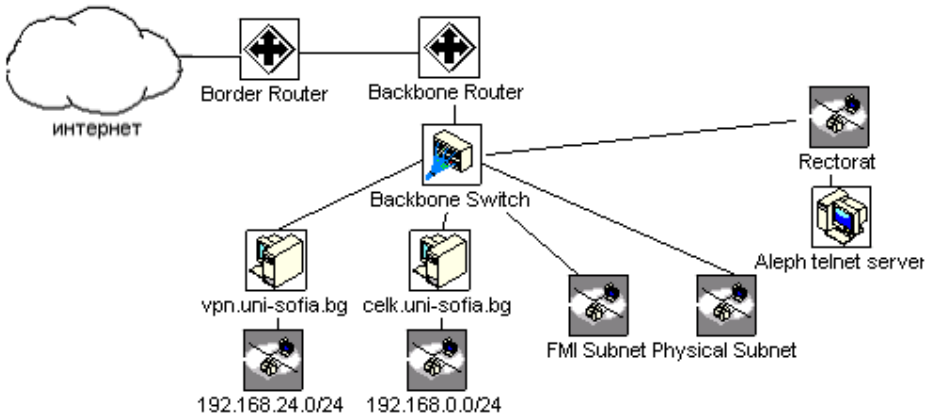
IPsec удостоверяване на основна комуникация може значително да затрудни атаките на протоколи от по-горно ниво. В частност то предотвратява атаки от типа „man-in-the-middle”.

- IPsec удостоверява машини, а не потребители. IPsec използва силни удостоверяващи механизми за контрол на съобщенията, но не притежава концепцията на ID на потребителя, която е от важно значение за много други механизми и политики за сигурност. IPsec контролира кои машини се свързват към сървъра и може да гарантира, че трансферът на данни до тези машини е осъществен сигурно, но това е всичко. В този случай или самите машини трябва да контролират достъпа на потребителите, или трябва да има форма на удостоверяване на потребителите към базата данни, независима от IPsec.

- IPsec не спира анализа на трафика (traffic analysis). Анализът на трафика представлява опит за извличане на информация от съобщения без значение от тяхното съдържание. В случая на IPsec това ще означава анализ на данни, видими в некриптираните хедъри на криптираните пакети – размерът на пакетите на адресите на gateway на източника и местоназначението и др.

3. Описание и конфигуриране на Openswan VPN сървър и клиент.

Целта на настоящата дипломна работа е да се конфигурира VPN сървъра на СУ за работа с различни клиенти, за които е необходимо да имат достъп до мрежата на СУ. Мрежата на СУ- SUNet представлява мрежа от клас С с адрес – 62.44.96.0/19.



Фигура 3.1. Част от мрежата на Софийския университет - SUNet

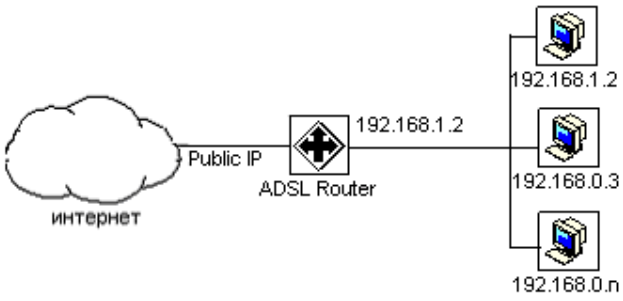
Мрежата е разделена на различни звена, като съществуват мрежи изцяло с публични адреси, като например мрежата на ФМИ и на Физическия факултет, както и мрежи с частни адреси зад NAT маршрутизатор, като например мрежата на Центъра за източни езици и култури (ЦИЕК) - celk.uni-sofia.bg с публичен адрес 62.44.96.183 и частна мрежа 192.168.0.0/24.

Ще бъдат разгледани два основни случая:

- Свързване във VPN на индивидуални потребители с операционните системи Windows 2000/XP.
- Свързване във VPN на частна мрежа, намираща се зад маршрутизатор с Линукс сървър към мрежата на СУ. Тук ще бъдат разгледани два варианта: когато сървърът има публичен IP адрес и когато сървърът е с частен адрес за връзка към Интернет - зад NAT (Network Address Translation - Транслиране на адреси в мрежата, според определението в RFC 1631). Такав е случаят с Интернет достъпа, предлаган от БТК ADSL и голяма част от другите Интернет доставчици.

Услугата ADSL на БТК се осъществява чрез ADSL маршрутизатор, който конвертира ADSL сигнала по комутируемата телефонна или ISDN линия в използваема връзка с Интернет, представена като 10BaseT RJ45 Ethernet.

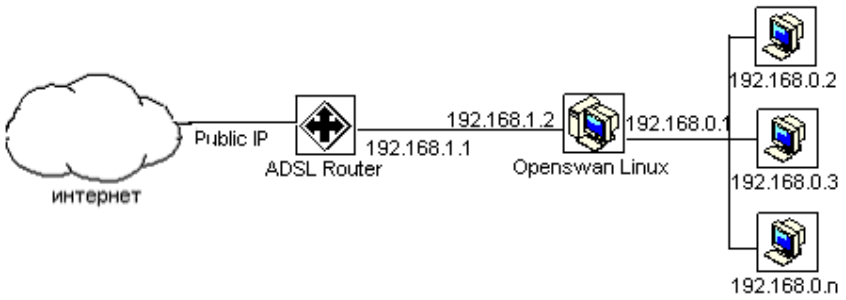
Устройството е конфигурирано да предоставя динамичен NAT между външните (WAN) и вътрешните (LAN) интерфейси. Това обстоятелство притежава две основни предимства: позволява на потребителя да предоставя услуга на поредица хостове без да се съобразява с правилата за присвояване на IP адреси на IANA/ RIPE, като предоставя също и добра степен на сигурност без необходимост от защитна стена и/или прокси. Обикновено при включване на локална мрежа към ADSL на БТК за компютрите от мрежата се определя поредица от "частни" IP адреси чрез DHCP от вътрешния (LAN) интерфейс. Поредицата от "частни" IP адреси са адреси, които са резервирани за използване в частни мрежи според определението в RFC 1918 и не се маршрутизират в по-широкообхватния Обществен Интернет. Поредицата от частни адреси е от "Клас С" 192.168.1.x. Това позволява до 253 хоста в локалната мрежа.Фигура 3. Заради използването на NAT при БТК ADSL е необходимо използването на IPSec VPN софтуер, който поддържа техниката за енкапсулиране NAT-T, дефинирана в RFC-3942 и RFC-3948 в началото на 2005 година.Чрез тази техника се дава възможност на клиенти зад NAT да могат да използват протокола IPSec, като ESP и AH пакетите се енкапсулират в UDP пакети използвайки порт 4500 и по този начин IPSec пакетите остават непроменени при NAT маскирането.



