Securitate pe Internet (*IPSec*)

Prof. Dr. Adrian Atanasiu

April 12, 2016

Definiție

*IPSec (Internet Protocol Security) este o secvență de protocoale având ca scop securizarea protocoalelor de comunicare prin Internet (**I**nternet **P**rotocol – IP), autentificând și criptând fiecare pachet IP de șir de date.*

Definiție

*IPSec (Internet Protocol Security) este o secvență de protocoale având ca scop securizarea protocoalelor de comunicare prin Internet (**I**nternet **P**rotocol – IP), autentificând și criptând fiecare pachet IP de șir de date.*

IPSec include de asemenea protocoale de autentificare reciprocă între parteneri la începutul sesiunii și stabilirea cheilor criptografice care vor fi folosite pe durata sesiunii de comunicare.

Definiție

*IPSec (Internet Protocol Security) este o secvență de protocoale având ca scop securizarea protocoalelor de comunicare prin Internet (**I**nternet **P**rotocol – IP), autentificând și criptând fiecare pachet IP de șir de date.*

IPSec include de asemenea protocoale de autentificare reciprocă între parteneri la începutul sesiunii și stabilirea cheilor criptografice care vor fi folosite pe durata sesiunii de comunicare.
IPSec mai este utilizat la protejarea fluxurilor de date între două entități (de exemplu un calculator și un server), între două gateway-uri (porți) de securitate (de exemplu routere sau firewalls), sau între un gateway de securitate și un utilizator.

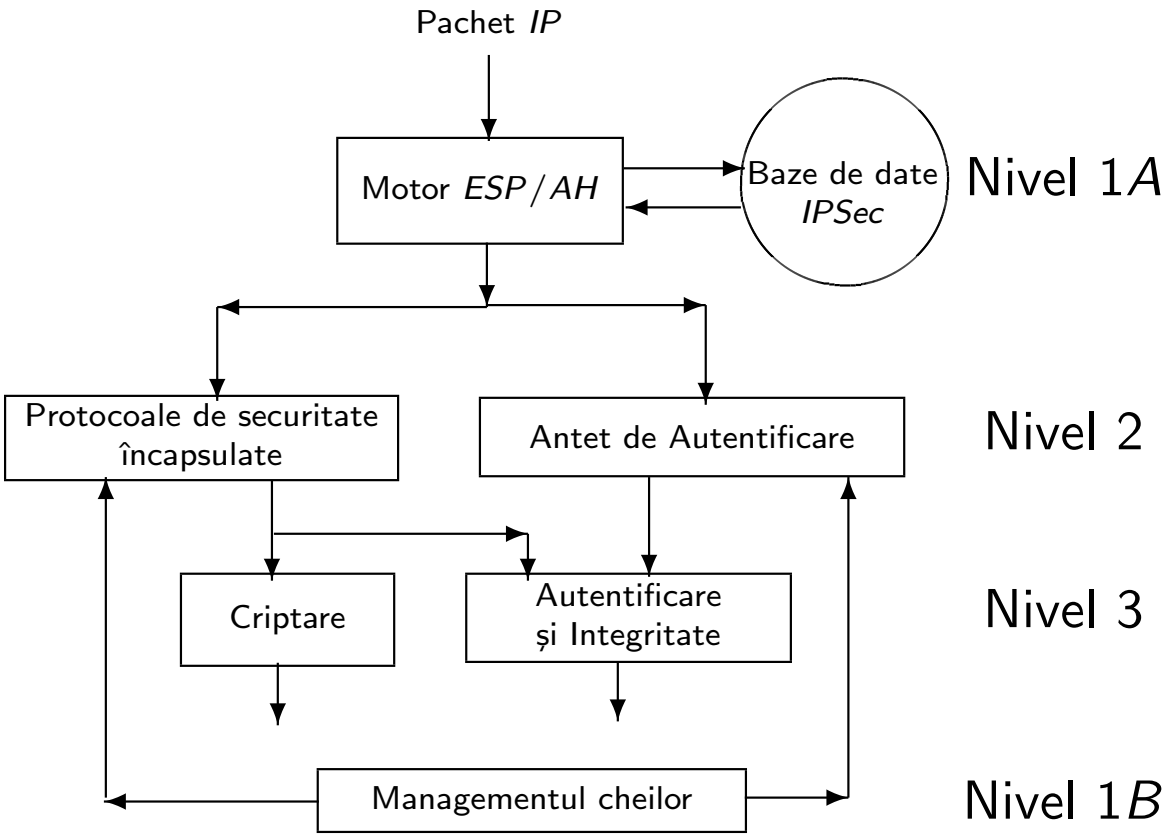
Istoric, *IPSec* este un succesor al standardului *ISO NLSP* (**N**etwork **L**ayer **S**ecurity **P**rotocol).
NLSP era definit pe baza protocolului *SP3* publicat de *NIST*, care desemna un proiect de securitate a datelor prin rețea aparținând *NSA* (**N**ational **S**ecurity **A**gency).

Istoric, *IPSec* este un succesor al standardului *ISO NLSP* (**N**etwork **L**ayer **S**ecurity **P**rotocol).
NLSP era definit pe baza protocolului *SP3* publicat de *NIST*, care desemna un proiect de securitate a datelor prin rețea aparținând *NSA* (**N**ational **S**ecurity **A**gency).

Oficial, standardele *IPSec* sunt specificate de *IETF* (**I**nternet **E**ngineering **T**ask **F**orce) printr-o serie de comentarii privind diverse componente și extensii (inclusiv definirea termenilor și a nivelurilor de securitate).

Arhitectura *IPSec*

În termeni generali, arhitectura unui *IPSec* este:



Bazele de date *IPSec*

De obicei sunt trei baze de date standard la acest nivel:

Bazele de date *IPSec*

De obicei sunt trei baze de date standard la acest nivel:

1. *SPD* (*Security Policy Database*): Specifică controlul de trafic *IP* al ambelor părți implicate (expeditor și destinatar).

Bazele de date *IPSec*

De obicei sunt trei baze de date standard la acest nivel:

1. *SPD* (*Security Policy Database*): Specifică controlul de trafic *IP* al ambelor părți implicate (expeditor și destinatar).
2. *PAD* (*Peer Authorisation Database*): asigură legătura între *SPD* și un protocol de gestiune a securității (de exemplu *IKE*).

3 *SAD* (*Security Association Database*): Conține parametrii fiecărei asocieri de securitate stabilite (*SA*). Fiecare *SA* are un element în *SAD*.

3 *SAD* (*Security Association Database*): Conține parametrii fiecărei asocieri de securitate stabilite (*SA*). Fiecare *SA* are un element în *SAD*.
În general, un *SAD* conține:

- Indexul parametrului de securitate (*SPI*).

3 *SAD* (*Security Association Database*): Conține parametrii fiecărei asocieri de securitate stabilite (*SA*). Fiecare *SA* are un element în *SAD*.
În general, un *SAD* conține:

- Indexul parametrului de securitate (*SPI*).
- Serviciile de securitate încapsulate (*ESP* – *Encapsulated Security Payload/Protocols*) împreună cu datele curente transmise: algoritmi de criptare și integritate, modul de generare al cheilor, valorile inițiale (*IV*) etc.

3 *SAD* (*Security Association Database*): Conține parametrii fiecărei asocieri de securitate stabilite (*SA*). Fiecare *SA* are un element în *SAD*.
În general, un *SAD* conține:

- Indexul parametrului de securitate (*SPI*).
- Serviciile de securitate încapsulate (*ESP* – *Encapsulated Security Payload/Protocols*) împreună cu datele curente transmise: algoritmi de criptare și integritate, modul de generare al cheilor, valorile inițiale (*IV*) etc.
- Antetul de autentificare (*AH*), cu algoritmul de autentificare, cheile *MAC* etc.

3 *SAD* (*Security Association Database*): Conține parametrii fiecărei asocieri de securitate stabilite (*SA*). Fiecare *SA* are un element în *SAD*.
În general, un *SAD* conține:

- Indexul parametrului de securitate (*SPI*).
- Serviciile de securitate încapsulate (*ESP* – *Encapsulated Security Payload/Protocols*) împreună cu datele curente transmise: algoritmi de criptare și integritate, modul de generare al cheilor, valorile inițiale (*IV*) etc.
- Antetul de autentificare (*AH*), cu algoritmul de autentificare, cheile *MAC* etc.
- Timpul de valabilitate al *SA*.

3 *SAD* (*Security Association Database*): Conține parametrii fiecărei asocieri de securitate stabilite (*SA*). Fiecare *SA* are un element în *SAD*.
În general, un *SAD* conține:

- Indexul parametrului de securitate (*SPI*).
- Serviciile de securitate încapsulate (*ESP* – *Encapsulated Security Payload/Protocols*) împreună cu datele curente transmise: algoritmi de criptare și integritate, modul de generare al cheilor, valorile inițiale (*IV*) etc.
- Antetul de autentificare (*AH*), cu algoritmul de autentificare, cheile *MAC* etc.
- Timpul de valabilitate al *SA*.
- Tipul de protocol *IPSec* ("*tunel*" sau "*transport*") aplicat lui *SA*.

Nivel 2

Cele două protocoale de securitate de la acest nivel (care au sub control și Nivelul 3) sunt:

Nivel 2

Cele două protocoale de securitate de la acest nivel (care au sub control și Nivelul 3) sunt:

1. *AH* (*IP* Authentication Header) folosit pentru autentificare; este bazat pe standardul *RFC* 4302.

Nivel 2

Cele două protocoale de securitate de la acest nivel (care au sub control și Nivelul 3) sunt:

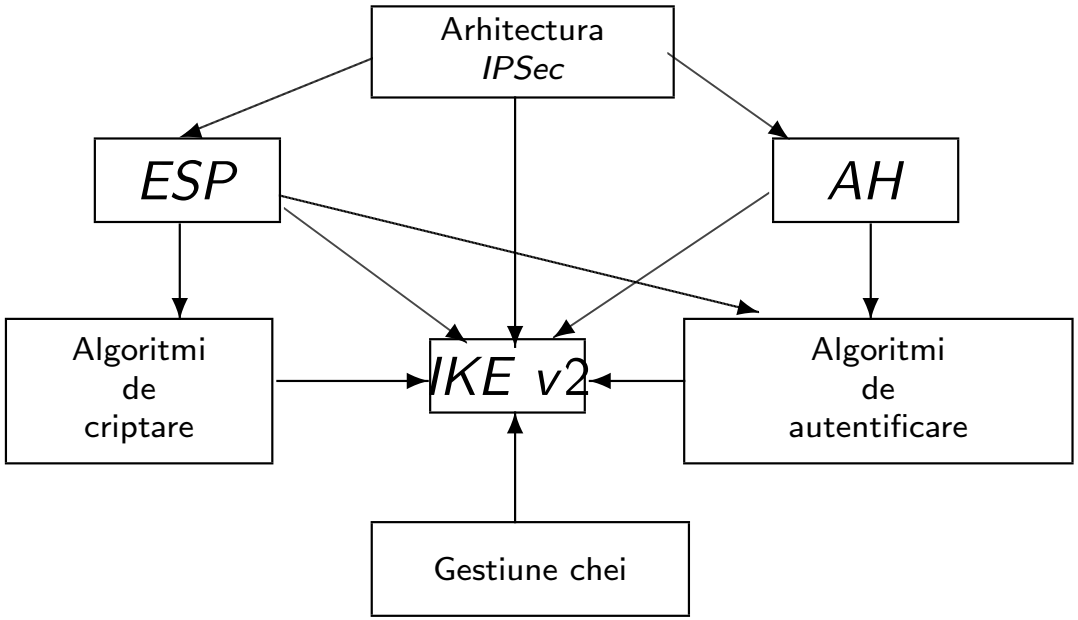
1. *AH* (*IP* Authentication Header) folosit pentru autentificare; este bazat pe standardul *RFC* 4302.
2. *ESP* (bazat pe standardul *RFC* 4303): este folosit pentru criptare și autentificare.

- Nivel 3:
  Fiecare algoritm criptografic pentru autentificare și criptare este definit de un standard *RFC* specific.
  În mod uzual, pentru criptare se folosesc *AES* și *3DES*, iar pentru autentificare și integritate – funcții de dispersie cu cheie.

- Nivel 3:
  Fiecare algoritm criptografic pentru autentificare și criptare este definit de un standard *RFC* specific.
  În mod uzual, pentru criptare se folosesc *AES* și *3DES*, iar pentru autentificare și integritate – funcții de dispersie cu cheie.

- Protocoale de gestiune a cheilor:
  Sunt descrise în *IKEv2* (Internet Key Exchange, version 2), bazate pe un standard din 2005 (*RFC* 4306).

