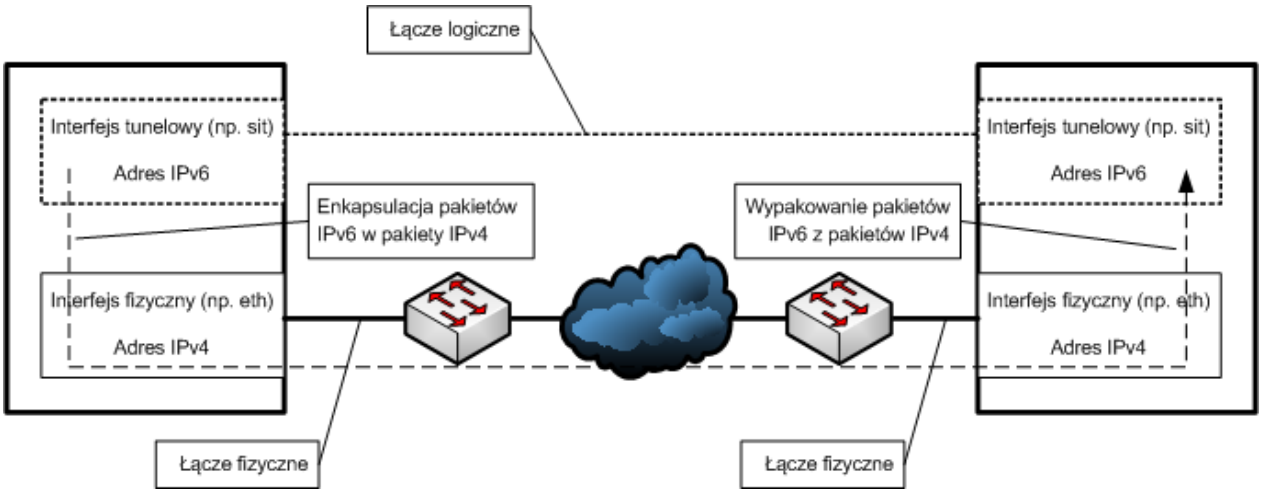
1. Tunele

Tunele sieciowe porównać można do wirtualnych łączy punkt-punkt (najczęściej), wykorzystujących w celu przenoszenia ruchu proces enkapsulacji.
Oznacza to, że jednostki danych określonego protokołu, przenoszone są z użyciem (w polu danych) jednostek należących do protokołu tej samej lub wyższej warstwy ISO-OSI. W większości przypadków protokół przenoszony i przenoszący są różne (IPv6 w IPv4), lecz nie jest to regułą (tunele IPv4 w IPv4).
Rozwiązanie tego rodzaju skutkuje w praktyce ukryciem ruchu protokołu przenoszonego, przed elementami sieci – przetwarzają one wyłącznie informacje dotyczące protokołu przenoszącego, traktując dane protokołu przenoszonego, jako bliżej nieokreślone dane użytkownika.
Zastosowania tuneli obejmują np.:

- Przenoszenie ruchu protokołów przez nieobsługiwanych przez daną sieć tranzytową.
- Łączenie odległych elementów systemu sieciowego.
- Zabezpieczenie transmitowanych danych.

Należy zwrócić uwagę, różne rodzaje tuneli wykorzystują nie tylko różne protokoły przenoszące, lecz również często nakładają ograniczenia na dopuszczalne protokoły przenoszone. Na przykład tunele SIT (patrz dalej) posługują się protokołem IPv4 jako protokołem przenoszącym, a jedynym dozwolonym protokołem przenoszonym jest IPv6. Z kolei tunele GRE mogą być tworzone z użyciem zarówno IPv4 jak i IPv6 jako protokołu przenoszącego, a liczba możliwych protokołów przenoszonych jest jeszcze liczniejsza i zawiera także protokoły warstwy 2 ISO-OSI.

Podczas laboratorium interesują nas tzw. tunele statyczne. W ich przypadku punkty końcowe tunelu są stałe i jednoznacznie określone poprzez podanie adresów właściwych dla protokołu przenoszącego.
Utworzenie tunelu wymaga, w tym przypadku określenia konkretnego rodzaju tunelu, podania adresów jego końców oraz ewentualnych opcji dodatkowych (np. dotyczących kompresji czy zabezpieczeń).
Tunele tego typu są najczęściej widoczne w systemie operacyjnym, jako nowe (wirtualne) interfejsy sieciowe. Interfejs taki można porównać do nowej karty sieciowej zainstalowanej w urządzeniu, połączonej bezpośrednio z analogiczną kartą sieciową w urządzeniu na drugim końcu tunelu.



Można powiedzieć, że interfejsy tunelowe są połączone ze sobą bezpośrednio łączem logicznym, które wykorzystuje do swojego działania protokół przenoszący.

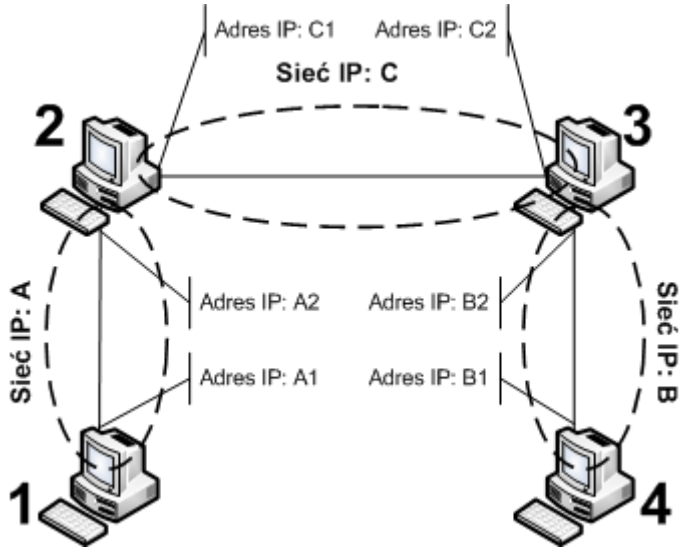
Należy pamiętać, iż utworzenie tego rodzaju tunelu powoduje pojawienie się nowego interfejsu, który możemy wykorzystać zgodnie z ogólnie przyjętymi zasadami, np. można przypisać mu adres, ustawić routing przez ten interfejs, uruchomić serwer wykorzystujący adres przypisany do tego interfejsu, itp. Samo utworzenie tunelu nie powoduje automatycznego skierowania przez niego jakiegokolwiek ruchu.