Фигура 3.2 Свързване на частна мрежа към интернет през ADSL маршрутизатор

Главната задача на дипломната работа е осигуряване на достъпа на мрежата на Астрономическата обсерватория до частната мрежа зад VPN сървъра vpn.uni-sofia.bg. Има два начина, по които може да се реализира тази задача. Първият е чрез свързване на всеки един от компютрите в мрежата на астрономическата обсерватория към SUNet чрез VPN връзка от типа "хост-към-мрежа". Вторият начин е чрез инсталиране на допълнителен компютър маршрутизатор зад този на БТК- ADSL, на който да бъде

инсталиран IPSec VPN софтуер фигура 3.. По този начин ще се реализира VPN връзка от типа “мрежа-към-мрежа”.



Фигура 3.3 Свързване на частна мрежа през Openswan Линукс маршрутизатор и ADSL към интернет.

Друго приложение на VPN е свързването на Центъра за славяно-византийски проучвания “Иван Дуйчев” с библиотека от услуги, предоставяни на телнет порт 23 от сървъра Aleph 62.44.11.2, намиращ се в SUNet. Въпреки че сървърът е с публичен адрес, достъпът до порт 23 от Интернет е спрян от съображения за сигурност от Border маршрутизатора. Чрез реализиране на тунел към VPN сървъра, който се намира зад маршрутизатора, може да се направи заобикаляне на ограничението за достъп до телнет. Центърът “Иван Дуйчев” също ползва БТК- ADSL за достъп до Интернет както при астрономическата обсерватория, така че проблемите в този случай са аналогични.

Съществуват имплементации на избрания протокол за VPN IPSec за различни операционни системи. IPSec е част от Windows 2000, Windows XP и Windows 2003 Server, а за Линукс съществуват Openswan и Strongwan - наследници на първия реализиран IPSec пакет за Линукс Frees/WAN.

Експериментите в дипломната работа ще бъдат провеждани на VPN сървъра vpn.uni-sofia.bg, който е с операционната система Линукс Fedora Core 5 и представлява компютър, играещ ролята на шлюз между експериментална частна мрежа в СУ-SUNet с адрес 192.168.24.0/24 и Интернет. Това налага избора на Линукс базирано IPSec решение.

За конкретната реализация е избран продуктът Openswan. Openswan е имплементация на протокола IPSec за Линукс. Той работи на различни платформи и предлага всички основни методи за криптиране и удостоверяване, включени в стандарта IPSec. Ценното в този продукт е, че позволява работа и в мрежа с частни IP

адреси, скрита зад NAT шлюз, необходим заради клиенти с достъп до Интернет от БТК ADSL. Той поддържа и удостоверяване чрез X.509 сертификати и има възможност да работи с Линукс ядро 2.6., за разлика от другия наследник на Frees/WAN – Strongwan, който работи само с ядра 2.4. Друго съществено предимство е това, че се разпространява безплатно под GNU General Public License.

Openswan имплементацията на IPSec притежава три основни части:

- KLIPS (kernel IPSec) - имплементира AH, ESP и пакет, работещ в рамките на ядрото.
- Pluto (IKE daemon) - имплементира IKE, договаряйки връзки с други системи.
- Различни скриптове осигуряват администраторски интерфейс на машината.

3.1. Конфигурационен файл ipsec.conf. Описание на параметрите на Openswan.

Двата файла, които се използват за конфигурация на Openswan VPN, са файловете /etc/ipsec.secrets, съдържащ ключовете за удостоверяване, и /etc/ipsec.conf, съдържащ конфигурациите.

Файлът ipsec.conf от своя страна се състои от две части - основна конфигурация **config setup,** и секция с конкретните връзки, зададени чрез **conn.**

Секция *CONN.*

Секцията CONN съдържа спецификация на връзката, задавайки необходимите параметри, за да бъде осъществена посредством протокола IPSec. Името на връзката може да бъде произволно.

Например при наличие на две частни мрежи – лява с адрес 172.16.0.0/24 и дясна с адрес 10.0.0.0, които трябва да се свържат във виртуална частна мрежа през Интернет чрез използване на Openswan. За това са необходими две Линукс машини, които служат като защитна стена и маршрутизатор и съответно разпределят Интернет към другите компютри от вътрешната мрежа. Те трябва да имат инсталиран Openswan. Необходимо е също така да знаем външните IP адреси на двата gateway-а.

Конфигурирането на връзката се задава във файла /etc/ipsec.conf.

conn left-to-right	име на връзката
left=192.0.2.2	външен IP адрес на Линукс в Ляво
leftsubnet=172.16.0.0/24	вътрешна мрежа на Линукс в Ляво

leftnexthop=1.1.1.1	gateway на Линукс в Ляво
right=2.2.2.2	външен IP адрес на Линукс в Дясно
rightsubnet=10.0.0.0/24	вътрешна мрежа на Линукс в Дясно
rightnexthop=2.2.2.1	gateway на Линукс в Дясно
auto=add #	автоматично добавяне на връзката при стартиране на системата

Openswan поддържа 2 вида размяна на ключове - manual и auto (ръчен и автоматичен). При automatic keying (автоматичното указване на ключа) комуникацията преминава през две фази: предаване на IP пакети между отделните потребители и “gateway-to-gateway” преговаряне за ключовете и общия контрол.

За да се избегне тривиалното редактиране на конфигурационните файлове за постигане на съвместимост с всяка система, въвлечена във връзката, вместо от гледна точка на локална и отдалечена страна, спецификациите на връзката се подготвят от гледна точка на левия и десния участник. Изборът на това кой от участниците ще бъде ляв или десен е произволен и се определя от IPSec на базата на вътрешна информация. Това позволява използването на идентични спецификации на връзката в двата края.

Много от параметрите на връзката се отнасят до единия или другия участник. Тук ще бъдат разгледани само тези, отнасящи се до левия участник, но всеки параметър, чието име започва с **left** има съответен **right** параметър.

Параметрите са незадължителни, освен ако са маркирани като задължителни; един параметър, задължителен за ръчното указване на ключа, няма нужда да бъде включен във връзка, която ще използва само автоматично указване на ключа и обратно.

Общи параметри на CONN.