O relație între aceste protocoale:



Negociere *IPSec*

Când un pachet de date este transmis, are loc un protocol de negociere lansat de expeditor (*Alice*) către destinatar (*Bob*).

Negociere *IPSec*

Când un pachet de date este transmis, are loc un protocol de negociere lansat de expeditor (*Alice*) către destinatar (*Bob*).

A. Pachetul de ieșire
Serverul *Alice* vrea să trimită pachetul de date α spre serverul *Bob*.

Negociere *IPSec*

Când un pachet de date este transmis, are loc un protocol de negociere lansat de expeditor (*Alice*) către destinatar (*Bob*).

A. Pachetul de ieșire
Serverul *Alice* vrea să trimită pachetul de date α spre serverul *Bob*. Pentru aceasta se deschide o aplicație *IPSec*, care funcționează după un protocol (numit "negociere").

1. Aplicația apelează stiva *TCP/IP* pentru a prelua α.

1. Aplicația apelează stiva *TCP/IP* pentru a prelua α.
2. α este preluat de un algoritm de negociere $\mathcal{AN}$, care construiește un "înveliș" de protecție.

1. Aplicația apelează stiva *TCP/IP* pentru a prelua α.
2. α este preluat de un algoritm de negociere $\mathcal{AN}$, care construiește un "înveliș" de protecție.
3. $\mathcal{AN}$ caută pachetul α în baza de date a protocoalelor de securitate și decide dacă este necesară o protejare a sa sau doar un permis de trecere *IPSec*.

1. Aplicația apelează stiva *TCP/IP* pentru a prelua α.
2. α este preluat de un algoritm de negociere $\mathcal{AN}$, care construiește un "înveliș" de protecție.
3. $\mathcal{AN}$ caută pachetul α în baza de date a protocoalelor de securitate și decide dacă este necesară o protejare a sa sau doar un permis de trecere *IPSec*. În final, α poate fi
   1. protejat folosind serviciile *IPSec*;

1. Aplicația apelează stiva *TCP/IP* pentru a prelua α.
2. α este preluat de un algoritm de negociere $\mathcal{AN}$, care construiește un "înveliș" de protecție.
3. $\mathcal{AN}$ caută pachetul α în baza de date a protocoalelor de securitate și decide dacă este necesară o protejare a sa sau doar un permis de trecere *IPSec*. În final, α poate fi
   1. protejat folosind serviciile *IPSec*;
   2. eliminat;

1. Aplicația apelează stiva *TCP/IP* pentru a prelua α.
2. α este preluat de un algoritm de negociere $\mathcal{AN}$, care construiește un "înveliș" de protecție.
3. $\mathcal{AN}$ caută pachetul α în baza de date a protocoalelor de securitate și decide dacă este necesară o protejare a sa sau doar un permis de trecere *IPSec*. În final, α poate fi
   1. protejat folosind serviciile *IPSec*;
   2. eliminat;
   3. asociat cu un permis de trecere *IPSec*.

4 Dacă α trebuie protejat, $\mathcal{AN}$ trimite aplicației adresa lui *Bob*, care o caută în *SA* și *SPI* din bazele sale interne de date.

4 Dacă α trebuie protejat, $\mathcal{AN}$ trimite aplicației adresa lui *Bob*, care o caută în *SA* și *SPI* din bazele sale interne de date.

5 Dacă nu a fost negociat încă un *SA* pentru adresa respectivă, atunci aplicația lansează o negociere *IKE* cu adresa respectivă, pentru crearea unui *SA*.

4 Dacă α trebuie protejat, $\mathcal{AN}$ trimite aplicației adresa lui *Bob*, care o caută în *SA* și *SPI* din bazele sale interne de date.

5 Dacă nu a fost negociat încă un *SA* pentru adresa respectivă, atunci aplicația lansează o negociere *IKE* cu adresa respectivă, pentru crearea unui *SA*.

6 După terminarea negocierii, aplicația trimite *SPI* și *SA* spre $\mathcal{AN}$; acesta va construi o protecție pentru α folosind cheia negociată de aplicație.

Pachetul de intrare

Dacă ce *Bob* primește un pachet de date α în portul rezervat negocierii *IKE*, atunci:

Pachetul de intrare

Dacă ce *Bob* primește un pachet de date α în portul rezervat negocierii *IKE*, atunci:

1. Dacă pentru adresa primită nu a fost negociat încă nici un *SA*, atunci $\mathcal{AN}$ va transmite pachetul α aplicației de negociere.

Pachetul de intrare

Dacă ce *Bob* primește un pachet de date α în portul rezervat negocierii *IKE*, atunci:

1. Dacă pentru adresa primită nu a fost negociat încă nici un *SA*, atunci $\mathcal{AN}$ va transmite pachetul α aplicației de negociere.
2. Dacă α are un *SPI* asociat, atunci este căutat *SA*-ul corespunzător în baza de date *IPSec*. Dacă *SPI* nu are corespondent în baza de date, pachetul α este respins.

Pachetul de intrare

Dacă ce *Bob* primește un pachet de date α în portul rezervat negocierii *IKE*, atunci:

1. Dacă pentru adresa primită nu a fost negociat încă nici un *SA*, atunci $\mathcal{AN}$ va transmite pachetul α aplicației de negociere.
2. Dacă α are un *SPI* asociat, atunci este căutat *SA*-ul corespunzător în baza de date *IPSec*. Dacă *SPI* nu are corespondent în baza de date, pachetul α este respins.
3. Dacă α nu conține nici un *SPI*, atunci $\mathcal{AN}$ poate deduce că α nu are asociat nici un *SA* și îl respinge.

Definiție

O asociere de securitate atașează parametri de securitate pachetelor de date care sunt trimise în trafic. Un SA descrie parametrii de securitate acceptați de Alice și Bob.

Definiție

O asociere de securitate atașează parametri de securitate pachetelor de date care sunt trimise în trafic. Un SA descrie parametrii de securitate acceptați de Alice și Bob.

În această fază se stabilește o conexiune între o sursă și o destinație, în care cei doi parteneri trebuie să cadă de acord – printre altele – asupra algoritmilor de criptare și autentificare, asupra cheilor de criptare (mărimea, durata, modul de trimitere), valorile de inițializare, precum și alți parametri de securitate.

Definiție

O asociere de securitate atașează parametri de securitate pachetelor de date care sunt trimise în trafic. Un SA descrie parametrii de securitate acceptați de Alice și Bob.

În această fază se stabilește o conexiune între o sursă și o destinație, în care cei doi parteneri trebuie să cadă de acord – printre altele – asupra algoritmilor de criptare și autentificare, asupra cheilor de criptare (mărimea, durata, modul de trimitere), valorile de inițializare, precum și alți parametri de securitate.

După ce s-a definit *SA*-ul unei conexiuni, lui i se atribuie un index (*SPI – Security Parameter Index*) și este stocat în *SPD*.

Definiție

O asociere de securitate atașează parametri de securitate pachetelor de date care sunt trimise în trafic. Un SA descrie parametrii de securitate acceptați de Alice și Bob.

În această fază se stabilește o conexiune între o sursă și o destinație, în care cei doi parteneri trebuie să cadă de acord – printre altele – asupra algoritmilor de criptare și autentificare, asupra cheilor de criptare (mărimea, durata, modul de trimitere), valorile de inițializare, precum și alți parametri de securitate.

După ce s-a definit *SA*-ul unei conexiuni, lui i se atribuie un index (*SPI – Security Parameter Index*) și este stocat în *SPD*.
În interiorul acestei baze de date, la *SA* se adaugă adresele *IP* ale sursei și destinației.

Deci, toată informația dintr-un *SA* este grupată în:

Deci, toată informația dintr-un *SA* este grupată în:

- **Indexul parametrului de securitate** (*SPI*). Scopul lui este de a asocia un *SA* cu o conexiune particulară.

Deci, toată informația dintr-un *SA* este grupată în:

- **Indexul parametrului de securitate** (*SPI*). Scopul lui este de a asocia un *SA* cu o conexiune particulară.
- **Adresa *IP* de destinație**: de obicei adresa unui user sau a unui gateway (router sau firewall).

Deci, toată informația dintr-un *SA* este grupată în:

- **Indexul parametrului de securitate** (*SPI*). Scopul lui este de a asocia un *SA* cu o conexiune particulară.
- **Adresa *IP* de destinație**: de obicei adresa unui user sau a unui gateway (router sau firewall).
- **Protocolul pentru securitatea *ID***: Indică dacă protocolul de securitate este *ESP* sau *AH*.

Deci, toată informația dintr-un *SA* este grupată în:

- **Indexul parametrului de securitate** (*SPI*). Scopul lui este de a asocia un *SA* cu o conexiune particulară.
- **Adresa *IP* de destinație**: de obicei adresa unui user sau a unui gateway (router sau firewall).
- **Protocolul pentru securitatea *ID***: Indică dacă protocolul de securitate este *ESP* sau *AH*.

Asocierea de securitate corespunzătoare unei conexiuni poate fi un *ESP* sau un *AH*, dar nu amândouă.

Deci, toată informația dintr-un *SA* este grupată în:

- **Indexul parametrului de securitate** (*SPI*). Scopul lui este de a asocia un *SA* cu o conexiune particulară.
- **Adresa *IP* de destinație**: de obicei adresa unui user sau a unui gateway (router sau firewall).
- **Protocolul pentru securitatea *ID***: Indică dacă protocolul de securitate este *ESP* sau *AH*.

Asocierea de securitate corespunzătoare unei conexiuni poate fi un *ESP* sau un *AH*, dar nu amândouă.
O comunicare securizată bidirecțională tipică între două entități necesită două asocieri de securitate (câte una pentru fiecare direcție).

Deci, toată informația dintr-un *SA* este grupată în:

- **Indexul parametrului de securitate** (*SPI*). Scopul lui este de a asocia un *SA* cu o conexiune particulară.
- **Adresa *IP* de destinație**: de obicei adresa unui user sau a unui gateway (router sau firewall).
- **Protocolul pentru securitatea *ID***: Indică dacă protocolul de securitate este *ESP* sau *AH*.

Asocierea de securitate corespunzătoare unei conexiuni poate fi un *ESP* sau un *AH*, dar nu amândouă.
O comunicare securizată bidirecțională tipică între două entități necesită două asocieri de securitate (câte una pentru fiecare direcție).
Atât protocolul de securitate *ESP* cât și *AH* suportă două moduri de operare: modul transport și modul tunel.

Antetul de autentificare (*AH*)

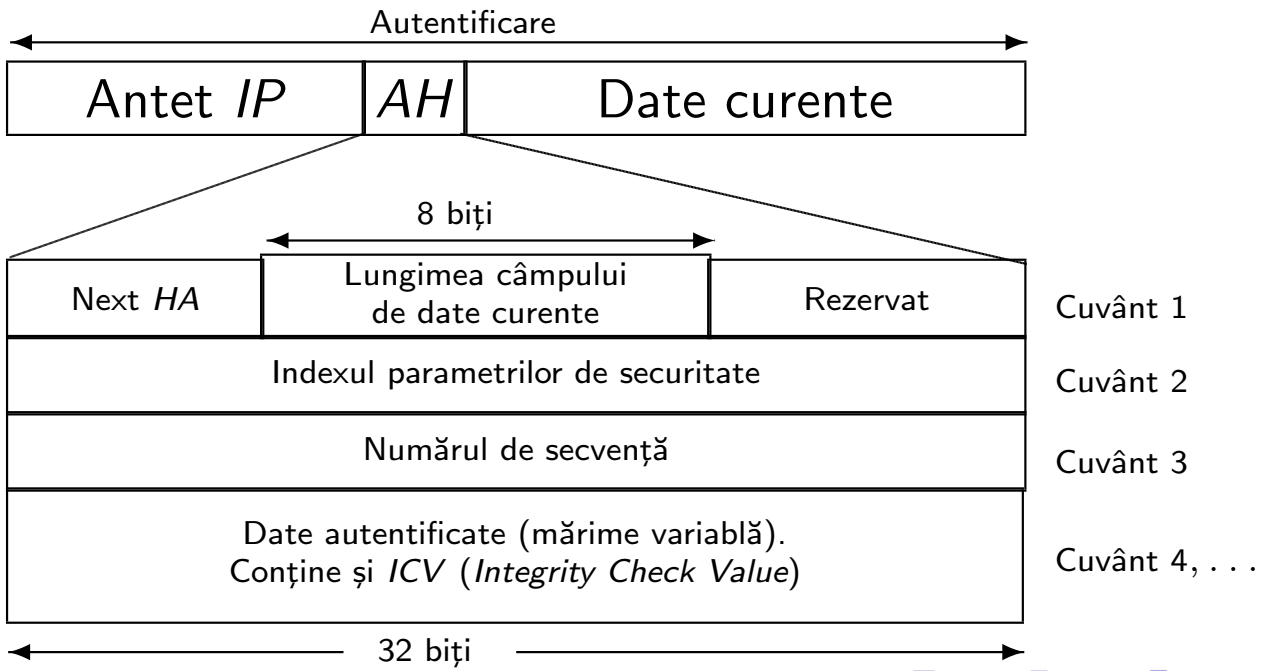
Protocolul *AH* definește formatul pachetelor *IPSec*; el asigură integritatea conexiunii, autentificarea originii datelor și – opțional – un serviciu anti-replay.