Do tego rodzaju tuneli obsługiwanych w systemie Linux należą np.:

- ipip – przenosi ruch IPv4 w pakietach IPv4,
- ip6ip6 – przenosi ruch IPv6 w pakietach IPv6,
- Simple IP Transition (SIT) – przenosi ruch IPv6 w pakietach IPv4,
- ipip6 – przenosi ruch IPv4 w pakietach IPv6,
- GRE – jest w stanie przenosić ruch różnych protokołów z użyciem pakietów IPv4 lub IPv6.

2. Przykładowy scenariusz routingu IP

W celu przypomnienia podstaw routingu IP prosimy o zapoznanie się z dokumentem „Routing statyczny IP - podstawowe mechanizmy.pdf".
Poniżej przedstawiono analizę routingu w przykładowej sieci złożonej z 4 urządzeń i 3 sieci IP.



Każde z widocznych urządzeń (1,2,3,4) posiada automatycznie dopisane trasy (lokalne), do sieci IP do których samo należy:

- 1: do A,
- 2: do A i C,
- 3: do C i B,
- 4: do B.

Trasy te są trasami lokalnymi, tzn. sprowadzają się do wysłania danych na określony interfejs.
Oznacza to, że 1 nie wysłać danych do 3 i 4 (nie zna trasy do żadnego z ich adresów). Nie może też wysyłać danych do adresu urządzenia 2, należącego do sieci C (bo nie zna do niej trasy).
Podobnie 4 nie może wysyłać do 2 i 1 oraz do adresu urządzenia 3 należącego do sieci C.

Urządzenie 2 może wysyłać do 1 i na adres 3 należący do sieci C, gdyż samo należy do sieci A i C, więc posiada do nich automatycznie dopisane trasy. Nie zna trasy do adresu urządzenia 4 oraz do adresu urządzenia 3 w sieci B.
Urządzenie 3 może wysyłać do 4 i na adres 2 należący do sieci C, gdyż samo należy do sieci B i C, więc posiada do nich automatycznie dopisane trasy. Nie zna trasy do adresu urządzenia 1 oraz do adresu urządzenia 2 w sieci A.

W celu zapewnienia pełnej łączności w powyższym środowisku, musimy dodać następujące trasy:

- w 1: do sieci B i C
- w 2: do sieci B,
- w 3: do sieci A,
- w 4: do sieci A i C.

Wszystkie z tych tras będą trasami dostarczania zdalnego, tzn. nie każemy tu, po prostu wysyłać ruchu przez określony interfejs (dostarczanie lokalne), lecz wysyłać go pod określony adres IP. W przeciwnym przypadku żadne z urządzeń nie odbierze takiego ruchu – ostatecznego odbiorcy nie ma na łączu, a router nie będzie wiedział, że powinien odebrać i przekazać dalej taki ruch.
Wyraźne rozróżnienie pomiędzy dostarczaniem lokalnym i zdalnym występuje, gdy ruch ma opuścić nasze urządzenie przez łącze zrealizowane w technice wielodostępowej, tzn. takiej, do która pozwala łączyć jednocześnie więcej niż 2 urządzenia (np. Ethernet). W takim właściwy wybór – dostarczanie zdalne czy lokalne – jest kluczowy dla działania systemu.
Jeśli natomiast interfejs którym ruch będzie wysyłany jest typu punkt-punkt (np. większość tuneli), to można zawsze stosować dostarczanie lokalne – nawet jeśli ruch nie jest adresowany do urządzenia bezpośrednio podłączonego do tego interfejsu, tylko ma być przez nie przekazany dalej. Jest to możliwe, gdyż w przypadku połączenia punkt-punkt mamy do czynienia tylko i wyłącznie z pojedynczym możliwym odbiorcą – urządzeniem po drugiej stronie łącza. Może ono zatem założyć, że wszystko co wysyłamy powinno zostać przez nie odebrane – po odebraniu urządzenie stwierdzi , czy dane są przeznaczone dla niego (wtedy przekaże je warstwom wyższym) czy też powinno przekazać je dalej..

Jeśli zakładamy, że wszystkie połączenia na schemacie zrealizowane będą z użyciem wielodostępowych technik transmisji, wymagane będą następujące trasy:

- w 1:
    - do sieci B: wysyłaj pod adres A2,
    - do sieci C: wysyłaj pod adres A2.
- w 2:
    - do sieci B: wysyłaj pod adres C2.
- w 3:
    - do sieci A: wysyłaj pod adres C1.
- w 4:
    - do sieci A: wysyłaj pod adres B2,
    - do sieci C: wysyłaj pod adres B2.

Należy też pamiętać o włączeniu przekazywania pakietów IP (forwarding'u) na urządzeniach 2 i 3.

3. IPSec

IPSec jest zestawem protokołów, który pozwala na zestawienie bezpiecznego tunelu VPN pomiędzy sieciami za pośrednictwem Internetu.
Oferuje następujące usługi:

- Uwierzytelnienie
- Integralność danych
- Kontrola dostępu
- Poufność

W architekturze IPSec, w dokumencie RFC2401 wyróżniono następujące elementy funkcjonalne:

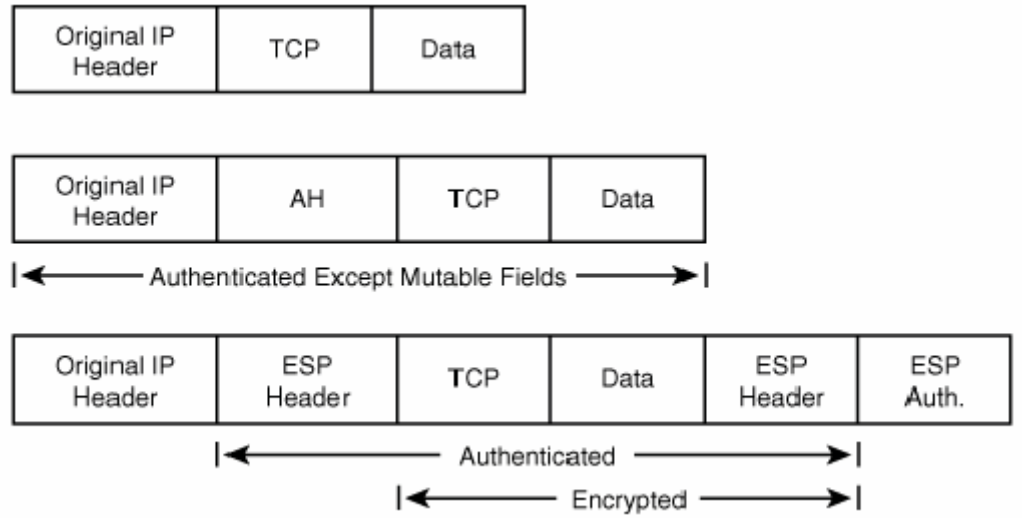
- **Protokoły bezpieczeństwa** *Authentication Header (*AH*)* oraz *Encapsulation Security Payload (*ESP*)*
- **Zarządzanie kluczami** ISAKMP, IKE, SKEME
- **Algorytmy** szyfrowania i uwierzytelnienia