Описаните параметри се отнасят и за двата метода на удостоверяване - чрез автоматично указване на ключа и чрез ръчно указване на ключа. За да работи една мрежова връзка, е необходимо и двете страни да имат еднакви настройки на параметрите.

type - определя типа на връзката. Валидни стойности са **tunnel,** която е стойност по подразбиране. Използва се при връзки от типа „хост към хост”, „хост към подмрежа” и „подмрежа към подмрежа”. Стоността **transport** указва „хост към хост” транспортен режим.

left - този параметър е задължителен и указва публичния IP адрес на мрежовия интерфейс на левия участник във връзката. Ако стойностите **%defaultroute** и **interfaces=%defaultroute** са използвани в config секцията, параметърът left ще бъде попълнен автоматично с локалния адрес на интерфейса за маршрутизиране по подразбиране , което ще отмени стойности, зададени чрез leftnexthop. Както left, така и right могат да имат стойност %defaultroute, но не и двете едновременно. Стойността %any позволява IP адресът да бъде зададен по време на договаряне при автоматично указване на ключа.

leftsubnet - определя частната подмрежа зад левия участник във формат network/netmask.

leftnexthop - този параметър задава gateway IP адреса на левия участник към публичната мрежа.

leftupdown - определя кой скрипт да бъде стартиран за настройване на маршрутизирането и защитната стена при промяна на статуса на връзката.

Leftfirewall - определя дали левият участник използва защитна стена (включително маскиране за трафик от *лявата подмрежа*, което след установяване на връзката трябва да бъде изключено (за трафик към другата подмрежа). Валидни стойности са **yes** и (по подразбиране) **no.**

Параметри на секцията CONN при автоматично указване на ключа.

Описаните параметри се отнасят само при автоматично указване на ключа и се игнорират при ръчното указване на ключа. Двете страни трябва да се договорят за еднакви стойности на тези параметри.

keyexchange - задава метода на размяна на ключове. Стойността по подразбиране и единствената валидна стойност е **ike**.

auto - определя коя операция ще бъде осъществена автоматично при стартирането на IPSec. Понастоящем валидните стойности са **add** (изразявайки **ipsec auto --add**), **route** (**ipsec auto --route**), **start** (**ipsec auto --up**) и **ignore** (също по подразбиране) (означаващо автоматично стартиране на операцията). Този параметър се игнорира, освен ако конфигурационният параметър **plutoload** или **plutostart** е съответно зададен. Приложим само локално, не е необходимо съгласието на другата страна (но за връзка, създадена с намерение да бъде постоянна, двете страни трябва да

използват **auto=start,** за да се гарантира, че всяко стартиране води до незабавно предоговаряне).

auth - определя дали удостоверяването да бъде част от **ESP** криптирането или отделно чрез протокола **AH**. Възможни стойности са **esp** и **ah.**

authby - определя как двата gateway се удостоверяват един друг. Валидни стойности са **secret** при използване на споделена тайна и **rsasid** при rsa електронни подписи.

leftid - определя как левият участник идентифицира себе си при удостоверяване. Стойност по подразбиране е **left**, но може да бъде IP адрес или FQDN, предшествано от @.

leftrsasigkey - задава публичния ключ за RSA удостоверяване съгласно RFC 2537.

leftrsasigkey2 - задава втори публичен ключ, ако такъв е необходим.

pfs - определя дали да се използва Perfect Forward Secrecy при размяна на ключовете (с pfs при пробив в протокола за размяна на ключове не се излагат на риск ключове, договорени по-рано). Валидни стойности са **yes** и **no**.

keylife - определя колко време дадена връзка (множество от криптиращи и удостоверяващи ключове) ще продължи от момента на успешното й установяване до прекратяването й. Стойностите са или в секунди, или в число, следвано от m, h или d, съответно за минута, час или ден.

rekey - определя дали връзката да бъде установена отново при изтичане на времето на съществуването й.

rekeymargin - определя колко време преди прекратяване на връзката да започне договарянето на нови параметри. Стойностите са като при keylife.

rekeyfuzz - задава максималния процент, по който **rekeymargin** трябва случайно да увеличи интервала на установяване на ключове. Това е много важно при хостове, поддържащи много връзки. За стойности се приемат цели числа, като стойност по подразбиране е 100%, но може да бъде и повече.

keyingtries - задава броя на опитите, които да се направят при установяване на връзка или замяна на разпаднала се такава. Стойност по подразбиране е 3, а при стойност 0 се правят безкраен брой опити.

ikelifetime - определя за колко време да бъде предоговорен каналът за установяване на ключове на дадена връзка. Стойност по подразбиране е 1 час, а максимумът е 8 часа.

compress - задава дали да се използва компресия на предаваните данни. Компресията трябва да бъде приложена преди криптирането на данните. Възможни стойност са **yes** и **no**, като **no** е стойност по подразбиране.

disablearrivalcheck - задава дали KLIPS проверява пакетите, идващи по тунела, за достоверен адрес в своя хедър. Валидни стойности са **yes** и **no**.

Параметри на CONN при ръчно указване на ключа.

Описаните параметри са приложими само при ръчно указване на ключа и са игнорирани при автоматичното му указване. Уеднаквяването на стойностите на параметрите в двата края на връзката е задължително. Използването на поне един от протоколите - AH или ESP, също е задължително.

spi - задава spi номера за връзката. Той трябва да е във формата ***0x****hex*, където hex е шестнадесетично число, като за KLIPS е задължително да приема стойност най-малко 0x100, препоръчително е да се използват стойности между 0x100 и 0xfff.

Spibase - този параметър или spi задават spi номера на връзката. Той трябва да е във формата ***0x****hex0*, където hex е шестнадесетично число, като за KLIPS е задължително да приема стойност най-малко 0x100, препоръчително е да се използват стойности между 0x100 и 0xff0.

esp - задава ESP алгоритъм за криптиране и удостоверяване за връзката. Във формат, определен от *ipsec_spi*(8), например **3des-md5-96.** Стойност по подразбиране е да не се използва ESP.

espenckey - задава ESP криптиращ ключ. Той може да бъде зададен отделно за двете страни чрез използване на параметрите **leftespenckey** и **rightespenckey.**

espauthkey - задава ESP удостоверяващ ключ. Може да бъде зададен отделно за двете страни чрез използване на параметрите **leftespauthkey** и **rightespauthkey.**

espreplay_window - задава стойността на ESP replay-window и може да заема стойност от 0 до 64. Приложим е, само ако се използва ESP удостоверяване.

leftespspi - задава SPI, което да се използва за лявата страна на ESP асоциация, като отменя стойността, зададена от spi и spi base.

ah - задава AH удостоверяващ алгоритъм, който да се използва за връзката, например **hmac-md5-96.**

ahkey - задава AH удостоверяващ ключ, ако се използва протоколът AH. Може да бъде зададен поотделно за двете страни чрез **leftahkey** и **rightahkey.**

ahreplay_window - задава AH replay-window като число от 0 (стойност по подразбиране, забраняваща повторни атаки) до 64.

leftahspi - задава SPI, което да се използва за лявата страна на AH-асоциация и отменя стойността, зададена от spi и spi base.

Секция CONFIG.