Antetul de autentificare (*AH*)

Protocolul *AH* definește formatul pachetelor *IPSec*; el asigură integritatea conexiunii, autentificarea originii datelor și – opțional – un serviciu anti-replay.
AH poate fi folosit singur, combinat cu *ESP*, sau într-o formă internă a modului de utilizare tunel.

Antetul de autentificare (*AH*)

Protocolul *AH* definește formatul pachetelor *IPSec*; el asigură integritatea conexiunii, autentificarea originii datelor și – opțional – un serviciu anti-replay.
AH poate fi folosit singur, combinat cu *ESP*, sau într-o formă internă a modului de utilizare tunel.



- Next *HA*: zonă de 8 biți care identifică tipul antetului pe care îl are următorul câmp de date curente (payload).

- Next *HA*: zonă de 8 biți care identifică tipul antetului pe care îl are următorul câmp de date curente (payload).
- Lungimea câmpului de date curente: Dacă *AH* este format din s cuvinte, conținutul acestui câmp este $s - 2$.

- Next *HA*: zonă de 8 biți care identifică tipul antetului pe care îl are următorul câmp de date curente (payload).
- Lungimea câmpului de date curente: Dacă *AH* este format din s cuvinte, conținutul acestui câmp este $s - 2$.
- Rezervat: Câmp de 16 biți, nefolosit momentan.

- Next *HA*: zonă de 8 biți care identifică tipul antetului pe care îl are următorul câmp de date curente (payload).
- Lungimea câmpului de date curente: Dacă *AH* este format din s cuvinte, conținutul acestui câmp este $s - 2$.
- Rezervat: Câmp de 16 biți, nefolosit momentan.
- Indexul parametrilor de securitate (*SPI*): indică protocoalele de securitate, algoritmii și cheile folosite. Valoarea *SPI* este selectată de (sistemul) destinatar în momentul stabilirii *SA*, din intervalul $[1, 255]$ (codificare asigurată de *IANA – The Internet Assigned Numbers Authority*).

- Next *HA*: zonă de 8 biți care identifică tipul antetului pe care îl are următorul câmp de date curente (payload).
- Lungimea câmpului de date curente: Dacă *AH* este format din s cuvinte, conținutul acestui câmp este $s - 2$.
- Rezervat: Câmp de 16 biți, nefolosit momentan.
- Indexul parametrilor de securitate (*SPI*): indică protocoalele de securitate, algoritmii și cheile folosite. Valoarea *SPI* este selectată de (sistemul) destinatar în momentul stabilirii *SA*, din intervalul $[1, 255]$ (codificare asigurată de *IANA – The Internet Assigned Numbers Authority*).
- Numărul de secvență: Un contor crescător care spune câte pachete au fost trimise și asigură protecție non-reply.

- Next *HA*: zonă de 8 biți care identifică tipul antetului pe care îl are următorul câmp de date curente (payload).
- Lungimea câmpului de date curente: Dacă *AH* este format din s cuvinte, conținutul acestui câmp este $s - 2$.
- Rezervat: Câmp de 16 biți, nefolosit momentan.
- Indexul parametrilor de securitate (*SPI*): indică protocoalele de securitate, algoritmii și cheile folosite. Valoarea *SPI* este selectată de (sistemul) destinatar în momentul stabilirii *SA*, din intervalul $[1, 255]$ (codificare asigurată de *IANA – The Internet Assigned Numbers Authority*).
- Numărul de secvență: Un contor crescător care spune câte pachete au fost trimise și asigură protecție non-reply.
- *ICV*: Valoare de control a integrității pachetului de date transmis. Este multiplu de 32 biți (*IPv4*) sau 64 biți (*IPv6*), variante selectate printr-un câmp suplimentar.

Pentru a calcula *ICV*-ul unui *AH* se ține cont de:

Pentru a calcula *ICV*-ul unui *AH* se ține cont de:

- Câmpurile antetelor *IP*, care sunt sau neschimbate în trafic, sau sunt predictibile ca valoare (pentru un *AH*) în momentul recepției.

Pentru a calcula *ICV*-ul unui *AH* se ține cont de:

- Câmpurile antetelor *IP*, care sunt sau neschimbate în trafic, sau sunt predictibile ca valoare (pentru un *AH*) în momentul recepției.
- Toate datele din antetul *AH*.

Pentru a calcula *ICV*-ul unui *AH* se ține cont de:

- Câmpurile antetelor *IP*, care sunt sau neschimbate în trafic, sau sunt predictibile ca valoare (pentru un *AH*) în momentul recepției.
- Toate datele din antetul *AH*.
- Datele oferite de nivelul de protocol și setul de date curente – despre care se presupune că sunt neschimbate în trafic.

ESP

Protocolul *ESP* (*RFC* 4303) oferă – pe lângă serviciile asigurate de un *AH* – confidențialitatea datelor, și – într-o formă limitată – a traficului pe canal (deci criptarea datelor).

ESP

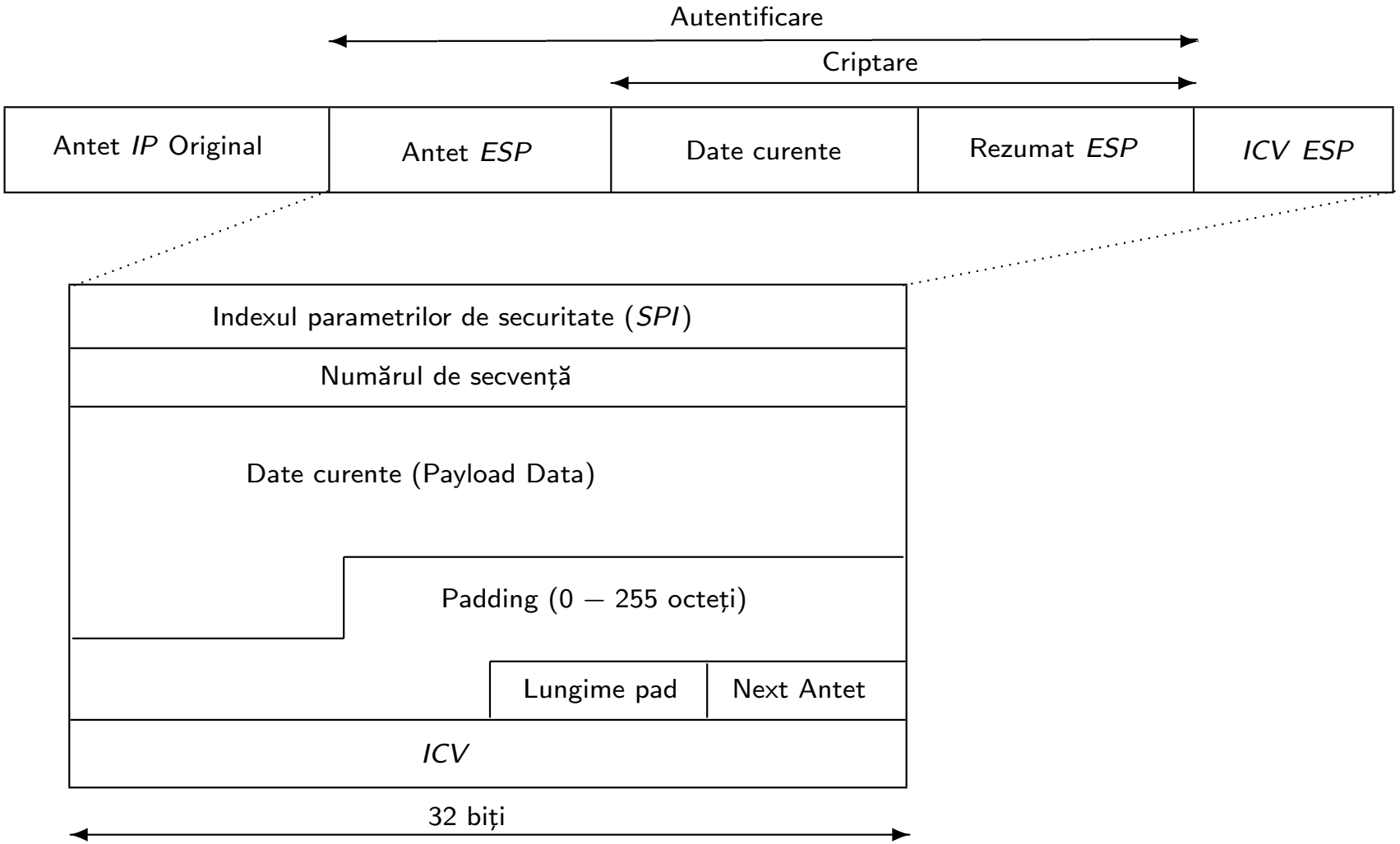
Protocolul *ESP* (*RFC* 4303) oferă – pe lângă serviciile asigurate de un *AH* – confidențialitatea datelor, și – într-o formă limitată – a traficului pe canal (deci criptarea datelor).
Autentificarea originii datelor și corectitudinea informațiilor de conectare sunt servicii auxiliare numite "*integritate*".
Pentru protocol trebuie selectat cel puțin un serviciu de confidențialitate sau integritate.

ESP

Protocolul *ESP* (*RFC* 4303) oferă – pe lângă serviciile asigurate de un *AH* – confidențialitatea datelor, și – într-o formă limitată – a traficului pe canal (deci criptarea datelor).
Autentificarea originii datelor și corectitudinea informațiilor de conectare sunt servicii auxiliare numite "*integritate*".
Pentru protocol trebuie selectat cel puțin un serviciu de confidențialitate sau integritate.
Serviciul non-replay poate fi ales doar împreună cu cel de integritate, iar această alegere este în totalitate la dispoziția destinatarului.

ESP

Protocolul *ESP* (*RFC* 4303) oferă – pe lângă serviciile asigurate de un *AH* – confidențialitatea datelor, și – într-o formă limitată – a traficului pe canal (deci criptarea datelor).
Autentificarea originii datelor și corectitudinea informațiilor de conectare sunt servicii auxiliare numite "*integritate*".
Pentru protocol trebuie selectat cel puțin un serviciu de confidențialitate sau integritate.
Serviciul non-replay poate fi ales doar împreună cu cel de integritate, iar această alegere este în totalitate la dispoziția destinatarului.
Confidențialitatea fluxului din trafic necesită alegerea modului tunel.



ESP asigură criptarea unui *IP* la nivel de pachet, folosind algoritmi de criptare simetrici (recomandat *AES*).

Descrierea câmpurilor dintr-o încapsulare *ESP*

- **SPI** și Numărul de secvență: Identice cu formatul *AH*.

Descrierea câmpurilor dintr-o încapsulare *ESP*

- **SPI** și Numărul de secvență: Identice cu formatul *AH*.
- Date curente (Payload data): Câmp de lungime variabilă care conține datele descrise de câmpul *Next Antet*.

Descrierea câmpurilor dintr-o încapsulare *ESP*

- **SPI** și Numărul de secvență: Identice cu formatul *AH*.
- Date curente (Payload data): Câmp de lungime variabilă care conține datele descrise de câmpul *Next Antet*.
  ESP folosește algoritmi de criptare simetrici și – deoarece pachetele *IP* pot veni într-o ordine arbitrară – fiecare pachet va conține și informații de sincronizare criptografică.

Descrierea câmpurilor dintr-o încapsulare *ESP*

- **SPI** și Numărul de secvență: Identice cu formatul *AH*.
- Date curente (Payload data): Câmp de lungime variabilă care conține datele descrise de câmpul *Next Antet*.
  ESP folosește algoritmi de criptare simetrici și – deoarece pachetele *IP* pot veni într-o ordine arbitrară – fiecare pachet va conține și informații de sincronizare criptografică.
- Zona anexă (Padding) – de maxim 255 octeți.

Descrierea câmpurilor dintr-o încapsulare *ESP*

- **SPI** și Numărul de secvență: Identice cu formatul *AH*.
- Date curente (Payload data): Câmp de lungime variabilă care conține datele descrise de câmpul *Next Antet*.
  ESP folosește algoritmi de criptare simetrici și – deoarece pachetele *IP* pot veni într-o ordine arbitrară – fiecare pachet va conține și informații de sincronizare criptografică.
- Zona anexă (Padding) – de maxim 255 octeți.Necesară deoarece:
  - Unii algoritmi de criptare folosesc blocuri de text clar de lungime fixată. Zona anexă este folosită pentru a completa textul clar până la lungimea solicitată.