IPSec oferuje dwa tryby pracy, które świadczą różne usługi:

- Transportowy (*ang. Transport Mode*)
- Tunelowy (*ang. Tunnel Mode*)

3.1. Tryby pracy IPSec

3.1.1. Transport Mode

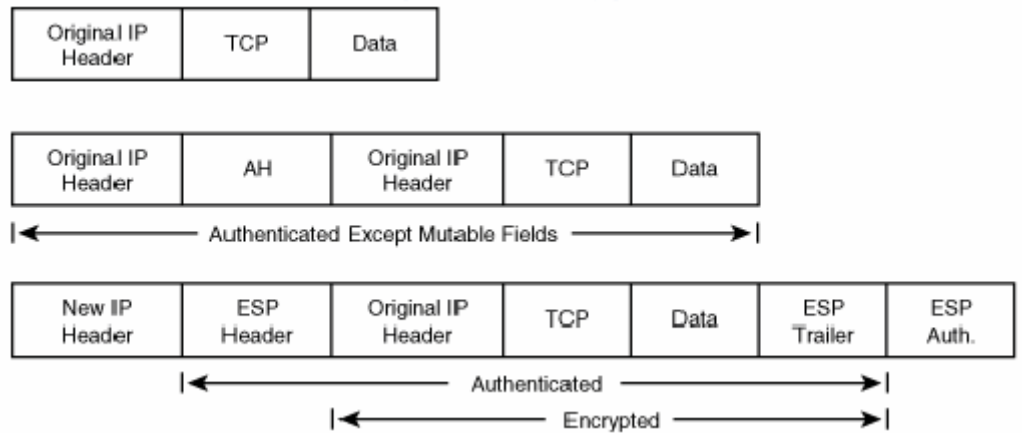


Tryb Transportowy charakteryzuje się tym, że struktura nagłówka IP pozostaje niezmieniona. Ponieważ zachowany jest oryginalny nagłówek (a więc i adresy IP), adresy IP komunikujących się stron muszą być routowalne na całej trasie chronionej transmisji. Jest to niewątpliwie ograniczenie tego trybu. Kolejnym ograniczeniem wynikającym z poprzedniego jest brak współpracy tego trybu z translacją adresów (NAT).

Za niezmienionym nagłówkiem IP wstawiane są pola protokołów ESP bądź AH. Wybierając AH zapewniona będzie tylko ochrona integralności ruchu, natomiast ESP zapewnia ochronę integralności i poufności według powyższej ilustracji.

Tryb transportowy stosuje się w przypadku, gdy chcemy zabezpieczyć połączenie pomiędzy dwoma elementami infrastruktury sieci o określonych adresach.

3.1.2. Tunnel Mode



W tym trybie oryginalny nagłówek IP wraz innymi nagłówkami oraz danymi jest umieszczany w polu danych nowego pakietu IP. Następnie dodawane są nagłówki IPSec (AH lub ESP). Obszary pakietu objęte ochroną integralności i poufności przedstawia rysunek powyżej.

Z racji opisanej powyżej enkapsulacji nie występuje ograniczenie obecne w trybie transportowym, co do używania adresów routowalnych na całej ścieżce pomiędzy stronami. Efekt jest tożsamy z zestawieniem pomiędzy urządzeniami tunelu IpIp i objęciem go ochroną z użyciem mechanizmów IPSec. Dzięki temu umożliwia on zestawienie bezpiecznego tunelu pomiędzy użytkownikami końcowymi z prywatnymi adresami IP (np. dwoma oddziałami firmy) poprzez sieć publiczną.

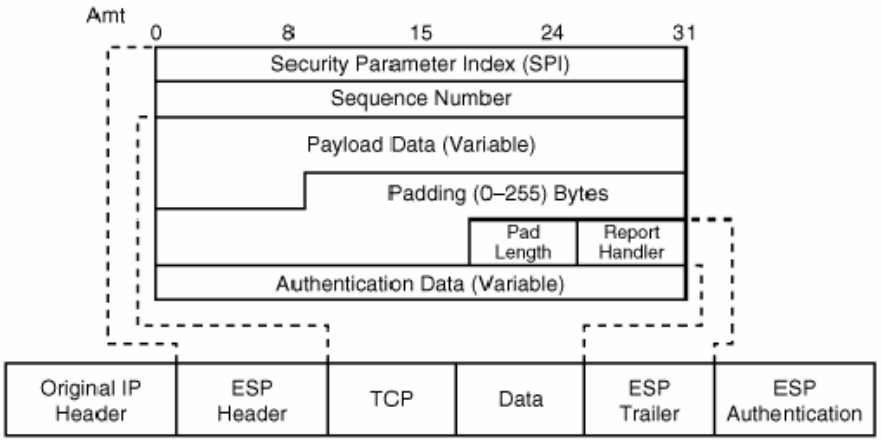
3.2. Protokoły bezpieczeństwa w IPSec

3.2.1. Encapsulating Security Payload (ESP)

ESP zapewnia:

- Poufność
- Integralność danych
- Uwierzytelnienie pochodzenia danych (opcjonalne)
- Zabezpieczenie przed atakami powtórzeniowymi (opcjonalne)

ESP jest protokołem IP identyfikowanym numerem 50 w nagłówku IP.



- **Security Parameter Index (SPI)** – 32bitowa wartość, która w połączeniu z docelowym adresem IP identyfikuje *Security Association (SA)*, które ma być użyte do przetworzenia pakietu

- **Sequence Number** – jest liczbą zwiększaną sekwencyjnie przez stronę nadającą. Wraz z mechanizmem okna przesuwnego zapobiega atakom powtórzeniowym
- **Payload Data** – dane podlegające szyfrowaniu przez ESP
- **Padding** – dodatkowe bity dodawane do nagłówka ESP. Długość zależna od wykorzystywanego algorytmu szyfrującego
- **Pad Lenght** – długość pola Padding
- **Report Handler** – określa typ danych w Payload. Dla ESP w trybie Tunnel wartość będzie równa 4
- **Authentication Data** – wartość kontrolna pozwalająca sprawdzić integralność danych. Sprawdzenie integralności odbywa się przed odszyfrowaniem wiadomości.