Единствената секция в частта config е setup, която съдържа информация, използвана при стартиране на Openswan.

```
config setup
interfaces="ipsec0=eth1 ipsec1=ppp0"
klipsdebug=none
plutodebug=all
manualstart=
plutoload="snta sntb sntc sntd"
plutostart=
```

По-долу са изброени параметрите, които са незадължителни, освен ако изрично не е указано обратното:

interfaces - задължителен параметър, който задава физическите интерфейси, използвани от IPSec.

forwardcontrol - определя дали да се включи ip forwarding при стартиране (ако вече не е стартиран) и да го изключи при спиране на IPSec.

syslog - задава името на файла за log съобщенията при стартиране и спиране на IPSec.

klipsdebug - определя до каква степен се отпечатват KLIPS дебъг съобщения. Стойността **none** забранява тези съобщения, а **all** регистрира всички.

plutodebug - задава до каква степен се отпечатват Pluto дебъг съобщения. Стойността **none** забранява тези съобщения, а **all** регистрира всички.

manualstart - задава кои връзки с ръчно указване на ключа да бъдат стартирани в началото от IPSec. Стойността може да бъде име, няколко имена на връзки, разделени със запетая, или да бъде празна.

pluto - задава дали да стартира Pluto или не. Валидни стойности – **yes** и **no**.

plutoload - задава кои връзки (по име) да бъдат заредени в базата данни на Pluto. Стойността може да бъде име, няколко имена на връзки, разделени със запетая, или да бъде празна. Ако се използва стойността **%search,** се зареждат всички връзки с **auto=add**, **auto=route** или **auto=start**.

plutostart - задава кои връзки да се стартират при стартиране на IPSec. Стойността може да бъде име, няколко имена на връзки, разделени със запетая, или да бъде празна. Ако се използва стойността **%search,** се зареждат всички връзки с **auto=add**, **auto=route** или **auto=start**.

plutowait - задава дали Pluto да изчака завършването на договарянето на връзка от Plutostart преди да премине към договаряне на следващата връзка в списъка.

plutobackgroundload - това е параметър, който не се използва. Той определя дали зареждането и стартирането на връзка да бъде като фонов процес или не.

prepluto - команда от shell, която да се изпълни преди стартиране на Pluto.

postpluto - команда от shell, която да се изпълни след стартиране на Pluto.

packetdefault - определя какво трябва да се направи с пакет, който е достигнал KLIPS, но не е намерена необходимата за маршрутизиране информация – eroute. Валидни стойности са **pass**, която разрешава преминаването, **drop** (стойност по подразбиране), която не разрешава преминаването, и **reject** - също като drop, но изпраща icmp съобщение към източника на пакета.

hidetos - определя дали полето TOS на тунелния пакет трябва да бъде зададено на **0** вместо да бъде копирано от пакета на потребителя. Валидни стойности са **yes** (по подразбиране) и **no**.

uniqueids - задава дали специфично ID на участника трябва да бъде запазено уникално, с всяка нова връзка, използваща едно ID от различен IP адрес. Валидните стойности са **yes** и **no,** която е стойност по подразбиране.

overridemtu - определя дали да отхвърли стойността по подразбиране на MTU на IPSec интерфейс.

3.2. Конфигурационен файл ipsec.secrets.

- Удостоверяване с предварително споделена тайна. Форматът на файла се състои от двата IP адреса на участниците във връзката, последвани от предварително споделената тайна в кавички:

```
# sample /etc/ipsec.secrets file for 10.1.0.1
10.1.0.1 10.2.0.1: PSK "secret shared by two hosts"
```

- Удостоверяване чрез RSA публичен ключ. Във файла ipsec.secrets се копира частният ключ на алгоритъма RSA.

```
# an RSA private key.
@my.com: rsa {
    Modulus: 0syXpo/6waam+ZhSs8Lt6jnBzu3C4grtt...
    PublicExponent: 0sAw==
    PrivateExponent: 0shlGbVR1m8Z+7rhzSyenCaBN...
    Prime1: 0s8njV7WTxzVzRz7AP+0OraDxmEAt1BL5l...
    Prime2: 0s1LgR7/oUMo9BvfU8yRFNos1s211KX5K0...
    Exponent1: 0soaXj85ihM5M2inVf/NfHmtLutVz4r...
    Exponent2: 0sjdAL9VFizF+BKU4ohguJFzOd55OG6...
    Coefficient: 0sK1LWwgnNrNFGZsS/2GuMBg9nYVZ...
    }
```

- Удостоверяване чрез сертификати: във файла се съдържа името на частния ключ на сертификата, последвано от защитаваща парола. Възможно е да се използва опцията **%prompt** вместо парола. Тогава системата ще изисква да се въведе защитната парола при стартиране на Pluto.

```
# X.509 certificate
: RSA host.example.com.key "password"
```

4. Реализация на достъпа на астрономическата обсерватория на SUNet през ADSL мрежата на БТК. Проблеми и възможности за включване на други звена.

4.1. Инсталиране на Openswan под Линукс.

Инсталация на OpensWAN заедно с X.509: Съществуват два начина за инсталиране на Openswan - чрез компилиране на кода, разпространяван от Openswan (http://www.openswan.org/), и прилагане на съответните пачове - като поддръжка на NAT-T, или чрез инсталиране на компилирания модул RPM за съответната платформа. От посочения сайт може да се свали RPM за fedora, mandrake, suse и други.

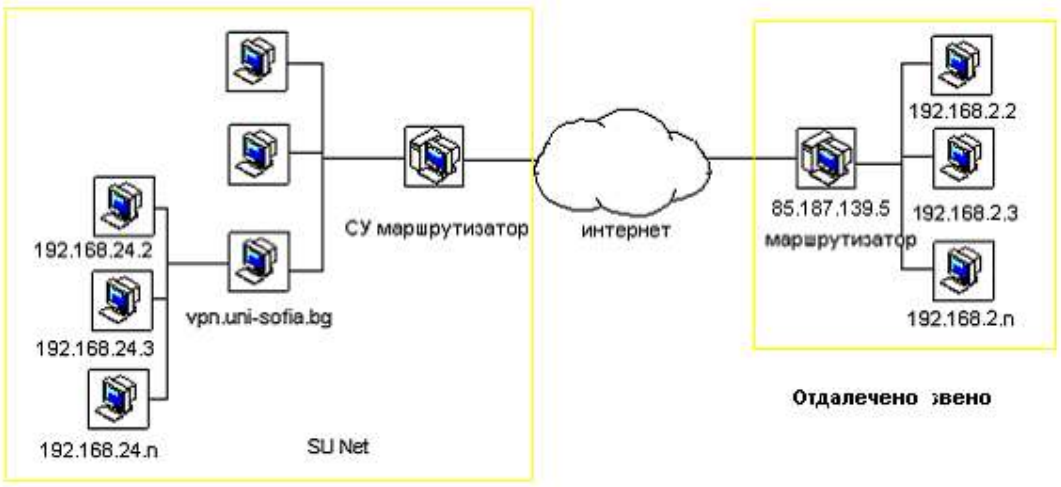
В конкретния случай се използва Линукс Fedora и след сваляне на файла http://www.openswan.org/download/binaries/fedora/3/i386/openswan-2.4.2-1.i386.rpm, се инсталира чрез скрипта **rpm -ivh openswan-2.4.2-1.i386.rpm.**

4.2. Конфигуриране на Openswan за Линукс сървъра vpn.uni-sofia.bg.