Descrierea câmpurilor dintr-o încapsulare *ESP*

- **SPI** și Numărul de secvență: Identice cu formatul *AH*.
- Date curente (Payload data): Câmp de lungime variabilă care conține datele descrise de câmpul *Next Antet*.
  ESP folosește algoritmi de criptare simetrici și – deoarece pachetele *IP* pot veni într-o ordine arbitrară – fiecare pachet va conține și informații de sincronizare criptografică.
- Zona anexă (Padding) – de maxim 255 octeți.Necesară deoarece:
  - Unii algoritmi de criptare folosesc blocuri de text clar de lungime fixată. Zona anexă este folosită pentru a completa textul clar până la lungimea solicitată.
  - Anexa este adăugată la textul criptat, pentru ca în final numărul de biți al textului criptat să fie multiplu de 32.

- Lungime pad: Indică numărul de octeți din zona anexă. Valoarea este un număr din intervalul $[0, 255]$, unde 0 înseamnă că zona anexă nu este prezentă.

- Lungime pad: Indică numărul de octeți din zona anexă. Valoarea este un număr din intervalul $[0, 255]$, unde 0 înseamnă că zona anexă nu este prezentă.
- Next antet: Câmp de un octet care identifică tipul de date din zona datelor curente.

- Lungime pad: Indică numărul de octeți din zona anexă. Valoarea este un număr din intervalul $[0, 255]$, unde 0 înseamnă că zona anexă nu este prezentă.
- Next antet: Câmp de un octet care identifică tipul de date din zona datelor curente.Codificarea folosită este indicată pe pagina *IANA*. De exemplu, valoarea 4 indică *IPv4*, valoarea 41 indică *IPv6* etc.

- Lungime pad: Indică numărul de octeți din zona anexă. Valoarea este un număr din intervalul $[0, 255]$, unde 0 înseamnă că zona anexă nu este prezentă.
- Next antet: Câmp de un octet care identifică tipul de date din zona datelor curente.Codificarea folosită este indicată pe pagina *IANA*. De exemplu, valoarea 4 indică *IPv4*, valoarea 41 indică *IPv6* etc.
  De asemenea, se poate identifica un protocol de nivel superior; de exemplu, valoarea 6 indică protocolul *TCP*.

- Lungime pad: Indică numărul de octeți din zona anexă. Valoarea este un număr din intervalul $[0, 255]$, unde 0 înseamnă că zona anexă nu este prezentă.
- Next antet: Câmp de un octet care identifică tipul de date din zona datelor curente.Codificarea folosită este indicată pe pagina *IANA*. De exemplu, valoarea 4 indică *IPv4*, valoarea 41 indică *IPv6* etc.
  De asemenea, se poate identifica un protocol de nivel superior; de exemplu, valoarea 6 indică protocolul *TCP*.
- *ICV*: Valoarea de control a integrității este identică cu cea de la *AH*, cu singura specificație că aici ea este calculată pentru 3 câmpuri: *Antet ESP*, *Rezumat ESP* și *Date curente*.

- Lungime pad: Indică numărul de octeți din zona anexă. Valoarea este un număr din intervalul $[0, 255]$, unde 0 înseamnă că zona anexă nu este prezentă.
- Next antet: Câmp de un octet care identifică tipul de date din zona datelor curente.Codificarea folosită este indicată pe pagina *IANA*. De exemplu, valoarea 4 indică *IPv4*, valoarea 41 indică *IPv6* etc.
  De asemenea, se poate identifica un protocol de nivel superior; de exemplu, valoarea 6 indică protocolul *TCP*.
- *ICV*: Valoarea de control a integrității este identică cu cea de la *AH*, cu singura specificație că aici ea este calculată pentru 3 câmpuri: *Antet ESP*, *Rezumat ESP* și *Date curente*.
  Dacă s-a selectat serviciul de criptare, ultimele două câmpuri sunt sub formă criptată (criptarea fiind aplicată înainte de autentificare).

Modurile de operare *AH* și *ESP*

IPSec poate fi implementat pe două tipuri de echipament: pe server gazdă sau pe poartă (gateway) de securitate.

Modurile de operare *AH* și *ESP*

IPSec poate fi implementat pe două tipuri de echipament: pe server gazdă sau pe poartă (gateway) de securitate.

Definiție

Un "gateway de securitate" este un sistem intermediar între două rețele. Canalul de intrare într-un gateway este nesigur, iar canalul de ieșire este securizat.

Modurile de operare *AH* și *ESP*

IPSec poate fi implementat pe două tipuri de echipament: pe server gazdă sau pe poartă (gateway) de securitate.

Definiție

Un "gateway de securitate" este un sistem intermediar între două rețele. Canalul de intrare într-un gateway este nesigur, iar canalul de ieșire este securizat.

Un gateway implementează *IPSec* pe interfața nesigură, pentru a permite comunicații securizate între servere gazdă de pe ambele laturi.

Modurile de operare *AH* și *ESP*

IPSec poate fi implementat pe două tipuri de echipament: pe server gazdă sau pe poartă (gateway) de securitate.

Definiție

Un "gateway de securitate" este un sistem intermediar între două rețele. Canalul de intrare într-un gateway este nesigur, iar canalul de ieșire este securizat.

Un gateway implementează *IPSec* pe interfața nesigură, pentru a permite comunicații securizate între servere gazdă de pe ambele laturi.

Astfel, serviciile de securitate pot fi asigurate:

1. între orice două servere care comunică între ele;

Modurile de operare *AH* și *ESP*

IPSec poate fi implementat pe două tipuri de echipament: pe server gazdă sau pe poartă (gateway) de securitate.

Definiție

Un "gateway de securitate" este un sistem intermediar între două rețele. Canalul de intrare într-un gateway este nesigur, iar canalul de ieșire este securizat.

Un gateway implementează *IPSec* pe interfața nesigură, pentru a permite comunicații securizate între servere gazdă de pe ambele laturi.

Astfel, serviciile de securitate pot fi asigurate:

1. între orice două servere care comunică între ele;
2. între o pereche de gateway de securitate;

Modurile de operare *AH* și *ESP*

IPSec poate fi implementat pe două tipuri de echipament: pe server gazdă sau pe poartă (gateway) de securitate.

Definiție

Un "gateway de securitate" este un sistem intermediar între două rețele. Canalul de intrare într-un gateway este nesigur, iar canalul de ieșire este securizat.

Un gateway implementează *IPSec* pe interfața nesigură, pentru a permite comunicații securizate între servere gazdă de pe ambele laturi.

Astfel, serviciile de securitate pot fi asigurate:

1. între orice două servere care comunică între ele;
2. între o pereche de gateway de securitate;
3. între un gateway și un server.

AH și *ESP* suportă două moduri de operare: modul transport și modul tunel.

AH și *ESP* suportă două moduri de operare: modul transport și modul tunel.

- Un *SA* în **mod transport** este o asociere de securitate între două servere.
  Când un gateway de securitate lucrează în mod transport, el acționează ca un server: traficul îi este destinat lui.

AH și *ESP* suportă două moduri de operare: modul transport și modul tunel.

- Un *SA* în **mod transport** este o asociere de securitate între două servere.
  Când un gateway de securitate lucrează în mod transport, el acționează ca un server: traficul îi este destinat lui.
- Un *SA* în **mod tunel** este o asociere de securitate în care cel puțin unul din parteneri este un gateway.

AH și *ESP* suportă două moduri de operare: modul transport și modul tunel.

- Un *SA* în **mod transport** este o asociere de securitate între două servere.
  Când un gateway de securitate lucrează în mod transport, el acționează ca un server: traficul îi este destinat lui.
- Un *SA* în **mod tunel** este o asociere de securitate în care cel puțin unul din parteneri este un gateway.

Modul transport este folosit pentru protejarea protocoalelor de comunicare, pe când modul tunel are ca principal scop protejarea pachetelor *IP* (tot pachetul *IP* este încapsulat în alt pachet *IP* și este adăugat un nou antet *IP* între antetele interne și externe).

1. În modul transport, antetul protocolului de securitate apare imediat după antetul *IP* originar și înaintea datelor curente.

1. În modul transport, antetul protocolului de securitate apare imediat după antetul *IP* originar și înaintea datelor curente. Pentru *ESP* implementat în modul transport, *SA* asigură servicii de securitate numai pentru protocoale de nivel înalt, nu și pentru antetul *IP* originar.

1. În modul transport, antetul protocolului de securitate apare imediat după antetul *IP* originar și înaintea datelor curente. Pentru *ESP* implementat în modul transport, *SA* asigură servicii de securitate numai pentru protocoale de nivel înalt, nu și pentru antetul *IP* originar.
   Pentru *AH*, protecția este extinsă la antetul *IP* originar.

1. În modul transport, antetul protocolului de securitate apare imediat după antetul *IP* originar și înaintea datelor curente. Pentru *ESP* implementat în modul transport, *SA* asigură servicii de securitate numai pentru protocoale de nivel înalt, nu și pentru antetul *IP* originar.
   Pentru *AH*, protecția este extinsă la antetul *IP* originar.
2. Pentru *SA* în modul tunel, apar încă două antete:

1. În modul transport, antetul protocolului de securitate apare imediat după antetul *IP* originar și înaintea datelor curente. Pentru *ESP* implementat în modul transport, *SA* asigură servicii de securitate numai pentru protocoale de nivel înalt, nu și pentru antetul *IP* originar.
   Pentru *AH*, protecția este extinsă la antetul *IP* originar.
2. Pentru *SA* în modul tunel, apar încă două antete:
   - unul extern – care specifică destinația finală *IPSec* și detalii legate de această destinație;

1. În modul transport, antetul protocolului de securitate apare imediat după antetul *IP* originar și înaintea datelor curente.
   Pentru *ESP* implementat în modul transport, *SA* asigură servicii de securitate numai pentru protocoale de nivel înalt, nu și pentru antetul *IP* originar.
   Pentru *AH*, protecția este extinsă la antetul *IP* originar.
2. Pentru *SA* în modul tunel, apar încă două antete:
   - unul extern – care specifică destinația finală *IPSec* și detalii legate de această destinație;
   - un antet intern care specifică ultima destinație (aparentă) a pachetului.
   Antetul protocolului de securitate apare după antetul *IP* exterior și înaintea celui interior.

1. În modul transport, antetul protocolului de securitate apare imediat după antetul *IP* originar și înaintea datelor curente.
   Pentru *ESP* implementat în modul transport, *SA* asigură servicii de securitate numai pentru protocoale de nivel înalt, nu și pentru antetul *IP* originar.
   Pentru *AH*, protecția este extinsă la antetul *IP* originar.
2. Pentru *SA* în modul tunel, apar încă două antete:
   - unul extern – care specifică destinația finală *IPSec* și detalii legate de această destinație;
   - un antet intern care specifică ultima destinație (aparentă) a pachetului.
   Antetul protocolului de securitate apare după antetul *IP* exterior și înaintea celui interior.
   Dacă se folosește *ESP*, protecția este asigurată numai pentru pachetele interne, nu și pentru antetul exterior.

Deoarece serviciile de securitate *IP* folosesc sisteme simetrice de criptare, este necesar ca ambele părți (*Alice* și *Bob*) să cadă de acord asupra unor mecanisme care să le permită să stabilească chei secrete pentru criptare, autentificare și integritate. *IPSec* permite o distribuție a cheilor atât manual cât și automat.

Deoarece serviciile de securitate *IP* folosesc sisteme simetrice de criptare, este necesar ca ambele părți (*Alice* și *Bob*) să cadă de acord asupra unor mecanisme care să le permită să stabilească chei secrete pentru criptare, autentificare și integritate. *IPSec* permite o distribuție a cheilor atât manual cât și automat.

Definiție

Protocolul IKE (Internet Key Exchange) stabilește proceduri și formate de pachete de date care stau la baza definirii, negocierii, modificării și eliminării asocierilor de securitate (SA).

Deoarece serviciile de securitate *IP* folosesc sisteme simetrice de criptare, este necesar ca ambele părți (*Alice* și *Bob*) să cadă de acord asupra unor mecanisme care să le permită să stabilească chei secrete pentru criptare, autentificare și integritate. *IPSec* permite o distribuție a cheilor atât manual cât și automat.

Definiție

Protocolul IKE (Internet Key Exchange) stabilește proceduri și formate de pachete de date care stau la baza definirii, negocierii, modificării și eliminării asocierilor de securitate (SA).El asigură cadrul de securitate pentru transferul de chei și autentificarea datelor, independent de mecanismul de generare al cheilor, algoritmii de criptare sau mecanismele de autentificare.