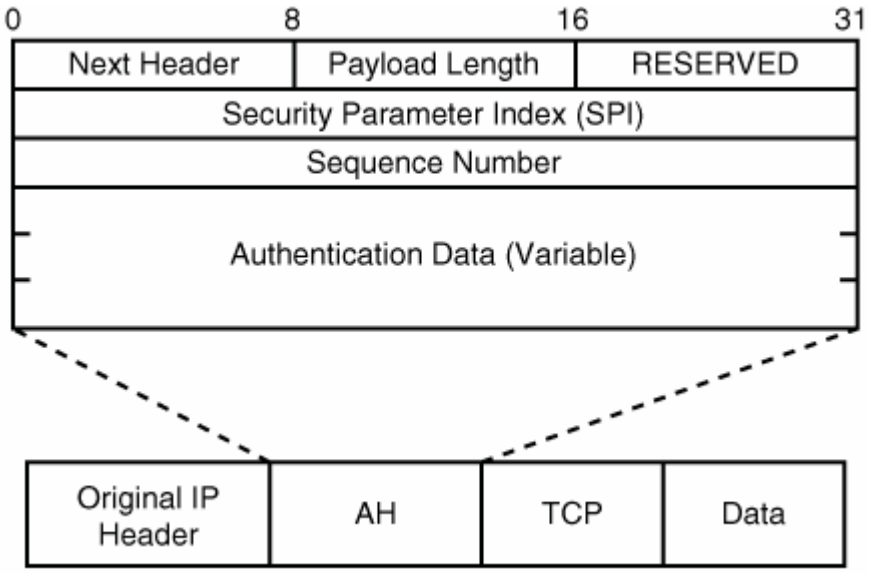
3.2.2. Authentication Header (AH)

AH zapewnia:

- Bezpołączeniową integralność danych
- Uwierzytelnienie pochodzenia danych
- Zabezpieczenie przed atakami powtórzeniowymi (opcjonalnie)

W porównaniu z ESP nie zapewnia poufności.

AH jest protokołem IP identyfikowanym numerem 51 w nagłówku IP.



- **Next Header** – wskazuje co następuje po nagłówku AH. W trybie transportowym będzie to wartość protokołu warstwy wyższej (np. TCP,UDP), a w trybie tunelowania wartość będzie ustawiona na 4
- **Payload Lenght** – długość pola Payload
- **SPI** – to samo co w ESP
- **Authentication Data** – podobnie jak w ESP, z tą różnicą, że przed obliczeniem hash'a wszystkie informacje zawarte w nagłówku IP, które podczas transmisji ulegają zmianie (np. TTL) są zerowane

3.3. Zarządzanie kluczami

Do ochrony integralności i poufności transmisji z użyciem mechanizmów IPSec mogą być wykorzystywane różne algorytmy. Ze względów wydajnościowych, są to głównie algorytmy symetryczne, dlatego też należy zapewnić bezpieczną wymianę kluczy pomiędzy partycypującymi stronami.

Klucze niezbędne do działania wspomnianych algorytmów można skonfigurować na urządzeniach statycznie lub też wykorzystać mechanizmy ich automatycznej generacji i wymiany.
Wszystkie powyższe elementy (algorytmy i niezbędne do ich działania dane – np. klucze) przechowywane są w postaci asocjacji bezpieczeństwa (Security Associations – SA) opisanych dalej w sekcji 5.2.

Do automatycznego tworzenia asocjacji bezpieczeństwa pomiędzy komunikującymi się stronami, na potrzeby mechanizmów IPSec wykorzystywany jest protokół IKE (ang. *Internet Key Exchange*).
Odbywa się to 2 fazach:

1. Faza pierwsza polega na:
    a. Wyborze algorytmów (np. uwierzytelniania, ochrony poufności, integralności, sposobu ustalania klucza sesji itp.) wykorzystywanych w ramach fazy 1 i 2 protokołu IKE.
    b. Wzajemnym uwierzytelnieniu komunikujących się stron.
    c. Ustaleniu symetrycznego klucza sesji komunikacyjnej pomiędzy nimi.
    Wynikiem fazy 1 IKE jest utworzenie SA zwanej ISAKMP SA. Służy ona ochronie (poufność i integralność) dalszej komunikacji z użyciem protokołu IKE – czyli ochronie komunikacji odbywającej się w ramach fazy 2 IKE.
    Asocjacja ISAKMP SA jest dwukierunkowa – tzn. ta sama asocjacja jest stosowana do ochrony ruchu IKE w obu kierunkach.
2. Faza druga odpowiada za utworzenia SA zwanej IPSec SA, która będzie używana do ochrony ruchu użytkownika z użyciem mechanizmów IPSec (protokołów ESP i AH).
    Komunikacja między stronami realizowana w ramach fazy 2 IKE odbywa się z użyciem bezpiecznego kanału łączności utworzonego w fazie 1 poprzez wynegocjowanie ISAKMP SA.
    IPSec SA jest jednokierunkowa, więc do obsługi pojedynczego połączenia to potrzebne są co najmniej dwie.

Asocjacje bezpieczeństwa mają określony czas życia – tzn. okres przez który mogą być wykorzystywane. Po jego upływnie muszą zostać zastąpione nowymi. Czas życia może być określony w sekundach lub bajtach przenoszonego ruchu sieciowego.
ISAKMP SA ma najczęściej dłuższy czas życia od IPSec SA, stąd możliwe jest wielokrotne powtarzanie samej fazy 2, pod warunkiem, że ISAKMP SA pozostaje ważna.

4. Składnia poleceń – podstawowa obsługa IP i tunele

Obsłudze elementów związanych z protokołami IP oraz tuneli statycznych służy w systemie Linux polecenie **ip** o składni podanej poniżej.
Parametry polecenia **ip** można skracać:

Parametr w pełnym brzmieniu	Skrót
address	a
route	r
tunnel	t
link	l
add	a
delete	d
flush	f
list	ls
set	s

4.1. Ogólna informacja na temat konfiguracji interfejsów sieciowych

Wyświetleniu ogólnej informacji o konfiguracji interfejsów sieciowych służy polecenie:

`ip address list`

czyli w skrócie: **`ip a ls`**

a ponieważ ls jest przyjmowane domyślnie, jeśli występowałoby na końcu podawanego polecenia: **`ip a`**

Przykładowy wynik:

```
1: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
    link/ether 00:07:e9:07:70:da brd ff:ff:ff:ff:ff:ff
    inet 10.15.1.1/24 brd 10.15.255.255 scope global eth0
    inet6 fe80::207:e9ff:fe07:70da/64 scope link
       valid_lft forever preferred_lft forever
2: tunel0@NONE: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1480 qdisc noqueue
    link/sit 10.10.1.4 peer 10.11.2.5
    inet6 fe80::9913:31f9/128 scope link
       valid_lft forever preferred_lft forever
```

Widoczne są 2 interfejsy: eth0 i tunel0.