След успешното инсталиране на пакетите може да се премине към конфигуриране на сървъра за различните случаи на VPN: „мрежа към мрежа”, „хост към мрежа”, „хост към хост” и „windows хост към мрежа”. За целта ще се използва VPN сървъра vpn.uni-sofia.bg които е с операционната система Линукс Fedora 3.

Както бе разгледано в точка 3, настройките на VPN връзките се задават във файла /etc/ipsec.conf.

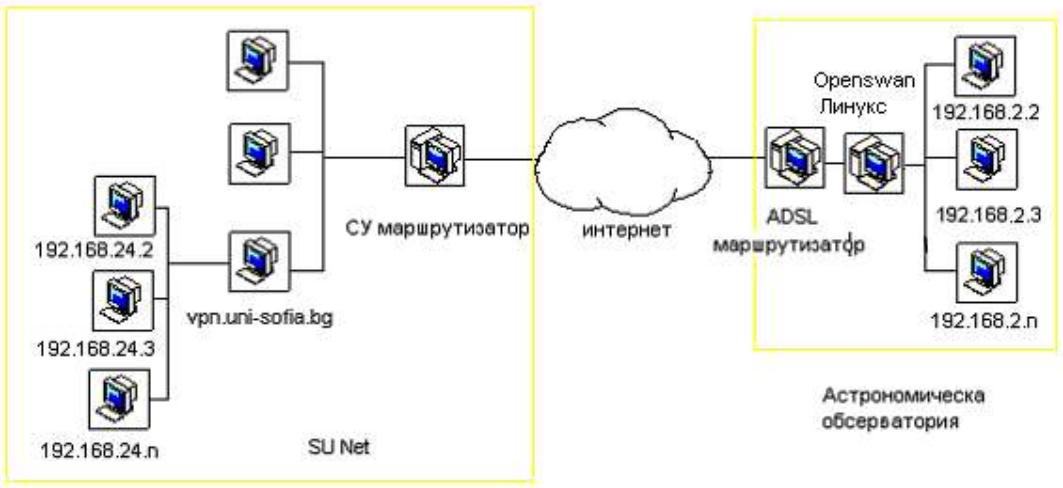
Връзката **home-to-su-icmp** задава конфигурацията Openswan сървър - Openswan клиент и дефинира VPN връзка от типа “мрежа към мрежа” (фиг. 4.1.).



Фигура 4.1.

Чрез нея се дефинира VPN между подмрежата 192.168.24.0/24 , която се намира зад vpn.uni-sofia.bg и подмрежата 192.168.2.0/24 зад 85.187.139.5, служещ за маршрутизатор.

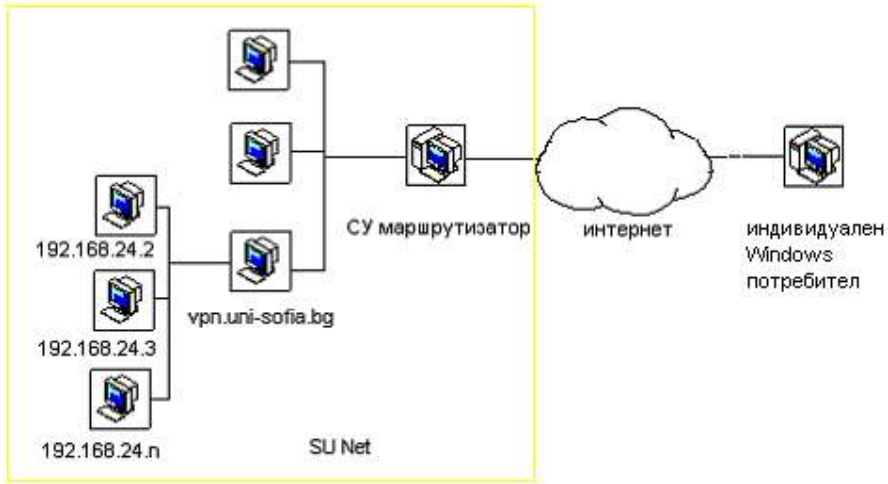
Връзката **home-to-su-icmp-nat** задава конфигурацията Openswan сървър - Openswan клиент зад NAT и дефинира VPN връзка от типа “мрежа към мрежа” – фиг. 4.2.



Фигура 4.2.

Чрез нея се дефинира VPN между подмрежата 192.168.24.0/24, която се намира зад vpn.uni-sofia.bg, и подмрежата 192.168.2.0/24 зад ADSL маршрутизатор с адрес 192.168.1.1.

В секцията **win-to-su-icmp** е дадена конфигурацията на Openswan сървър машина за клиенти с Windows 2000/XP - конфигурацията на VPN е показана на фиг. 4.3. Разглежданият случай покрива варианта на отдалечен достъп на клиенти с операционната система Windows и може да се използва, както за свързване на компютри от Астрономическата обсерватория и Центъра "Иван Дуйчев", така и на други клиенти с Windows към VPN сървъра на СУ.



Фигура 4.3.

Съдържанието на файла ipsec.conf за трите връзки conn е следното:

```
conn home-to-su-icmp
      authby=rsasig
      pfs=no
      auto=add
      rekey=no
      left=62.44.96.35
      leftnexthop=62.44.96.3
      leftsubnet=192.168.24.0/24
      leftrsasigkey=%cert
      leftcert=/etc/ipsec.d/certs/clientCert.pem
      leftprotoport=icmp
      #
      # The remote user.
      #
      right=85.187.139.5
      rightnexthop=85.187.139.5
      rightcert=/etc/ipsec.d/certs/hostCert.pem
      rightrsasigkey=%cert
      rightprotoport=icmp
      rightsubnet=192.168.2.0/24

conn home-to-su-icmp-nat
      authby=rsasig
      pfs=no
      auto=add
      rekey=no
      left=62.44.96.35
      leftnexthop=62.44.96.3
      leftsubnet=192.168.24.0/24
      leftrsasigkey=%cert
      leftcert=/etc/ipsec.d/certs/clientCert.pem
      leftprotoport=icmp
      #
      # The remote user.
      #
      right=%any
      rightcert=/etc/ipsec.d/certs/hostCert.pem
      rightrsasigkey=%cert
      rightprotoport=icmp
      rightsubnet=192.168.2.0/24

conn win-to-su-icmp
      authby=rsasig
      rekey=no
      left=62.44.96.35
      leftnexthop=62.44.96.3
      leftrsasigkey=%cert
      leftcert=/etc/ipsec.d/certs/clientCert.pem
      # For updated Windows 2000/XP clients,
      # to support old clients as well, use leftprotoport=17/%any
      leftprotoport=17/0
      #
      # The remote user.
      right=%any
      rightca=%same
      rightrsasigkey=%cert
      rightprotoport=17/1701
      rightsubnet=vhost:%priv,%no
      pfs=no
      auto=add
```