Deoarece serviciile de securitate *IP* folosesc sisteme simetrice de criptare, este necesar ca ambele părți (*Alice* și *Bob*) să cadă de acord asupra unor mecanisme care să le permită să stabilească chei secrete pentru criptare, autentificare și integritate. *IPSec* permite o distribuție a cheilor atât manual cât și automat.

Definiție

Protocolul IKE (Internet Key Exchange) stabilește proceduri și formate de pachete de date care stau la baza definirii, negocierii, modificării și eliminării asocierilor de securitate (SA).El asigură cadrul de securitate pentru transferul de chei și autentificarea datelor, independent de mecanismul de generare al cheilor, algoritmii de criptare sau mecanismele de autentificare.

Protocolul *IKE* este utilizat în două versiuni free: *IKE v1* și *IKE v2* (prima variantă în decembrie 2005 cu *RFC* 4306, iar forma completă în septembrie 2010, prin *RFC* 5996).

Parametrii unei asocieri de securitate

- Tipul de protecție folosit (*ESP* sau *AH*).

Parametrii unei asocieri de securitate

- Tipul de protecție folosit (*ESP* sau *AH*).
- Algoritmul de autentificare folosit cu *AH*.

Parametrii unei asocieri de securitate

- Tipul de protecție folosit (*ESP* sau *AH*).
- Algoritmul de autentificare folosit cu *AH*.
- Cheile folosite cu algoritmul de autentificare din *AH*.

Parametrii unei asocieri de securitate

- Tipul de protecție folosit (*ESP* sau *AH*).
- Algoritmul de autentificare folosit cu *AH*.
- Cheile folosite cu algoritmul de autentificare din *AH*.
- Algoritmul de criptare și modul, utilizate cu *ESP*.

Parametrii unei asocieri de securitate

- Tipul de protecție folosit (*ESP* sau *AH*).
- Algoritmul de autentificare folosit cu *AH*.
- Cheile folosite cu algoritmul de autentificare din *AH*.
- Algoritmul de criptare și modul, utilizate cu *ESP*.
- Cheia utilizată de algoritmul de criptare în *ESP*.

Parametrii unei asocieri de securitate

- Tipul de protecție folosit (*ESP* sau *AH*).
- Algoritmul de autentificare folosit cu *AH*.
- Cheile folosite cu algoritmul de autentificare din *AH*.
- Algoritmul de criptare și modul, utilizate cu *ESP*.
- Cheia utilizată de algoritmul de criptare în *ESP*.
- Vectorul de inițializare pentru algoritmul de criptare utilizat în *ESP*.

Parametrii unei asocieri de securitate

- Tipul de protecție folosit (*ESP* sau *AH*).
- Algoritmul de autentificare folosit cu *AH*.
- Cheile folosite cu algoritmul de autentificare din *AH*.
- Algoritmul de criptare și modul, utilizate cu *ESP*.
- Cheia utilizată de algoritmul de criptare în *ESP*.
- Vectorul de inițializare pentru algoritmul de criptare utilizat în *ESP*.
- Algoritmul de autentificare și modul, utilizate cu *ESP*.

Parametrii unei asocieri de securitate

- Tipul de protecție folosit (*ESP* sau *AH*).
- Algoritmul de autentificare folosit cu *AH*.
- Cheile folosite cu algoritmul de autentificare din *AH*.
- Algoritmul de criptare și modul, utilizate cu *ESP*.
- Cheia utilizată de algoritmul de criptare în *ESP*.
- Vectorul de inițializare pentru algoritmul de criptare utilizat în *ESP*.
- Algoritmul de autentificare și modul, utilizate cu *ESP*.
- Cheia de autentificare utilizată cu algoritmul de autentificare în *ESP*.

Parametrii unei asocieri de securitate

- Durata de valabilitate a cheii utilizate sau data când ea trebuie schimbată.

Parametrii unei asocieri de securitate

- Durata de valabilitate a cheii utilizate sau data când ea trebuie schimbată.
- Algoritmii de dispersie pentru crearea amprentei care va fi semnată.

Parametrii unei asocieri de securitate

- Durata de valabilitate a cheii utilizate sau data când ea trebuie schimbată.
- Algoritmii de dispersie pentru crearea amprentei care va fi semnată.
- Informații despre grupul peste care se va defini schimbul de chei Diffie - Hellman.

Parametrii unei asocieri de securitate

- Durata de valabilitate a cheii utilizate sau data când ea trebuie schimbată.
- Algoritmii de dispersie pentru crearea amprentei care va fi semnată.
- Informații despre grupul peste care se va defini schimbul de chei Diffie - Hellman.
- Perioada de valabilitate a asocierii de securitate curente.

Parametrii unei asocieri de securitate

- Durata de valabilitate a cheii utilizate sau data când ea trebuie schimbată.
- Algoritmii de dispersie pentru crearea amprentei care va fi semnată.
- Informații despre grupul peste care se va defini schimbul de chei Diffie - Hellman.
- Perioada de valabilitate a asocierii de securitate curente.
- Adresa sursei care a construit asocierea de securitate.

Selectarea algoritmilor

Alice negociază cu *Bob* algoritmii pentru construirea asocierii de securitate *IPSec*).

Selectarea algoritmilor

Alice negociază cu *Bob* algoritmii pentru construirea asocierii de securitate *IPSec*).

- Algoritmii de criptare care protejează datele:
  - $3DES$ (obligatoriu)
  - $AES - CBC - 128$ (recomandat)
  - $AES - CTR - 128$ (recomandat)

Selectarea algoritmilor

Alice negociază cu *Bob* algoritmii pentru construirea asocierii de securitate *IPSec*).

- Algoritmii de criptare care protejează datele:
  - $3DES$ (obligatoriu)
  - $AES - CBC - 128$ (recomandat)
  - $AES - CTR - 128$ (recomandat)
- Algoritmi de protecție a integrității datelor, care produc amprente:
  - $HMAC - SHA1 - 96$ (obligatoriu)
  - $AES - XCBC - 96$ (recomandat)
  - $HMAC - MD5 - 96$ (opțional)

- Informații despre grupul peste care operează protocolul Diffie - Hellman:
  - Grupul 2 – unde se calculează logaritmi discreți pe 1024 biți (obligatoriu)
  - Grupul 14 – unde se calculează logaritmi discreți pe 2048 biți (recomandat)
  - Curbe eliptice peste $GF(2^{155})$ sau $GF(2^{185})$ (opțional)

- Informații despre grupul peste care operează protocolul Diffie - Hellman:
  - Grupul 2 – unde se calculează logaritmi discreți pe 1024 biți (obligatoriu)
  - Grupul 14 – unde se calculează logaritmi discreți pe 2048 biți (recomandat)
  - Curbe eliptice peste $GF(2^{155})$ sau $GF(2^{185})$ (opțional)
- Generatori de numere pseudoaleatoare:
  - *PRF* – *HMAC* – *SHA*1 (conform standardului *RFC* 2104) (obligatoriu)
  - *PRF* – *AES* – *XCBC* – *PRF* – 128 (conform standardului *RFC* 3664) (recomandat)
  - *PRF* – *HMAC* – *MD*5 (conform standardului *RFC* 2104) (opțional)

Alice (care inițiază comunicarea) propune un set de algoritmi pe care poate să îi suporte sistemul său; *Bob* va selecta din ei algoritmii pe care îi agrează, creînd astfel un $IKE - SA$ (asocierea de securitate corespunzătoare *IKE*).

Alice (care inițiază comunicarea) propune un set de algoritmi pe care poate să îi suporte sistemul său; *Bob* va selecta din ei algoritmii pe care îi agrează, creînd astfel un $IKE - SA$ (asocierea de securitate corespunzătoare *IKE*).

În general, două entități (de exemplu două servere *IPSec*) pot negocia mai multe *SA*-uri.

Alice (care inițiază comunicarea) propune un set de algoritmi pe care poate să îi suporte sistemul său; *Bob* va selecta din ei algoritmii pe care îi agrează, creînd astfel un $IKE - SA$ (asocierea de securitate corespunzătoare *IKE*).

În general, două entități (de exemplu două servere *IPSec*) pot negocia mai multe *SA*-uri.

Comunicările *IKE* constau din perechi de mesaje: o cerere urmată de un răspuns.
Primul schimb de mesaje constă totdeauna din două cereri/răspuns: $IKE - SA - INIT$ și $IKE - AUTH$.

IKE-SA-INIT

Inițiatorul *Alice* și destinatarul *Bob* negociază utilizarea algoritmilor de criptare (definind un $IKE - SA$) și schimbă informațiile necesare stabilirii unui schimb de chei.

IKE-SA-INIT

Inițiatorul *Alice* și destinatarul *Bob* negociază utilizarea algoritmilor de criptare (definind un $IKE - SA$) și schimbă informațiile necesare stabilirii unui schimb de chei.

- La primul pas (Pas 1), *Alice* trimite un mesaj de forma

$$HDR \| SA_A 1 \| KE_A \| N_A$$

IKE-SA-INIT

Inițiatorul *Alice* și destinatarul *Bob* negociază utilizarea algoritmilor de criptare (definind un $IKE - SA$) și schimbă informațiile necesare stabilirii unui schimb de chei.

- La primul pas (Pas 1), *Alice* trimite un mesaj de forma

$$HDR \| SA_A 1 \| KE_A \| N_A$$

  - *HDR* – Zonă care conține *SPI*, numărul versiunii *IKE*, și alți identificatori (cum ar fi datele curente pentru KE_A și $SA_A 1$).

IKE-SA-INIT

Inițiatorul *Alice* și destinatarul *Bob* negociază utilizarea algoritmilor de criptare (definind un $IKE - SA$) și schimbă informațiile necesare stabilirii unui schimb de chei.

- La primul pas (Pas 1), *Alice* trimite un mesaj de forma

$$HDR \| SA_A1 \| KE_A \| N_A$$

  - *HDR* – Zonă care conține *SPI*, numărul versiunii *IKE*, și alți identificatori (cum ar fi datele curente pentru KE_A și SA_A1).
  - SA_A1 – Informația propusă de *Alice* pentru crearea $IKE - SA$: generatorul de numere pseudo-aleatoare și algoritmii criptografici agreați, precum și grupul de bază pentru protocolul Diffie - Hellman.

IKE-SA-INIT

Inițiatorul *Alice* și destinatarul *Bob* negociază utilizarea algoritmilor de criptare (definind un $IKE - SA$) și schimbă informațiile necesare stabilirii unui schimb de chei.

- La primul pas (Pas 1), *Alice* trimite un mesaj de forma

$$HDR \| SA_A1 \| KE_A \| N_A$$

  - *HDR* – Zonă care conține *SPI*, numărul versiunii *IKE*, și alți identificatori (cum ar fi datele curente pentru KE_A și SA_A1).
  - SA_A1 – Informația propusă de *Alice* pentru crearea $IKE - SA$: generatorul de numere pseudo-aleatoare și algoritmii criptografici agreați, precum și grupul de bază pentru protocolul Diffie - Hellman.
  - KE_A – Cheia Diffie - Hellman g^a a lui *Alice*.

IKE-SA-INIT

Inițiatorul *Alice* și destinatarul *Bob* negociază utilizarea algoritmilor de criptare (definind un $IKE - SA$) și schimbă informațiile necesare stabilirii unui schimb de chei.

- La primul pas (Pas 1), *Alice* trimite un mesaj de forma

$$HDR \| SA_A1 \| KE_A \| N_A$$

  - *HDR* – Zonă care conține *SPI*, numărul versiunii *IKE*, și alți identificatori (cum ar fi datele curente pentru KE_A și SA_A1).
  - SA_A1 – Informația propusă de *Alice* pentru crearea $IKE - SA$: generatorul de numere pseudo-aleatoare și algoritmii criptografici agreați, precum și grupul de bază pentru protocolul Diffie - Hellman.
  - KE_A – Cheia Diffie - Hellman g^a a lui *Alice*.
  - N_A – un nonce generat de *Alice* (contra unui atac reply).

- *Bob* răspunde (Pas 2) cu mesajul

$$HDR\|SA_B1\|KE_B\|N_B\|[CERTREQ]$$

- *Bob* răspunde (Pas 2) cu mesajul

$$HDR\|SA_B1\|KE_B\|N_B\|[CERTREQ]$$

unde

- Conținutul HRD, SA_B1, KE_B sunt similare cu informațiile analoge ale lui *Alice*. Șirul algoritmilor propuși de *Bob* sunt o selecție din intersecția listei trimise de *Alice* și proprii săi algoritmi agreați.