Interfejs typu Ethernet

W pierwszej linii wpisu dotyczącego interfejsu typu Ethernet widać:

```
1: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
```

Oznacza to, że:

- „**1:**” – interfejs jest pierwszym w kolejności (bez większego znaczenia),
- „**eth0**:” nazywa się eth0,
- „**<BROADCAST,MULTICAST,UP,LOWER_UP>**” – jest interfejsem działającym w technice rozgłoszeniowej („BROADCAST”) czyli do jednej sieci fizycznej można podłączyć wiele urządzeń; obsługuje ruch multicastowy („MULTICAST”); jest uruchomiony („UP”); ma podłączone medium transmisyjne, w tym przypadku kabel („LOWER_UP”).
- „**mtu 1500**” – podaje maksymalną wielkość porcji danych, którą może przesłać interfejs. Jeśli trzeba przesłać przez niego większą porcję, będzie ona podzielona (fragmentowana). W tym przypadku jest to 1500 bajtów.
- „**qdisc pfifo_fast qlen 1000**” – parametry bufora wyjściowego, w którym gromadzone są dane przeznaczone do wysłania przez dany interfejs.

Druga linia:

```
    link/ether 00:07:e9:07:70:da brd ff:ff:ff:ff:ff:ff
```

Słowo „**link**” oznacza, że linia podaje informacje na temat techniki transmisyjnej wykorzystywanej przez interfejs i używanej tam adresacji. „**ether**” jest nazwą konkretnej techniki transmisji – w tym przypadku Ethernetu. Dalej następuje adres tej techniki, który dany interfejs posiada – czyli w tym przypadku adres Ethernet MAC „**00:07:e9:07:70:da**”. Ostatnim element jest zależny od konkretnej techniki transmisji – w przypadku Ethernetu jest to „**brd**”, czyli adres rozgłoszeniowy, ważny dla wielu kluczowych mechanizmów tej techniki.

Kolejne linie:

```
    inet 10.15.1.1/24 brd 10.15.255.255 scope global eth0
    inet6 fe80::207:e9ff:fe07:70da/64 scope link
       valid_lft forever preferred_lft forever
    inet6 2001:4070:11:300::1/64 scope global
       valid_lft forever preferred_lft forever
```

zawierają listę adresów IPv4 “inet” lub IPv6 “inet6” przypisanych do interfejsu.

Adresy podawane są w formacie <adres IP>/<długość maski>.

Zakres ważności („**scope**”) jest ważnym parametrem. Podczas laboratorium będziemy mieli do czynienia z adresami o zakresach **global** i **link**.

- **Zakres global** – adres w unikalny sposób identyfikuje dany komputer. Adres taki nie może zostać przypisany w 2 miejscach jednocześnie. Ruch adresowany do pod adresy o zakresie globalnym, może być przekazywany przez routery.
- **Zakres link** – adresy tego typu są ważne tylko w obrębie pojedynczej sieci fizycznej. W związku z tym mogą się powtarzać, jeśli są w różnych sieciach fizycznych. Można je wykorzystać do przesłania danych do innego urządzania wyłącznie, jeśli jesteśmy z nim wspólnie podłączeni do takiej sieci.

Interfejs tunelowy

Dla porównania, widoczny na przykładzie interfejs tunelowy:

```
2: tunel0@NONE: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1480 qdisc noqueue
```

- „**2:**” – Jest 2 w kolejności na liście.
- „**tunel0@NONE**” – Nazwa interfejsu to “**tunel0**”. „**@NONE**” oznacza, że nie określamy, przez który interfejs fizyczny będzie wysyłany ruch który spróbujemy przesłać przez interfejs tunelowy. W takim przypadku decyzja ta jest podejmowana automatycznie, na podstawie adresu (protokołu przenoszącego) drugiego końca tunelu, podanego przy jego tworzeniu.
- „**<POINTOPOINT,NOARP,UP,LOWER_UP>**” – interfejs ten jest interfejsem typu punkt-punkt (“**POINTTOPOINT**”), co oznacza, że nie może on łączyć więcej niż 2 urządzeń (w przeciwieństwie do np. interfejsów typu Ethernet). W związku z tym nie wykorzystuje mechanizmów pozwalających na określenie adresu drugiego punktu docelowego („**NOARP**”) – co zostanie wysłane przez jeden koniec tunelu, zostanie odebrane przez drugi jego koniec. „**UP**” oznacza, że interfejs jest włączony. „**LOWER_UP**” nie ma dla nas praktycznego znaczenia.
- „**mtu 1480**” – Maksymalna wielkość porcji danych to 1480 bajtów.
- „**qdisc noqueue**” – Podany interfejs tunelowy nie używa buforowania wyjściowego, bo dane wysyłane przez ten interfejs zostaną enkapsulowane i po enkapsulacji wysłane przez interfejs fizyczny (który z kolei używa buforowania wyjściowego). Nie ma więc sensu buforować danych dwa razy: najpierw na poziomie interfejsu tunelowego, a potem na poziomie interfejsu fizycznego.

Druga linia:

```
    link/sit 10.10.1.4 peer 10.11.2.5
```

Podobnie jak w przypadku interfejsu typu Ethernet, słowo „**link**” oznacza, że linia podaje informacje na temat techniki transmisyjnej wykorzystywanej przez interfejs i używanej tam adresacji. „**sit**” oznacza, że techniką używaną przez interfejs do przysłania ruchu jest tunelowanie typu SIT (IPv6 w IPv4). Dalej podane są adresy, pomiędzy którymi zestawiony jest tunel – najpierw nasz adres, a potem (po słowie „**peer**”) adres drugiego końca tunelu.

Dalsze linie, tak samo jak w przypadku interfejsu typu Ethernet, podają przypisane do interfejsu adresy IP.