Необходимите сертификати и CA използвани в дипломната работа са генерирани с OpenSSL. Файловете със сертификатите `hostCert.pem clientCert.pem winhostCert.pem` се копират в папката /etc/ipsec.d/certs/, частните ключове hostKey.pem, clientKey.pem и winhostKey.pem се копират в папката /etc/ipsec.d/private/, а в /etc/ipsec.d/cacerts/ се копира файлът, съдържащ CA - caCert.pem.

Сертификатът `clientCert.pem` се използва за удостоверяване на VPN сървъра. `hostCert.pem` се използва за удостоверяване на отдалечените VPN клиенти с операционна система Линукс, а winhostCert.pem - за клиенти с операционната система Windows. За да се инсталира сертификатът winhostCert.pem под Windows, е необходимо генерирането с OpenSSL на winhostCert.p12 файл, които съдържа CA, сертификата и частния ключ.

След всяка промяна в конфигурационните файлове ipsec.conf и ipsec.secrets е необходимо Openswan да се рестартира чрез **service ipsec restart.**

Със следната команда **ipsec verify** се проверява дали ipsec е инсталиран и конфигуриран правилно. Резултатът при успешно конфигуриране изглежда по следния начин:

```
>ipsec verify
Checking your system to see if IPsec got installed and started correctly:
Version check and ipsec on-path                                 [OK]
Linux Openswan U2.4.4/K2.6.12-1.1381_FC3 (netkey)
Checking for IPsec support in kernel                            [OK]
Checking for RSA private key (/etc/ipsec.secrets)           [FAILED]
ipsec showhostkey: no default key in "/etc/ipsec.secrets"
Checking that pluto is running                                  [OK]
Two or more interfaces found, checking IP forwarding            [OK]
Checking NAT and MASQUERADEing                                  [OK]
Checking for 'ip' command                                       [OK]
Checking for 'iptables' command                                 [OK]
Checking for 'setkey' command for NETKEY IPsec stack support    [OK]
Opportunistic Encryption Support                          [DISABLED]
```

За правилната работа на IPSec,ако се използва защитна стена трябва да се отворят UDP порт 500 (IKE), протокол 50 (ESP), протокол 51(AH) и UDP порт 4500 при използване на NAT-T.

4.3. Конфигуриране на L2TP и PPP на сървъра vpn.uni-sofia.bg.

1) За поддръжката на Windows клиенти е необходимо да се инсталира и l2tp сървър под Линукс. Това се осъществява чрез изтегляне на файла

http://www.openswan.org/download/binaries/fedora/3/i386/l2tpd-0.69-13.i386.rpm и инсталирането му чрез скрипта rpm –ivh l2tpd-0.69-13.i386.rpm.

2) Конфигуриране на l2tpd. При Fedora е необходимо да се редактира файлът '/etc/l2tpd/l2tpd.conf'. По-долу е посочен пример :

```
[global]
auth file = /etc/l2tpd/l2tp-secrets
[lns default]
ip range = 172.22.127.2-172.22.127.250
local ip = 172.22.127.1
require chap = yes
refuse pap = yes
require authentication = yes
name = MyVPN
ppp debug = yes
pppoptfile = /etc/ppp/options.l2tpd.lns
length bit = yes
```

За адресно пространство на клиентите се заделя област от неизползвани във вътрешната мрежа IP адреси. Local IP задава адреса на VPN сървъра.

3) Конфигуриране на опциите PPP. В примера по-горе те са разположени в /etc/ppp/options.l2tpd.lns.

```
ipcp-accept-local
ipcp-accept-remote
ms-dns 172.22.127.1
ms-wins 172.22.127.1
auth
crtscts
idle 1800
mtu 1200
mru 1200
nodefaultroute
debug
lock
proxyarp
connect-delay 5000
nologfd
```

ms-dns и ms-wins трябва да съвпадат с DNS и WINS сървърите във вътрешната мрежа. mtu е със стойност 1200, тъй като, ако бъде оставен със стойността си по подразбиране 1500, може да възпрепятства правилната работа на някои протоколи като SMB например.

4) Задаване на удостоверяващия файл - /etc/ppp/chap-secrets.

```
# Secrets for authentication using CHAP
# client        server secret       IP addresses
  username      * password *
```

В този файл се задават потребителското име, паролата и съответно IP адрес, ако желаем да зададем статичен IP адрес.

След настройване на всички конфигурационни файлове се стартира l2tpd с командата '/etc/init.d/l2tpd start'.

4.4 Конфигуриране на Openswan на отдалечени Линукс клиенти.

Тук ще бъдат разгледани двата случая за отдалечените клиенти с операционна система Линукс - клиент с публичен адрес и клиент зад NAT. Инсталирането и настройването на Openswan на клиентския Линукс се осъществява по същия начин, както при сървъра.

При случая с публичен адрес връзката **home-to-su-icmp** е същата, като при сървъра:

```
conn home-to-su-icmp
      authby=rsasig
      pfs=no
      auto=add
      rekey=no
      left=62.44.96.35
      leftnexthop=62.44.96.3
      leftsubnet=192.168.24.0/24
      leftrsasigkey=%cert
      leftcert=/etc/ipsec.d/certs/clientCert.pem
      leftprotoport=icmp
      right=85.139.187.5
      rightnexthop=85.139.187.1
      rightsubnet=192.168.2.0/24
      rightcert=/etc/ipsec.d/certs/hostCert.pem
      rightrsasigkey=%cert
      rightprotoport=icmp
```

В случая, при който Линукс сървъра е зад NAT, връзката е следната:

```
conn home-to-su-icmp-nat
      authby=rsasig
      pfs=no
      auto=add
      rekey=no
      left=62.44.96.35
      leftnexthop=62.44.96.3
      leftsubnet=192.168.24.0/24
      leftrsasigkey=%cert
      leftcert=/etc/ipsec.d/certs/clientCert.pem
      leftprotoport=icmp
      right=192.168.1.2
      rightnexthop=192.168.1.1          #ADSL маршрутизатор
      rightsubnet=192.168.2.0/24
      rightcert=/etc/ipsec.d/certs/hostCert.pem
      rightrsasigkey=%cert
      rightprotoport=icmp
```

В този случай връзката може да се инициира само от клиентската машина (right страната) , тъй като VPN сървърът не знае IP адреса на клиента.