- *Bob* răspunde (Pas 2) cu mesajul

$$HDR\|SA_B1\|KE_B\|N_B\|[CERTREQ]$$

unde

- Conținutul *HRD*, SA_B1, KE_B sunt similare cu informațiile analoge ale lui *Alice*. Șirul algoritmilor propuși de *Bob* sunt o selecție din intersecția listei trimise de *Alice* și proprii săi algoritmi agreați.
  Cheia Diffie - Hellman g^b și nonce-ul N_B completează informația curentă.

- *Bob* răspunde (Pas 2) cu mesajul

$$HDR\|SA_B1\|KE_B\|N_B\|[CERTREQ]$$

unde

  - Conținutul *HRD*, SA_B1, KE_B sunt similare cu informațiile analoge ale lui *Alice*. Șirul algoritmilor propuși de *Bob* sunt o selecție din intersecția listei trimise de *Alice* și proprii săi algoritmi agreați.
    Cheia Diffie - Hellman g^b și nonce-ul N_B completează informația curentă.
  - Opțional, *Bob* poate solicita un anumit tip de certificat (de exemplu *X*.509), trimițând această cerere în *CERTREQ*.

După acest prim schimb de mesaje, partenerii au stabilit un $IKE - SA$ conținut în SA_B1, neautentificat încă.

După acest prim schimb de mesaje, partenerii au stabilit un $IKE - SA$ conținut în SA_B1, neautentificat încă.
Similar, după schimbul de chei Diffie-Hellman, ei au generat o cheie de sesiune neautentificată *SKEYSEED* din care vor deriva ulterior toate cheile necesare pentru $IKE - SA$:

După acest prim schimb de mesaje, partenerii au stabilit un $IKE - SA$ conținut în SA_B1, neautentificat încă.
Similar, după schimbul de chei Diffie-Hellman, ei au generat o cheie de sesiune neautentificată *SKEYSEED* din care vor deriva ulterior toate cheile necesare pentru $IKE - SA$:
SK_e – cheia de criptare (câte una pentru fiecare direcție);

După acest prim schimb de mesaje, partenerii au stabilit un $IKE - SA$ conținut în SA_B1, neautentificat încă.
Similar, după schimbul de chei Diffie-Hellman, ei au generat o cheie de sesiune neautentificată *SKEYSEED* din care vor deriva ulterior toate cheile necesare pentru $IKE - SA$:
SK_e – cheia de criptare (câte una pentru fiecare direcție);
SK_a – cheia pentru autentificarea și verificarea integrității mesajului (câte una pentru fiecare direcție);

După acest prim schimb de mesaje, partenerii au stabilit un $IKE - SA$ conținut în SA_B1, neautentificat încă.
Similar, după schimbul de chei Diffie-Hellman, ei au generat o cheie de sesiune neautentificată *SKEYSEED* din care vor deriva ulterior toate cheile necesare pentru $IKE - SA$:
SK_e – cheia de criptare (câte una pentru fiecare direcție);
SK_a – cheia pentru autentificarea și verificarea integrității mesajului (câte una pentru fiecare direcție);
SK_d – cheia pentru derivarea cheilor necesare asocierilor de securitate derivate;

După acest prim schimb de mesaje, partenerii au stabilit un $IKE - SA$ conținut în SA_B1, neautentificat încă.
Similar, după schimbul de chei Diffie-Hellman, ei au generat o cheie de sesiune neautentificată *SKEYSEED* din care vor deriva ulterior toate cheile necesare pentru $IKE - SA$:
SK_e – cheia de criptare (câte una pentru fiecare direcție);
SK_a – cheia pentru autentificarea și verificarea integrității mesajului (câte una pentru fiecare direcție);
SK_d – cheia pentru derivarea cheilor necesare asocierilor de securitate derivate;
SK_p – cheia pentru crearea datelor curente din schimbul de mesaje următor.

După acest prim schimb de mesaje, partenerii au stabilit un $IKE - SA$ conținut în SA_B1, neautentificat încă.
Similar, după schimbul de chei Diffie-Hellman, ei au generat o cheie de sesiune neautentificată *SKEYSEED* din care vor deriva ulterior toate cheile necesare pentru $IKE - SA$:
SK_e – cheia de criptare (câte una pentru fiecare direcție);
SK_a – cheia pentru autentificarea și verificarea integrității mesajului (câte una pentru fiecare direcție);
SK_d – cheia pentru derivarea cheilor necesare asocierilor de securitate derivate;
SK_p – cheia pentru crearea datelor curente din schimbul de mesaje următor.

În acest moment *Alice* și *Bob* au stabilit algoritmii care vor fi utilizați în comunicările ulterioare, dar încă nu s-au autentificat reciproc.

IKE-SA-AUTH

Alice și *Bob* se autentifică reciproc folosind diverse mecanisme de autentificare (semnături digitale, certificate, *EAP* – Extensible Authentication Protocol, chei partajate).

IKE-SA-AUTH

Alice și *Bob* se autentifică reciproc folosind diverse mecanisme de autentificare (semnături digitale, certificate, *EAP* – Extensible Authentication Protocol, chei partajate). Acum se crează primul $IKE - SA$ și asocierea sa de securitate *IPSec*; ele se numesc generic "$child - SA$".

IKE-SA-AUTH

Alice și *Bob* se autentifică reciproc folosind diverse mecanisme de autentificare (semnături digitale, certificate, *EAP* – Extensible Authentication Protocol, chei partajate). Acum se crează primul $IKE - SA$ și asocierea sa de securitate *IPSec*; ele se numesc generic "$child - SA$".

- *Alice* trimite (Pas 3) lui *Bob*

$$HDR \| SK\{ID_A, [CERT], [CERTREQ], [ID_B], AUTH, SA_A2, TS_A, TS_B\}$$

- *HDR* – antetul; include *SPI*-urile celor doi parteneri, numărul versiunii *IKE*, identificatorii de mesaje folosiți în $IKE - SA - INIT$.

- *HDR* – antetul; include *SPI*-urile celor doi parteneri, numărul versiunii *IKE*, identificatorii de mesaje folosiți în $IKE - SA - INIT$.
- $SK\{\dots\}$ – se asigură criptarea și integritatea datelor folosind SK_e și SK_a.

- *HDR* – antetul; include *SPI*-urile celor doi parteneri, numărul versiunii *IKE*, identificatorii de mesaje folosiți în $IKE - SA - INIT$.
- $SK\{\dots\}$ – se asigură criptarea și integritatea datelor folosind SK_e și SK_a.
- *CERT* – un certificat al lui *Alice*, eliberat de o entitate *PKI*.

- *HDR* – antetul; include *SPI*-urile celor doi parteneri, numărul versiunii *IKE*, identificatorii de mesaje folosiți în $IKE - SA - INIT$.
- $SK\{\dots\}$ – se asigură criptarea și integritatea datelor folosind SK_e și SK_a.
- *CERT* – un certificat al lui *Alice*, eliberat de o entitate *PKI*.
- *CERTREQ* – O listă de autorități de certificare de încredere.

- *HDR* – antetul; include *SPI*-urile celor doi parteneri, numărul versiunii *IKE*, identificatorii de mesaje folosiți în $IKE - SA - INIT$.
- $SK\{\dots\}$ – se asigură criptarea și integritatea datelor folosind SK_e și SK_a.
- *CERT* – un certificat al lui *Alice*, eliberat de o entitate *PKI*.
- *CERTREQ* – O listă de autorități de certificare de încredere.
- ID_A, ID_B – identificatorii celor doi parteneri (*Alice*, *Bob*).

- *HDR* – antetul; include *SPI*-urile celor doi parteneri, numărul versiunii *IKE*, identificatorii de mesaje folosiți în $IKE - SA - INIT$.
- $SK\{\dots\}$ – se asigură criptarea și integritatea datelor folosind SK_e și SK_a.
- *CERT* – un certificat al lui *Alice*, eliberat de o entitate *PKI*.
- *CERTREQ* – O listă de autorități de certificare de încredere.
- ID_A, ID_B – identificatorii celor doi parteneri (*Alice*, *Bob*).
- *AUTH* – autentificarea (un mesaj semnat).

- *HDR* – antetul; include *SPI*-urile celor doi parteneri, numărul versiunii *IKE*, identificatorii de mesaje folosiți în $IKE - SA - INIT$.
- $SK\{\dots\}$ – se asigură criptarea și integritatea datelor folosind SK_e și SK_a.
- *CERT* – un certificat al lui *Alice*, eliberat de o entitate *PKI*.
- *CERTREQ* – O listă de autorități de certificare de încredere.
- ID_A, ID_B – identificatorii celor doi parteneri (*Alice*, *Bob*).
- *AUTH* – autentificarea (un mesaj semnat).
- SA_A2 – Informația propusă de *Alice* pentru crearea primului $child - SA$; poate conține de exemplu antetul de autentificare (*AH* sau *ESP*), protocoalele utilizate etc.

- HDR – antetul; include *SPI*-urile celor doi parteneri, numărul versiunii *IKE*, identificatorii de mesaje folosiți în $IKE - SA - INIT$.
- $SK\{\dots\}$ – se asigură criptarea și integritatea datelor folosind SK_e și SK_a.
- $CERT$ – un certificat al lui *Alice*, eliberat de o entitate *PKI*.
- $CERTREQ$ – O listă de autorități de certificare de încredere.
- ID_A, ID_B – identificatorii celor doi parteneri (*Alice*, *Bob*).
- $AUTH$ – autentificarea (un mesaj semnat).
- SA_A2 – Informația propusă de *Alice* pentru crearea primului $child - SA$; poate conține de exemplu antetul de autentificare (*AH* sau *ESP*), protocoalele utilizate etc.
- TS_A, TS_B – selectori de trafic. Transmit celor doi parteneri adresele de contact, porturile și protocolul *IP* folosit.

Exemplu

Dacă *Alice* trimite

$$TS_A = \{192.0.1.0{-}192.0.1.255\}, \quad TS_B = \{192.0.2.0{-}192.0.2.255\}$$

Exemplu

Dacă *Alice* trimite

$$TS_A = \{192.0.1.0{-}192.0.1.255\}, \quad TS_B = \{192.0.2.0{-}192.0.2.255\}$$

înseamnă că ea dorește ca toată informația să îi fie trimisă la o adresă *IP* din domeniul

$$\{192.0.1.0 - 192.0.1.255\}$$

Exemplu

Dacă *Alice* trimite

$$TS_A = \{192.0.1.0{-}192.0.1.255\}, \quad TS_B = \{192.0.2.0{-}192.0.2.255\}$$

înseamnă că ea dorește ca toată informația să îi fie trimisă la o adresă *IP* din domeniul

$$\{192.0.1.0 - 192.0.1.255\}$$

și solicită ca tot ce transmite pe adresa lui *Bob* să fie la o adresă *IP* din domeniul

$$TS_B = \{192.0.2.0 - 192.0.2.255\}$$

- *Bob* răspunde (Pas 4) cu mesajul

$$HDR \| SK\{ID_B, [CERT], AUTH, SA_B2, TS_A, TS_B\}$$

- *Bob* răspunde (Pas 4) cu mesajul

$$HDR \| SK\{ID_B, [CERT], AUTH, SA_B2, TS_A, TS_B\}$$

unde

- *HDR* include *SPI*-ul lui *Alice* și *Bob*, versiunea *IKE* și identificatorul de mesaj trimis de *Alice*.

- *Bob* răspunde (Pas 4) cu mesajul

$$HDR \| SK\{ID_B, [CERT], AUTH, SA_B2, TS_A, TS_B\}$$

unde

- *HDR* include *SPI*-ul lui *Alice* și *Bob*, versiunea *IKE* și identificatorul de mesaj trimis de *Alice*.
- *CERT* (opțional) este certificatul lui *Bob* – trimis doar la cerere.

- *Bob* răspunde (Pas 4) cu mesajul

$$HDR \| SK\{ID_B, [CERT], AUTH, SA_B2, TS_A, TS_B\}$$

unde

- *HDR* include *SPI*-ul lui *Alice* și *Bob*, versiunea *IKE* și identificatorul de mesaj trimis de *Alice*.
- *CERT* (opțional) este certificatul lui *Bob* – trimis doar la cerere.
- *AUTH* – câmp utilizat de *Bob* pentru a se autentifica față de *Alice*.

- *Bob* răspunde (Pas 4) cu mesajul

$$HDR \| SK\{ID_B, [CERT], AUTH, SA_B2, TS_A, TS_B\}$$

unde

- *HDR* include *SPI*-ul lui *Alice* și *Bob*, versiunea *IKE* și identificatorul de mesaj trimis de *Alice*.
- *CERT* (opțional) este certificatul lui *Bob* – trimis doar la cerere.
- *AUTH* – câmp utilizat de *Bob* pentru a se autentifica față de *Alice*.
- SA_B2 completează negocierea pentru crearea $child - SA$, acceptând algoritmi propuși de *Alice* și identificând protocolul negociat (*AH* sau *ESP*).