4.2. Pozostałe operacje na interfejsach

```
ip link set up dev <interfejs> - włączenie interfejsu
ip link set down dev <interfejs> - wyłącznie interfejsu
ip link set <interfejs> up - włączenie interfejsu
ip link set <interfejs> down - wyłącznie interfejsu
```

4.3. Operacje na adresach IP

```
ip address list – lista adresów
ip -6 address list – lista adresów IPv6
ip address add <adres>/<maska> dev <interfejs> – dodanie adresu do interfejsu
ip address del <adres>/<maska> dev <interfejs> - usunięcie adresu z interfejsu
ip address flush dev <interfejs> - usunięcie wszystkich adresów IPv4 i IPv6 z interfejsu
ip -4 address flush dev <interfejs> - usunięcie wszystkich adresów IPv4 z interfejsu
ip -6 address flush dev <interfejs> - usunięcie wszystkich adresów IPv6 z interfejsu
```

4.4. Operacje na trasach routingu

```
ip route list – wyświetla trasy routingu IPv4
ip -6 route list – wyświetla trasy routingu IPv6

ip route add <adres sieci>/<maska> dev <interfejs> - dodanie trasy routingu IPv4 (dostarczanie zdalne)
ip route add <adres sieci>/<maska> via <adres routera> - dodanie trasy routingu IPv4 (dostarczanie
zdalne)
ip route del <adres sieci>/<maska> [via <adres routera>] dev <interfejs> - usunięcie trasy routingu IPv4

ip -6 route add <adres sieci>/<maska> dev <interfejs> - dodanie trasy routingu IPv6 (dostarczanie
lokalne)
ip -6 route add <adres sieci>/<maska> via <adres routera> - dodanie trasy routingu IPv6 (dostarczanie
zdalne)
ip -6 route del <adres sieci>/<maska> [via <adres routera>] dev <interfejs> - usunięcie trasy routingu
IPv6
```

4.5. Tunele statyczne

4.5.1. Lista tuneli

Wyświetlenie listy interfejsów na komputerze lokalnym, odpowiadających stworzonym tunelom możliwe jest dzięki poleceniu:

`ip tunnel show`

Przykładowe wyniki:

`tunel0: ipv6/ip  remote 10.10.1.4  local 10.11.2.5  ttl 255`

Tunel typu **sit** (ipv6 w ipv4) pomiędzy adresami IPv4: **10.10.1.4** (na naszym komputerze) i **10.11.2.5** (na komputerze odległym). Na naszym komputerze interfejs odpowiadający tunelowi nosi nazwę **tunel0**.

UWAGA: Z reguły wygodniej jest posłużyć się, opisanym wcześniej, poleceniem **`ip address list`** (**`ip a`**).

4.5.2. Dodawanie tuneli

Składnia polecenia różni się nieco, w zależności od tego, czy do przenoszenia ruchu tunel będzie wykorzystywał protokół IPv4 czy IPv6.

Dla tuneli wykorzystujących IPv4 jako protokół przenoszący (ipip, sit, gre):

`ip tunnel add name <nazwa> mode <typ> local <adres_lokalny> remote <adres_zdalny>`

Dla tuneli wykorzystujących IPv6 jako protokół przenoszący (ip6ip6, ipip6, gre):

`ip -6 tunnel add name <nazwa> mode <typ> local <adres_lokalny> remote <adres_odległy>`

Gdzie:

<nazwa> – nazwa nowego interfejsu, który na lokalnym komputerze umożliwiał skorzystanie z tunelu. Warto zauważyć, że na komputerze zdalnym nazwa interfejsu, którym kończy się nasz tunel, może być inna niż u nas. Nazwa ma wyłączenie znaczenie lokalne.
<typ> – typ tunelu: sit, ipip, gre, ip6ip6, ipip6
<adres_lokalny> – adres lokalnego (znajdującego się na aktualnie konfigurowanym urządzeniu) punktu końcowego tunelu
<adres_odległy> – adres odległego (nieznajdującego się na aktualnie konfigurowanym urządzeniu) punktu końcowego tunelu

Na przykład:
ip tunnel add name sit0 mode sit local 10.10.1.1 remote 10.10.1.100
Spowoduje stworzenie tunelu typu **sit** (ipv6 w ipv4) pomiędzy lokalnym adresem **10.10.1.1**, a odległym adresem **10.10.1.100**. Interfejs identyfikujący na lokalnym komputerze stworzony tunel, będzie nosił nazwę **sit0**.

4.5.3. Usuwanie tunelu

Usunięcie tunelu o podanej nazwie (interfejsu).
ip tunnel del name <nazwa>

5. Składnia poleceń – IPSec

Konfiguracji mechanizmów IPSec w systemie Linux dokonujemy za pomocą pojedynczego polecenia: **setkey**
W zależności od parametrów, pozwala ono na, między innymi:
setkey –D - wyświetlenie wszystkich asocjacji IPsec (SA)
setkey –DP – wyświetlenie wszystkich polityk IPsec (SP)
setkey –F – usunięcie wszystkich asocjacji IPsec (SA)
setkey –FP – usunięcie wszystkich polityk IPsec (SP)
setkey –f <nazwa_pliku> - wczytanie pliku konfiguracyjnego o podanej nazwie

Jak widać, większość z poleceń dotyczy odczytania aktualnej konfiguracji oraz jej usunięcia. W celu wprowadzenia nowej konfiguracji mechanizmów IPSec, należy przygotować tekstowy plik konfiguracyjny, zawierający odpowiednie polecenia , a następnie wczytać go, korzystając z polecenia: **setkey –f <nazwa>**
W pliku konfiguracyjnym, każde polecenie zapisujemy w osobnej linii, którą kończymy średnikiem „;”.

Na początku pliku konfiguracyjnego warto umieścić dwa poniższe polecenia, celem usunięcia istniejących polityk i asocjacji IPSec. Dzięki temu, wczytanie takiego pliku konfiguracyjnego, spowoduje zastąpienie istniejącej konfiguracji nową, a nie tylko dopisanie do niej nowych elementów.

flush; - usunięcie wszystkich asocjacji IPsec (SA)
spdflush; – usunięcie wszystkich polityk IPsec (SP)

W poniższych podrozdziałach (5.1 i 5.2) opisano polecenia, którymi należy posłużyć się w pliku konfiguracyjnym, celem utworzenia nowych polityk i asocjacji IPSec.