Подробно ще бъде разгледано установяването на връзката **home-to-su-icmp.**

Стартирането на връзката между двете страни се осъществява чрез скрипта **ipsec auto --up home-to-su-icmp.** Резултатът при успешно установяване на връзката е посочен по-долу:

```
ipsec auto --up home-to-su-icmp
104 "home-to-su-icmp" #1: STATE_MAIN_I1: initiate
003 "home-to-su-icmp" #1: received Vendor ID payload [Openswan (this version)
2.4.4  X.509-1.5.4 PLUTO_SENDS_VENDORID PLUTO_USES_KEYRR]
003 "home-to-su-icmp" #1: received Vendor ID payload [Dead Peer Detection]
003 "home-to-su-icmp" #1: received Vendor ID payload [RFC 3947] method set
to=109
106 "home-to-su-icmp" #1: STATE_MAIN_I2: sent MI2, expecting MR2
003 "home-to-su-icmp" #1: NAT-Traversal: Result using 3: no NAT detected
108 "home-to-su-icmp" #1: STATE_MAIN_I3: sent MI3, expecting MR3
004 "home-to-su-icmp" #1: STATE_MAIN_I4: ISAKMP SA established
{auth=OAKLEY_RSA_SIG cipher=oakley_3des_cbc_192 prf=oakley_md5 group=modp1536}
117 "home-to-su-icmp" #2: STATE_QUICK_I1: initiate
004 "home-to-su-icmp" #2: STATE_QUICK_I2: sent QI2, IPsec SA established
{ESP=>0x6f78f193 <0xd110f47f xfrm=AES_0-HMAC_SHA1 IPCOMP=>0x0000949d <0x0000dde7
NATD=62.44.96.35:500 DPD=none}
```

След успешно установяване на връзката комуникацията между двете мрежи по криптиран тунел в зависимост от протокола и портовете, разрешени в leftprotoport= и rightprotoport= е възможна. В случая за експеримента е разрешен протоколът ICMP и командата ping може да се изпълнява между компютрите от двете мрежи 192.168.2.0/24 и 192.168.24.0/24.

Проверяване на състоянието на връзката.

```
ipsec auto status

000 "home-to-su-icmp": 192.168.24.0/24===62.44.96.35[C=GB, ST=Berkshire, O=My
Company Ltd, CN=pc9, E=ceco1@yahoo.com]:1/0---62.44.96.3...%any[C=GB,
ST=Berkshire, O=My Company Ltd, CN=pc2, E=ceco@yahoo.com]:1/0===192.168.2.0/24;
unrouted; eroute owner: #0
000 "home-to-su-icmp":     srcip=unset; dstip=unset; srcup=ipsec _updown;
dstup=ipsec _updown;
000 "home-to-su-icmp":   CAs: 'C=GB, ST=Berkshire, L=Newbury, O=My Company Ltd,
CN=pc2, E=tsvetomir_h@yahoo.com'...'C=GB, ST=Berkshire, L=Newbury, O=My Company
Ltd, CN=pc2, E=tsvetomir_h@yahoo.com'
000 "home-to-su-icmp":   ike_life: 3600s; ipsec_life: 28800s; rekey_margin:
540s; rekey_fuzz: 100%; keyingtries: 0
000 "home-to-su-icmp":   policy: RSASIG+ENCRYPT+TUNNEL+DONTREKEY; prio: 24,24;
interface: eth0;
000 "home-to-su-icmp":   newest ISAKMP SA: #0; newest IPsec SA: #0;
000 "home-to-su-icmp"[1]: 192.168.24.0/24===62.44.96.35[C=GB, ST=Berkshire, O=My
Company Ltd, CN=pc9, E=ceco1@yahoo.com]:1/0---62.44.96.3...85.187.139.5[C=GB,
ST=Berkshire, O=My Company Ltd, CN=pc2, E=ceco@yahoo.com]:1/0===192.168.2.0/24;
erouted; eroute owner: #2
```

```
000 "home-to-su-icmp"[1]:     srcip=unset; dstip=unset; srcup=ipsec _updown;
dstup=ipsec _updown;
000 "home-to-su-icmp"[1]:   CAs: 'C=GB, ST=Berkshire, L=Newbury, O=My Company
Ltd, CN=pc2, E=tsvetomir_h@yahoo.com'...'C=GB, ST=Berkshire, L=Newbury, O=My
Company Ltd, CN=pc2, E=tsvetomir_h@yahoo.com'
000 "home-to-su-icmp"[1]:   ike_life: 3600s; ipsec_life: 28800s; rekey_margin:
540s; rekey_fuzz: 100%; keyingtries: 0
000 "home-to-su-icmp"[1]:   policy: RSASIG+ENCRYPT+TUNNEL+DONTREKEY; prio:
24,24; interface: eth0;
000 "home-to-su-icmp"[1]:   newest ISAKMP SA: #1; newest IPsec SA: #2;
000 "home-to-su-icmp"[1]:   IKE algorithm newest: 3DES_CBC_192-MD5-MODP1536
000 #2: "home-to-su-icmp"[1] 85.187.139.5:500 STATE_QUICK_R2 (IPsec SA
established); EVENT_SA_EXPIRE in 28485s; newest IPSEC; eroute owner
000 #2: "home-to-su-icmp"[1] 85.187.139.5 esp.a5509e0d@85.187.139.5
esp.38f9e656@62.44.96.35 comp.17fb@85.187.139.5 comp.acf6@62.44.96.35
tun.0@85.187.139.5 tun.0@62.44.96.35
000 #1: "home-to-su-icmp"[1] 85.187.139.5:500 STATE_MAIN_R3 (sent MR3, ISAKMP SA
established); EVENT_SA_EXPIRE in 3278s; newest ISAKMP; lastdpd=-1s(seq in:0
out:0)
```

Проблеми и дебъгване.

Има малко на брой опции за дебъгване за Openswan. Пълна debug информация може да се получи чрез задаване на стойноста **all** на **klipsdebug** и **plutodebug.** По този начин може да се получи информация за процеса на осъществяване на връзките, както и за възникнали проблеми.

Команда ipsec eroute позволява да се конфигурират виртуални маршрути през VPN тунел. Ако трябва да се добавят мрежови маршрути след като VPN тунелът е установен, може да се наложи маршрутът да се настрои с тази команда.

4.5. Конфигуриране на Windows XP клиент за свързване към Openswan по протокола l2tp/ipsec.