- *Bob* răspunde (Pas 4) cu mesajul

$$HDR \| SK\{ID_B, [CERT], AUTH, SA_B2, TS_A, TS_B\}$$

unde

- *HDR* include *SPI*-ul lui *Alice* și *Bob*, versiunea *IKE* și identificatorul de mesaj trimis de *Alice*.
- *CERT* (opțional) este certificatul lui *Bob* – trimis doar la cerere.
- *AUTH* – câmp utilizat de *Bob* pentru a se autentifica față de *Alice*.
- SA_B2 completează negocierea pentru crearea $child - SA$, acceptând algoritmi propuși de *Alice* și identificând protocolul negociat (*AH* sau *ESP*).
- TS_A, TS_B sunt selectorii de trafic.
  Dacă *Bob* este de acord cu selectorii propuși de *Alice*, aceste valori sunt identice cu cele din mesajul anterior.

Chiar și în cele mai simple scenarii de comunicare, aceste prime patru schimburi de mesaje (din $IKE - SA - INIT$ și $IKE - SA - AUTH$) sunt destul de costisitoare ca resurse.

Chiar și în cele mai simple scenarii de comunicare, aceste prime patru schimburi de mesaje (din $IKE - SA - INIT$ și $IKE - SA - AUTH$) sunt destul de costisitoare ca resurse.
În cele mai multe situații ele sunt însă preferabile, deoarece:

1. Deși entități ca serverele *IPSec* consumă timp pentru prelucrarea acestor mesaje, din informația obținută se pot crea $child - SA$ multiple care vor putea fi folosite în contactele următoare, fără a mai trece prin acest start complet.

Chiar și în cele mai simple scenarii de comunicare, aceste prime patru schimburi de mesaje (din $IKE - SA - INIT$ și $IKE - SA - AUTH$) sunt destul de costisitoare ca resurse.
În cele mai multe situații ele sunt însă preferabile, deoarece:

1. Deși entități ca serverele *IPSec* consumă timp pentru prelucrarea acestor mesaje, din informația obținută se pot crea $child - SA$ multiple care vor putea fi folosite în contactele următoare, fără a mai trece prin acest start complet.
2. $IKE - INIT$ stabilește un $IKE - SA$ care include informația secretă (partajată) utilizată în generarea asocierilor de securitate $child - SA$.

Chiar și în cele mai simple scenarii de comunicare, aceste prime patru schimburi de mesaje (din $IKE - SA - INIT$ și $IKE - SA - AUTH$) sunt destul de costisitoare ca resurse.
În cele mai multe situații ele sunt însă preferabile, deoarece:

1. Deși entități ca serverele *IPSec* consumă timp pentru prelucrarea acestor mesaje, din informația obținută se pot crea $child - SA$ multiple care vor putea fi folosite în contactele următoare, fără a mai trece prin acest start complet.
2. $IKE - INIT$ stabilește un $IKE - SA$ care include informația secretă (partajată) utilizată în generarea asocierilor de securitate $child - SA$.
3. În *IKEv2*, primul $child - SA$ este creat pe baza schimbului de mesaje $IKE - SA - AUTH$.
   Celelalte asocieri de securitate $child - SA$ vor necesita un singur schimb de mesaje – extrem de simplu și rapid – în $CREATE - child - SA$,

CREATE-child-SA

Noul schimb de mesaje dintre parteneri are ca scop generarea unor $child - SA$ noi și modificarea cheilor asocierilor de securitate active. Toate mesajele sunt protejate criptografic folosind algoritmi de criptare și chei negociate în $IKE - SA - INIT$ și $IKE - SA - AUTH$.

CREATE-child-SA

Noul schimb de mesaje dintre parteneri are ca scop generarea unor $child - SA$ noi și modificarea cheilor asocierilor de securitate active. Toate mesajele sunt protejate criptografic folosind algoritmi de criptare și chei negociate în $IKE - SA - INIT$ și $IKE - SA - AUTH$.

- Protocolul începe cu *Alice* care trimite mesajul

$$HDR \| SK\{[N^+], SA, N_A, [KE_A], TS_A, TS_B\}$$

- *HDR* – antet care include *SPI*-urile celor doi parteneri, numărul de versiune *IKE* și identificatorii de mesaj.

- *HDR* – antet care include *SPI*-urile celor doi parteneri, numărul de versiune *IKE* și identificatorii de mesaj.
- $SK\{\dots\}$ – criptarea datelor curente cu cheia SK_e și asigurarea integrității cu cheia SK_a.

- HDR – antet care include *SPI*-urile celor doi parteneri, numărul de versiune *IKE* și identificatorii de mesaj.
- $SK\{\dots\}$ – criptarea datelor curente cu cheia SK_e și asigurarea integrității cu cheia SK_a.
- $[N^+]$ – notificare (opțională) care conține detalii suplimentare pentru $child - SA$.

- HDR – antet care include *SPI*-urile celor doi parteneri, numărul de versiune *IKE* și identificatorii de mesaj.
- $SK\{\dots\}$ – criptarea datelor curente cu cheia SK_e și asigurarea integrității cu cheia SK_a.
- $[N^+]$ – notificare (opțională) care conține detalii suplimentare pentru $child - SA$.
- SA – asocierea de securitate propusă de *Alice*.

- HDR – antet care include *SPI*-urile celor doi parteneri, numărul de versiune *IKE* și identificatorii de mesaj.
- $SK\{\dots\}$ – criptarea datelor curente cu cheia SK_e și asigurarea integrității cu cheia SK_a.
- $[N^+]$ – notificare (opțională) care conține detalii suplimentare pentru $child - SA$.
- SA – asocierea de securitate propusă de *Alice*.
- N_A – un nonce.

- HDR – antet care include *SPI*-urile celor doi parteneri, numărul de versiune *IKE* și identificatorii de mesaj.
- $SK\{\dots\}$ – criptarea datelor curente cu cheia SK_e și asigurarea integrității cu cheia SK_a.
- $[N^+]$ – notificare (opțională) care conține detalii suplimentare pentru $child - SA$.
- SA – asocierea de securitate propusă de *Alice*.
- N_A – un nonce.
- KE_A – o valoare nouă g^a pentru cheia Diffie-Hellman (opțional).

- HDR – antet care include *SPI*-urile celor doi parteneri, numărul de versiune *IKE* și identificatorii de mesaj.
- $SK\{\dots\}$ – criptarea datelor curente cu cheia SK_e și asigurarea integrității cu cheia SK_a.
- $[N^+]$ – notificare (opțională) care conține detalii suplimentare pentru $child - SA$.
- SA – asocierea de securitate propusă de *Alice*.
- N_A – un nonce.
- KE_A – o valoare nouă g^a pentru cheia Diffie-Hellman (opțional).
- $TS_A,\ TS_B$ – selectori de trafic.

- HDR – antet care include *SPI*-urile celor doi parteneri, numărul de versiune *IKE* și identificatorii de mesaj.
- $SK\{\dots\}$ – criptarea datelor curente cu cheia SK_e și asigurarea integrității cu cheia SK_a.
- $[N^+]$ – notificare (opțională) care conține detalii suplimentare pentru $child - SA$.
- SA – asocierea de securitate propusă de *Alice*.
- N_A – un nonce.
- KE_A – o valoare nouă g^a pentru cheia Diffie-Hellman (opțional).
- $TS_A,\ TS_B$ – selectori de trafic.

Tot mesajul este criptat și integritatea sa protejată folosind cheile calculate din SK_d.

- *Bob* răspunde cu

$$HDR \| SK\{[N^+], SA, N_B, [KE_B], TS_A, TS_B\}$$

- *Bob* răspunde cu

$$HDR \| SK\{[N^+], SA, N_B, [KE_B], TS_A, TS_B\}$$

- *HDR* – similar antetului din mesajul primit.

- *Bob* răspunde cu

$$HDR \| SK\{[N^+], SA, N_B, [KE_B], TS_A, TS_B\}$$

- HDR – similar antetului din mesajul primit.
- $[N^+]$ – notificare (opțională) cu detalii suplimentare pentru $child - SA$.

- *Bob* răspunde cu

$$HDR \| SK\{[N^+], SA, N_B, [KE_B], TS_A, TS_B\}$$

- HDR – similar antetului din mesajul primit.
- $[N^+]$ – notificare (opțională) cu detalii suplimentare pentru $child - SA$.
- SA – algoritmii pe care îi agrează din asocierea de securitate primită.

- *Bob* răspunde cu

$$HDR \| SK\{[N^+], SA, N_B, [KE_B], TS_A, TS_B\}$$

- HDR – similar antetului din mesajul primit.
- $[N^+]$ – notificare (opțională) cu detalii suplimentare pentru $child - SA$.
- SA – algoritmii pe care îi agrează din asocierea de securitate primită.
- N_B – un nonce propriu.

- *Bob* răspunde cu

$$HDR \| SK\{[N^+], SA, N_B, [KE_B], TS_A, TS_B\}$$

- HDR – similar antetului din mesajul primit.
- $[N^+]$ – notificare (opțională) cu detalii suplimentare pentru $child - SA$.
- SA – algoritmii pe care îi agrează din asocierea de securitate primită.
- N_B – un nonce propriu.
- $[KE_B]$ – o nouă valoare Diffie-Hellman g^b (opțional).

- *Bob* răspunde cu

$$HDR \| SK\{[N^+], SA, N_B, [KE_B], TS_A, TS_B\}$$

- HDR – similar antetului din mesajul primit.
- $[N^+]$ – notificare (opțională) cu detalii suplimentare pentru $child - SA$.
- SA – algoritmii pe care îi agrează din asocierea de securitate primită.
- N_B – un nonce propriu.
- $[KE_B]$ – o nouă valoare Diffie-Hellman g^b (opțional).
- $T_A,\ TS_B$ – selectorii de trafic agreați.

- *Bob* răspunde cu

$$HDR \| SK\{[N^+], SA, N_B, [KE_B], TS_A, TS_B\}$$

- HDR – similar antetului din mesajul primit.
- $[N^+]$ – notificare (opțională) cu detalii suplimentare pentru $child - SA$.
- SA – algoritmii pe care îi agrează din asocierea de securitate primită.
- N_B – un nonce propriu.
- $[KE_B]$ – o nouă valoare Diffie-Hellman g^b (opțional).
- $T_A,\ TS_B$ – selectorii de trafic agreați.

Și acest mesaj este criptat și integritatea sa protejată folosind cheile calculate din SK_d.

Într-o asociere de securitate, *IKE*, *ESP* și *AH* folosesc cheile secrete doar o perioadă limitată de timp și numai pentru o cantitate limitată de date.

Într-o asociere de securitate, *IKE*, *ESP* și *AH* folosesc cheile secrete doar o perioadă limitată de timp și numai pentru o cantitate limitată de date.
După expirarea unui *SA* este stabilită o nouă asociere de securitate, derivând alte chei din $IKE - SA$ sau $child - SA$.

Într-o asociere de securitate, *IKE*, *ESP* și *AH* folosesc cheile secrete doar o perioadă limitată de timp și numai pentru o cantitate limitată de date.
După expirarea unui *SA* este stabilită o nouă asociere de securitate, derivând alte chei din $IKE - SA$ sau $child - SA$.

- Dacă $CREATE - child - SA$ este folosit în obținerea noilor chei pentru $IKE - SA$, atunci are loc schimbul de mesaje
  1. $Alice \longrightarrow Bob: \quad HDR \| SK\{SA, N_A, [Ke_A]\}$

Într-o asociere de securitate, *IKE*, *ESP* și *AH* folosesc cheile secrete doar o perioadă limitată de timp și numai pentru o cantitate limitată de date.
După expirarea unui *SA* este stabilită o nouă asociere de securitate, derivând alte chei din $IKE - SA$ sau $child - SA$.