5.1. Polityki bezpieczeństwa (Security Policies – SPs)

Polityka bezpieczeństwa (SP) pozwala na określnie, czy dany ruch ma podlegać zabezpieczeniu z użyciem protokołu IPsec.
Składnia pozwalająca na utworzenie nowego wpisu w bazie polityk bezpieczeństwa jest następująca:
`spdadd <src_range> <dst_range> <upperspec> -P <direction> <policy> ;`

Elementy `<src_range>` i `<dst_range>` pozwalają na określenie grup źródłowych i docelowych adresów IP, których ma dotyczyć dany wpis. Każdy z tych elementów może mieć postać:

- adresu IP, np.: `192.168.1.1`, `3ffe:8010:7:1703::11`
- prefiksu IP, np.: `192.168.1.0/24`, `3ffe:8010:7:1703::/64`

Można też dodać do każdego z powyższych formatów opcjonalny numer portu, wpisywany w nawiasach kwadratowych, np.: `192.168.1.1[110]`, `3ffe:8010:7:1703::/64[25]`
Jeśli nie podamy numeru portu, oznacza to dowolny port.
Kolejny element (`<upperspec>`), pozwala na określenie protokołu, którego dotyczyć ma dany wpis. Może to być dowolny z protokołów zdefiniowanych w pliku /etc/protocols, np.: `ip`, `ip6`, `icmp6`, `tcp`. Wartość `any` oznacza dowolny protokół.
Parametr `<direction>` określa, czy polityka bierze pod uwagę ruch:

- `in` – przychodzący (nasz komputer jest jego adresatem),
- `out` – wychodzący (nasz komputer jest jego nadawcą),
- `fwd` – przekazywany (nasz komputer pełni rolę routera – odbiera i przekazuje dalej dany ruch, lecz nie jest ani jego ostatecznym adresatem, ani pierwotnym nadawcą).

Jeśli dany pakiet pasuje do wszystkich powyższych warunków, tzn.:

- adres źródłowy zawiera się w `<src_range>` (oraz port źródłowy odpowiada wartości w nawiasach kwadratowych, jeśli zostały dodane po `<src_range>`),
- adres docelowy zawiera się w `<dst_range>` (oraz port docelowy odpowiada wartości w nawiasach kwadratowych, jeśli zostały dodane po `<dst_range>`),
- protokół odpowiada wartości `<upperspec>`,
- kierunek przepływu ruchu zgodny jest z `<direction>`,

podlega on zdefiniowanej dalej polityce: `<policy>`.

Polityka `<policy>` może przyjmować jedną z następujących wartości:

- `none` - pasujący do niej ruch nie jest w żaden sposób przetwarzany przez mechanizmy IPsec,
- `discard` - pasujący do niej ruch jest bezwarunkowo odrzucany,
- `ipsec <protocol>/<mode>/<src>-<dst>/<level>` - pasujący do niej ruch jest zabezpieczany z użyciem mechanizmów IPsec.

Jak widać, polityka nakazująca zabezpieczenie z użyciem mechanizmów IPsec wymaga podania dodatkowych informacji w postaci:
`<protocol>/<mode>/<src>-<dst>/<level>`

Parametr `<protocol>` oznacza, w tym przypadku, typ protokołu zabezpieczeń IPsec, który ma być użyty do ochrony ruchu. Interesujące nas protokoły to:

- `ah` – pozwala na ochronę integralności transmisji,
- `esp` – pozwala na ochronę poufności transmisji.

Parameter <mode> pozwala na wybór transportowego lub tunelowego trybu działania mechanizmów IPsec:

- transport – tryb transportowy,
- tunnel – tryb tunelowy.

Para parametrów <src>-<dst> używana jest w tylko w trybie tunelowym, gdzie <src> jest adresem IP jednego, a <dst> drugiego końca tunelu. Adresy te rozdzielone są znakiem „-„. W trybie transportowym nie podajemy tu ani adresów <src> oraz <dst>, ani znaku „-„. (przykład poniżej)
Parametr <level> określa sposób, w jaki przetwarzany jest ruch pasujący do danej polityki. Ma kilka możliwych wartości, z których interesują nas:

- use – ruch zostanie zabezpieczony zgodnie z ustawieniami parametru <protocol>, jeśli istnieje odpowiednia asocjacja bezpieczeństwa (SA), pasująca do danego ruchu. Jeśli jej nie ma, ruch nie zostanie zabezpieczony.
- require – uch zostanie zabezpieczony zgodnie z ustawieniami parametru <protocol>. Jeśli istnieje odpowiednia asocjacja bezpieczeństwa (SA), pasująca do danego ruchu – zostanie ona użyta. Jeśli jej nie ma, zostaną wykorzystane mechanizmy automatyczne (np. proces Racoon) w celu jej utworzenia. Jeśli się to nie uda, ruch zostanie odrzucony.

Należy dodać, iż można użyć kilku grup paramterów <protocol>/<mode>/<src>-<dst>/<level>, np. w celu zastosowania zarówno protokołu AH jak i ESP. W takim przypadku należy najpierw umieścić grupę dotyczącą protokołu ESP, a następnie, oddzieloną spacją, grupę dotyczącą protokołu AH.

Przykłady elementu <policy>:
Odrzucenie ruchu:

```
discard;
```

Wymagane użycie protokołu AH w trybie transportowym:

```
ipsec ah/transport//require;
```

Wymagane użycie protokołu ESP w trybie tunelowym. Tunel zestawiany jest pomiędzy adresami 192.168.1.1 oraz 192.168.1.2:

```
ipsec esp/tunnel/192.168.1.1-192.168.1.2/require;
```

Jeśli istnieją pasujące SA, użyj protokołów ESP i AH w trybie transportowym. Należy pamiętać, że każdy z protokołów wymaga osobnego SA. Jeśli będzie tylko jedno, to zostanie użyty tylko jeden z protokołów ESP/AH:

```
ipsec esp/transport//use ah/transport//use;
```

Koniec polecenia spdadd zaznaczamy średnikiem „;”.

Przykłady kompletnych poleceń pliku konfiguracyjne tworzących SP:
Odrzucaj (discard) generowany przez dany węzeł (out) ruch TCP o adresie źródłowym 192.168.1.1 i adresie docelowym 192.168.1.2, jeśli dodatkowo port źródłowy i docelowy jest równy 25:

```
spdadd 192.168.1.1[25] 192.168.1.2[25] tcp -P out discard;
```

Jeśli istnieje odpowiednie SA, zabezpieczaj protokołem ESP (w trybie transportowym) ruch UDP który jest wysyłany do danego węzła (in) z adresu 3ffe:8010:7:1703::1 (i dowolnego portu) z adresem docelowym 3ffe:8010:7:1703::100 i docelowym portem 123:

```
spdadd 3ffe:8010:7:1703::1 3ffe:8010:7:1703::100[123] udp -P in ipsec esp/transport//use;
```

5.2. Asocjacje bezpieczeństwa (Security associations – SAs)

Asocjacja bezpieczeństwa (SA) opisuje szczegółowy sposób ochrony ruchu pomiędzy zdefiniowanymi adresami IP <src> i <dst>.