Първо е необходимо да се инсталира Windows sp2, за да е сигурно, че е инсталирана поддръжка на NAT-Traversal, което ще позволи да се свързваме зад NAT шлюз. След това се инсталира сертификат, тъй като това е избраният метод за удостоверяване.

Инсталиране на цифров сертификат под Windows (Windows 2000/XP).

В повечето случаи сертификатите са във файлове във формат PKCS#12. Тези файлове съдържат сертификата на потребителя, съответния частен ключ и един или повече CA базови (root) сертификати. Съществуват два начина за инсталиране на сертификати: ръчен и автоматичен. Автоматичното инсталиране на сетификати се осъществява чрез допълнителни програми, които е необходимо да се свалят и

инсталират. Например certimport от Xelerance или pfxMachineImport от Keith Brown импортират PKCS#12 файл без намесата на потребителя.

Ръчното инсталиране на сертификати под Windows 2000 или XP се осъществява по следния начин:

1. Влизане с потребителски акаунт “Administrator” (инсталирането на сертификат за IPSec изисква администраторски правомощия).
2. Ако се използва Windows 2000 Professional, трябва да се инсталира “High Encryption Pack” или SP2+.
3. Стартира се Microsoft Management Console от “Start” и след това “Run” и въвеждане на “mmc”.
4. От менюто се избира “File” и след това “Add/Remove Snap-in”.
5. Натиска се бутонът “Add”.
6. От списъка се избира “Certificates”. Натиска се бутонът “Add”, след което бутонът “OK”.
7. Избира се “Computer account”. Това е необходимо, тъй като Майкрософт предполага, че IPSec сертификатите могат да идентифицират само компютри, а не хора.
8. Избира се “Local computer”.
9. За да се добави snap-in модулът “Certificates”, се натиска “OK”.
10. Разширяване на дървото на “Certificates (Local Computer)”. Най-отгоре се намира папка “Personal”. В нея се намира папка “Certificates”. Тя се избира с левия бутон на мишката, след което се отваря контекстното й меню с десния бутон на мишката. Избира се “All task”" и след това “Import”.
11. По този начин се стартира “Certificate Import Wizard”. Натиска се бутонът “Next”.
12. Появява се диалоговият прозорец “File to import”. Първоначално полето “File name” е празно. Натиска се “Browse”.
13. Променя се “Files of type” на “Personal Information Exchange (*.pfx, *.p12)”.
14. Избира се файлът със сертификата за този потребител и се натиска “Open”.
15. По този начин се осъществява връщане в диалоговия прозорец “File to import”. Натиска се “Next”.
16. Въвежда се парола за защита на файла със сертификата.

17. Избира се “Automatically select the certificate store”. (Не трябва да се използва “Place all certificates in the following store: Personal”). Натиска се “Next”.

18. Завършва се “Certificate Import Wizard” чрез натискане на бутона “Finish”.

19. Първоначално инсталираният сертификат няма да се вижда в прозореца. От менюто се избира “Action” и след това “Refresh”. Инсталираният сетрификат ще се появи.

20. От менюто се избира “File” и след това “Exit”.

След като сертификатът на потребителя бъде инсталиран, може да се конфигурира L2TP/IPSec връзка, която използва този сертификат за удостоверяване пред Linux Openswan сървър. Няма възможност да се избере кой сертификат да се използва за конкретната L2TP/IPSec връзка.

За създаване (настройване) на L2TP върху IPSec връзка са необходими следните стъпки:

1. Избира се “Network Connection” от Start – Control Panel.
2. От “File” се избира “New Connection”.
3. Селектира се “Connect to the Network at my Workplace”, след което се натиска “Next”.
4. Съветникът “New Connection Wizard” визуализира страницата “Network Connection”, която съдържа следните две опции: “Dial-Up Connection” и “Virtual Private Network Connection”. Избира се втората опция и се натиска “Next”.
5. След това се въвежда името на връзката в полето “Company Name”, а в полето “VPN Server Selection” – името на хоста или IP адреса на сървъра, към който се свързваме.
6. След това се завършва връзката и се избират “Properties”.
7. Зарежда се “Networking Tab”.
8. Променя се типът на VPN връзката от “Auto” на “L2TP IPSec VPN”.
9. Настройките се запазват.
10. Въвежда се потребителско име и парола.
11. Натиска се бутонът “Connect”, с което VPN връзката е осъществена.

След свързване Windows клиента получава частен IP адрес 172.22.127.11 според зададената конфигурация.

Важно е да се отбележи, че в опциите на TCP/IP протокола на създадената връзка, настройката “Use default gateway on remote network” трябва да не е разрешена. Ако това не е направено, целият интернет трафик ще се пренасочи през отдалечената мрежа. Необходимо е чрез командата route да се пренасочват маршрутите за достъп до отдалечената мрежа. Например маршрутът до подмрежата 192.168.24.0 се задава чрез командата - **Route add 192.168.24.0 mask 255.255.255.0 172.22.127.11.** По този начин трафикът към Интернет ще минава през default gateway интерфейса, а този, предназначен за отдалечената мрежа - през интерфейса на виртуалната частна мрежа 172.22.127.11**.**

Заключение

В настоящата дипломна работа е направен преглед на най-разпространените технологии за VPN. Целта на дипломната работа беше да се даде възможност за отдалечен достъп за различни клиенти до мрежата на СУ SUNet чрез VPN.За реализирането беше избран протоколът IPSec, който отговаря най-пълно на изискванията към VPN. Конкретния продукт за линукс който бе използван е Openswan, който е изцяло съвместим със стандартите на IPsec, а също поддържащ NAT-T, необходим ни заради използването на клиенти зад NAT. Разгледани са и са осъществени вариантите за VPN достъпът до мрежата на SU –Sunet за мрежи и индивидуални клиенти с операционната система Линукс и Windows. Въпреки че конкретно свързване на Астрономическата обсерватория към VPN сървъра на SU не бе осъществено, поради проблеми с достъпа до там, реализираните връзки покриват и случая с обсерваторията, така че целите поставени в дипломната работа са изпълнени. Възможно е разширяването конфигурациите на VPN сървъра, така че да поддържа и други клиенти - например различни хардуерни маршрутизатори с IPSec VPN. Изводът които може да се направи е, че разгледаното решение може успешно да намери приложение в реализирането на VPN свързаност при различни звена в СУ, както и при други организации и корпоративни мрежи, осигурявайки сигурна комуникация между мрежите им.

Използвана литература:

1. **книга:** *James S Tiller :VPN - A Technical Guide to IPSec Virtual Private Networks*
2. **книга:** *J.Davies and E. Lewis: Deploying VPN with Microsoft Windows Server 2003*
3. *Документация на Openswan http://www.openswan.org/docs/*
4. *Документация на Openvpn http://openvpn.net/*