Asocjacja jest używana, jeśli istnieje polityka bezpieczeństwa (SP), która nakazuje wykorzystanie IPsec'a do ochrony ruchu między tymi adresami - wówczas z asocjacji (SA) odczytywane są niezbędne szczegóły, takie jak klucze, algorytmy itp., które należy zastosować dla każdego z wymaganych polityką protokołów zabezpieczeń.
Drugim przypadkiem, kiedy określone SA jest wykorzystywane, jest sytuacja, gdy urządzenie odbierze pakiet zabezpieczony protokołem IPSec. Wówczas odczytywany jest z niego identyfikator SPI, określający SA użyte do jego zabezpieczenia. Następnie, na odbierającym urządzeniu, wyszukiwana jest SA z takim samym identyfikatorem SPI i to ona jest używana do przetworzenia pakietu (np. odszyfrowania, czy sprawdzenia integralności).

Asocjacja może być tworzona ręcznie. Wówczas należy pamiętać o utworzeniu SA dla każdej pary adresów, między którymi będziemy zabezpieczać ruch, czyli dla każdej, która może zostać zakwalifikowana do przez jakąś politykę (SP) do zabezpieczenia z użyciem IPsec. Jeśli dla takiej pary nie będzie odpowiedniej SA, łączność może zostać zablokowana (w przypadku polityki typu **require**) lub nie zostać zabezpieczona (w przypadku polityki typu **use**).

Drugą metodą tworzenia asocjacji jest wykorzystanie zewnętrznego procesu (np. Racoon). Wówczas, gdy jakaś polityka typu **require** zażąda zabezpieczenia ruchu, a nie będzie istniała odpowiednia asocjacja, zostanie podjęta próba automatycznego jej utworzenia poprzez negocjację odpowiednich parametrów między zainteresowanymi stronami.

Poniżej podano składnię polecenia pliku konfiguracyjnego umożliwiającego dodanie SA:
`add <src> <dst> <proto> <spi> -m <tryb> <algorytm> ;`

Gdzie:
`<src>` - źródłowy adres IP
`<dst>` - docelowy adres IP
`<proto>` - protokół, który ma być użyty do zabezpieczenia komunikacji: `esp` (poufność) lub `ah` (integralność)
`<spi>` - unikalny identyfikator asocjacji (musi być > 255), przesyłany wraz z chronionym ruchem. Dzięki niemu strona odbierająca ruch wie, jakiej SA użyć do jego obsługi. Wynika stąd, że SPI odpowiadających sobie (takich samych) asocjacji na komunikujących się urządzeniach muszą być ze sobą zgodne.
`<tryb>` - SA dla trybu **tunelowego** (`tunnel`) lub **transportowego** (`transport`)
`<algorytm>` - jedno z poniższych:

- `-E <algorytm_esp> <klucz_esp>` - określenie algorytmu i klucza dla protokołu ESP,
- `-A <algorytm_ah> <klucz_ah>` - określenie algorytmu i klucza dla protokołu AH,
- `-E <algorytm_esp> <klucz_esp> -A <algorytm_ah> <klucz_ah>` - określenie algorytmu i klucza dla protokołu ESP i AH (muszą być w tej kolejności). W opcji <proto> (powyżej) należy podać ESP.

Możliwe wartości `<algorytm_esp>` i `<algorytm_ah>` zebrano w poniższych tabelach. Podano w nich też wymagane dla każdego z algorytmów długości klucza – musi on mieć DOKŁADNIE odpowiednią długość. Trzecia kolumna tabeli podaje ewentualne odniesienie do dokumentu opisującego dany algorytm.

Klucze `<klucz_esp>` i `<klucz_ah>` możemy podać w jednej z 2 postaci:

- ciągu znaków ASCII ujętego w cudzysłowia, np. "12345678" (klucz 64 bitowy),
- wartości szesnastkowych poprzedzonych przedrostkiem 0x, np.: 0xb27436a4ff1f (klucz 64 bitowy).

Algorytmy AH

algorithm	keylen (bits)	
hmac-md5	128	ah: rfc2403
hmac-sha1	160	ah: rfc2404
keyed-md5	128	ah: 96bit ICV (no document)
keyed-sha1	160	ah: 96bit ICV (no document)
null	0 to 2048	for debugging
hmac-sha256	256	ah: 96bit ICV (draft-ietf-ipsec-ciph-sha-256-00)
hmac-sha384	384	ah: 96bit ICV (no document)
hmac-sha512	512	ah: 96bit ICV (no document)
hmac-ripemd160	160	ah: 96bit ICV (RFC2857)
aes-xcbc-mac	128	ah: 96bit ICV (RFC3566)

Algorytmy ESP

algorithm	keylen (bits)	
des-cbc	64	esp-old: rfc1829, esp: rfc2405
3des-cbc	192	rfc2451
null	0 to 2048	rfc2410
blowfish-cbc	40 to 448	rfc2451
cast128-cbc	40 to 128	rfc2451
des-deriv	64	ipsec-ciph-des-derived-01
3des-deriv	192	no document
rijndael-cbc	128/192/256	rfc3602
twofish-cbc	0 to 256	draft-ietf-ipsec-ciph-aes-cbc-01
aes-ctr	160/224/288	draft-ietf-ipsec-ciph-aes-ctr-03
camellia-cbc	128/192/256	rfc4312

Przykładowe definicje SA:

Zabezpieczenie komunikacji pomiędzy adresami 10.10.1.4 I 10.11.5.7, realizowane w trybie transportowym z użyciem protokołu ESP, odbywać się będzie przy wykorzystaniu algorytmu des-cbc i klucza podanego w ASCII „12345678”.

```
add 10.10.1.4 10.11.5.7 esp 0x1000 –m transport –E des-cbc „12345678”
```

Zabezpieczenie komunikacji pomiędzy adresami 10.10.1.4 I 10.11.5.7, realizowane w trybie transportowym z użyciem protokołu AH, odbywać się będzie przy wykorzystaniu algorytmu hmac-md5 i klucza podanego w ASCII „1234567890123456”.

```
add 192.168.1.1 192.168.2.5 ah 0x1a01 –m transport –A hmac-
md5„1234567890123456”
```

Zabezpieczenie komunikacji pomiędzy adresami 10.10.1.4 I 10.11.5.7, realizowane w trybie transportowym z użyciem protokołów ESP i AH, odbywać się będzie przy wykorzystaniu algorytmów i kluczy:

- dla ESP: des-cbc i klucza podanego w formie szesnastkowej 0x21435721af4e,
- dla AH: hmac-md5 i klucza podanego w ASCII „1234567890123456”.

```
add 2001:1::3 2001:12af::fed1 esp 0x2f02 –m transport –E des-cbc
0x21435721af4e –A hmac-md5 „1234567890123456”
```