- Dacă $CREATE - child - SA$ este folosit în obținerea noilor chei pentru $IKE - SA$, atunci are loc schimbul de mesaje
  1. $Alice \longrightarrow Bob: \quad HDR \| SK\{SA, N_A, [Ke_A]\}$
  2. $Bob \longrightarrow Alice: \quad HDR \| SK\{SA, N_B, [Ke_B]\}$

- Dacă *CREATE* – *child* – *SA* este folosit în obținerea noilor chei pentru *child* – *SA*, atunci are loc schimbul de mesaje

- Dacă $CREATE - child - SA$ este folosit în obținerea noilor chei pentru $child - SA$, atunci are loc schimbul de mesaje
  1. $Alice \longrightarrow Bob$:
     $HDR \| SK\{N(REKEY - SA), [N^+], SA, N_A, [Ke_A], TS_A, TS_B\}$

- Dacă $CREATE - child - SA$ este folosit în obținerea noilor chei pentru $child - SA$, atunci are loc schimbul de mesaje
  1. $Alice \longrightarrow Bob$:
     $HDR \| SK\{N(REKEY - SA), [N^+], SA, N_A, [Ke_A], TS_A, TS_B\}$
  2. $Bob \longrightarrow Alice$: $HDR \| SK\{[N^+], SA, N_B, [Ke_B], TS_A, TS_B\}$

- Dacă $CREATE - child - SA$ este folosit în obținerea noilor chei pentru $child - SA$, atunci are loc schimbul de mesaje
  1. $Alice \longrightarrow Bob$:
     $HDR \| SK\{N(REKEY - SA), [N^+], SA, N_A, [Ke_A], TS_A, TS_B\}$
  2. $Bob \longrightarrow Alice$: $HDR \| SK\{[N^+], SA, N_B, [Ke_B], TS_A, TS_B\}$

De remarcat că *Alice* identifică $child - SA$ ale cărui chei trebuie schimbate notificând datele curente prin $N(REKEY - SA)$.

Schimbul de informații în *IKE*

După crearea unui $IKE - SA$, *Alice* și *Bob* își pot transmite mesaje de control privind erori sau diverse notificări. Mesajele schimbate pot fi notificări (N), ștergeri (D) și configurări de date (Configuration Payloads - CP).

Schimbul de informații în *IKE*

După crearea unui $IKE - SA$, *Alice* și *Bob* își pot transmite mesaje de control privind erori sau diverse notificări. Mesajele schimbate pot fi notificări (N), ștergeri (D) și configurări de date (Configuration Payloads - CP).

Un schimb de informații este de forma:

$$\begin{array}{cc} \textit{Alice} & \textit{Bob} \\ HDR\|SK\{[N],[D],[CP],\dots\} & \longrightarrow \\ & \longleftarrow \quad HDR\|SK\{[N],[D],[CP],\dots\} \end{array}$$

Generarea componentelor cheii în *IKE*

Într-un *IKE* – *SA* sunt negociați patru algoritmi criptografici: algoritmi de criptare, de protecția integrității, grupul Diffie-Hellman și generatorul de numere pseudoaleatoare (*prf*).

Generarea componentelor cheii în *IKE*

Într-un $IKE - SA$ sunt negociați patru algoritmi criptografici: algoritmi de criptare, de protecția integrității, grupul Diffie-Hellman și generatorul de numere pseudoaleatoare (*prf*). Componentele cheii pentru toți algoritmii criptografici folosiți în $IKE - SA$ și $child - SA$ sunt obținute totdeauna ca ieșiri ale unui algoritm *prf*.

Generarea componentelor cheii în *IKE*

Într-un $IKE - SA$ sunt negociați patru algoritmi criptografici: algoritmi de criptare, de protecția integrității, grupul Diffie-Hellman și generatorul de numere pseudoaleatoare (*prf*).
Componentele cheii pentru toți algoritmii criptografici folosiți în $IKE - SA$ și $child - SA$ sunt obținute totdeauna ca ieșiri ale unui algoritm *prf*.
IKEv2 folosește pentru schimbul de chei numai algoritmul Diffie-Hellman.
Informația necesară (g^a, g^b și nonce-urile N_A, N_B) este în KE_A respectiv KE_B din $IKE - INIT$.

Generarea componentelor cheii în *IKE*

Într-un $IKE - SA$ sunt negociați patru algoritmi criptografici: algoritmi de criptare, de protecția integrității, grupul Diffie-Hellman și generatorul de numere pseudoaleatoare (prf).
Componentele cheii pentru toți algoritmii criptografici folosiți în $IKE - SA$ și $child - SA$ sunt obținute totdeauna ca ieșiri ale unui algoritm prf.
IKEv2 folosește pentru schimbul de chei numai algoritmul Diffie-Hellman.
Informația necesară (g^a, g^b și nonce-urile N_A, N_B) este în KE_A respectiv KE_B din $IKE - INIT$.
Cheia comună de sesiune $SKEYSEED$ este calculată apoi de parteneri după formula

$$SKEYSEED = prf(N_A \| N_B, g^{ab})$$

Exemplu

Alice și *Bob* au convenit prin schimbul de mesaje din $IKE - INIT$ la valorile comune $p = 47,\ g = 12$.

Exemplu

Alice și *Bob* au convenit prin schimbul de mesaje din $IKE - INIT$ la valorile comune $p = 47,\ g = 12$.
Alice alege drept cheie secretă $a = 3$ și nonce $N_A = 11$.

Exemplu

Alice și *Bob* au convenit prin schimbul de mesaje din $IKE - INIT$ la valorile comune $p = 47,\ g = 12$.
Alice alege drept cheie secretă $a = 3$ și nonce $N_A = 11$.
Ea va calcula $g^a = 12^3 = 36\ (mod\ 47)$ și trimite lui *Bob* $KE_A = (36, 11)$.

Exemplu

Alice și *Bob* au convenit prin schimbul de mesaje din $IKE - INIT$ la valorile comune $p = 47,\ g = 12$.
Alice alege drept cheie secretă $a = 3$ și nonce $N_A = 11$.
Ea va calcula $g^a = 12^3 = 36\ (mod\ 47)$ și trimite lui *Bob* $KE_A = (36, 11)$.
Bob alege drept cheie secretă $b = 5$ și nonce $N_B = 7$.

Exemplu

Alice și *Bob* au convenit prin schimbul de mesaje din $IKE - INIT$ la valorile comune $p = 47,\ g = 12$.
Alice alege drept cheie secretă $a = 3$ și nonce $N_A = 11$.
Ea va calcula $g^a = 12^3 = 36\ (mod\ 47)$ și trimite lui *Bob* $KE_A = (36, 11)$.
Bob alege drept cheie secretă $b = 5$ și nonce $N_B = 7$.
Va calcula $g^b = 12^5 = 14\ (mod\ 47)$ și trimite lui *Alice* $KE_B = (14, 7)$.

Exemplu

Alice și *Bob* au convenit prin schimbul de mesaje din $IKE - INIT$ la valorile comune $p = 47,\ g = 12$.
Alice alege drept cheie secretă $a = 3$ și nonce $N_A = 11$.
Ea va calcula $g^a = 12^3 = 36\ (mod\ 47)$ și trimite lui *Bob* $KE_A = (36, 11)$.
Bob alege drept cheie secretă $b = 5$ și nonce $N_B = 7$.
Va calcula $g^b = 12^5 = 14\ (mod\ 47)$ și trimite lui *Alice* $KE_B = (14, 7)$.
La primirea lui KE_B, *Alice* calculează cheia comună $\left(g^b\right)^a = 14^3 = 18\ (mod\ 47)$. Similar, *Bob* primește KE_A și calculează $\left(g^a\right)^b = 36^5 = 18\ (mod\ 47)$.

Exemplu

Alice și *Bob* au convenit prin schimbul de mesaje din $IKE - INIT$ la valorile comune $p = 47,\ g = 12$.
Alice alege drept cheie secretă $a = 3$ și nonce $N_A = 11$.
Ea va calcula $g^a = 12^3 = 36\ (mod\ 47)$ și trimite lui *Bob* $KE_A = (36, 11)$.
Bob alege drept cheie secretă $b = 5$ și nonce $N_B = 7$.
Va calcula $g^b = 12^5 = 14\ (mod\ 47)$ și trimite lui *Alice* $KE_B = (14, 7)$.
La primirea lui KE_B, *Alice* calculează cheia comună $\left(g^b\right)^a = 14^3 = 18\ (mod\ 47)$. Similar, *Bob* primește KE_A și calculează $\left(g^a\right)^b = 36^5 = 18\ (mod\ 47)$.
Din aceste valori, ambii parteneri determină

$$SKEYSEED = prf(N_A \| N_B, g^{ab}) = prf(117, 18)$$

După ce *Alice* și *Bob* calculează valoarea comună *SKEYSEED*, obțin din ea alte șapte chei:

- SK_d pentru derivarea cheilor noi folosite în $child - SA$.

După ce *Alice* și *Bob* calculează valoarea comună *SKEYSEED*, obțin din ea alte șapte chei:

- SK_d pentru derivarea cheilor noi folosite în *child* – *SA*.
- SK_{aA}, SK_{aB} – pentru protejarea integrității.

După ce *Alice* și *Bob* calculează valoarea comună *SKEYSEED*, obțin din ea alte șapte chei:

- SK_d pentru derivarea cheilor noi folosite în $child - SA$.
- SK_{aA}, SK_{aB} – pentru protejarea integrității.
- SK_{eA}, SK_{eB} pentru criptare și decriptare.

După ce *Alice* și *Bob* calculează valoarea comună *SKEYSEED*, obțin din ea alte șapte chei:

- SK_d pentru derivarea cheilor noi folosite în $child - SA$.
- SK_{aA}, SK_{aB} – pentru protejarea integrității.
- SK_{eA}, SK_{eB} pentru criptare și decriptare.
- SK_{pA}, SK_{pB} pentru autentificare.

Derivatele lui *SKEYSEED* sunt calculate astfel:

$$\{SK_d \| SK_{aA} \| SK_{aB} \| SK_{eA} \| SK_{pA} \| SK_{pB}\} =$$

$$= prf^{+}(SKEYSEED, N_A \| N_B \| SPI_A \| SPI_B)$$

Derivatele lui *SKEYSEED* sunt calculate astfel:

$$\{SK_d \| SK_{aA} \| SK_{aB} \| SK_{eA} \| SK_{pA} \| SK_{pB}\} =$$

$$= prf^{+}(SKEYSEED, N_A \| N_B \| SPI_A \| SPI_B)$$

unde

- *prf* este o funcție generatoare de numere pseudoaleatoare.

Derivatele lui *SKEYSEED* sunt calculate astfel:

$$\{SK_d \| SK_{aA} \| SK_{aB} \| SK_{eA} \| SK_{pA} \| SK_{pB}\} =$$

$$= prf^{+}(SKEYSEED, N_A \| N_B \| SPI_A \| SPI_B)$$

unde

- *prf* este o funcție generatoare de numere pseudoaleatoare.
- SPI_A, SPI_B sunt parametri de securitate indexați de *Alice* respectiv *Bob*.

Derivatele lui *SKEYSEED* sunt calculate astfel:

$$\{SK_d \| SK_{aA} \| SK_{aB} \| SK_{eA} \| SK_{pA} \| SK_{pB}\} =$$

$$= prf^+(SKEYSEED, N_A \| N_B \| SPI_A \| SPI_B)$$

unde

- prf este o funcție generatoare de numere pseudoaleatoare.
- SPI_A, SPI_B sunt parametri de securitate indexați de *Alice* respectiv *Bob*.
- $prf^+(K, S) = T_1 \| T_2 \| \ldots \| T_n$, cu

$$T_1 = prf(SKEYSEED, N_A \| N_B \| SPI_A \| SPI_B \| 0x01),$$
$$T_i = prf(SKEYSEED, T_{i-1} \| N_A \| N_B \| SPI_A \| SPI_B \| 0x0i), \quad i > 1$$

Derivatele lui *SKEYSEED* sunt calculate astfel:

$$\{SK_d \| SK_{aA} \| SK_{aB} \| SK_{eA} \| SK_{pA} \| SK_{pB}\} =$$

$$= prf^+(SKEYSEED, N_A \| N_B \| SPI_A \| SPI_B)$$

unde

- prf este o funcție generatoare de numere pseudoaleatoare.
- SPI_A, SPI_B sunt parametri de securitate indexați de *Alice* respectiv *Bob*.
- $prf^+(K, S) = T_1 \| T_2 \| \ldots \| T_n$, cu

$$T_1 = prf(SKEYSEED, N_A \| N_B \| SPI_A \| SPI_B \| 0x01),$$
$$T_i = prf(SKEYSEED, T_{i-1} \| N_A \| N_B \| SPI_A \| SPI_B \| 0x0i), \quad i > 1$$

n este ales suficient de mare astfel ca să se acopere toți biții necesari definirii celor 7 chei noi.

1. Fiecare direcție de trafic folosește chei diferite; deci SK_{eA} și SK_{aA} vor proteja mesajele trimise de *Alice*, iar SK_{eB} și SK_{aB} – pe cele trimise de *Bob*.

1. Fiecare direcție de trafic folosește chei diferite; deci SK_{eA} și SK_{aA} vor proteja mesajele trimise de *Alice*, iar SK_{eB} și SK_{aB} – pe cele trimise de *Bob*.
2. Deoarece N_A, N_B sunt folosite drept chei în *prf*, aceste nonce-uri trebuie să constituie cel puțin jumătate din mărimea cheii din *prf*-ul negociat.

Generarea componentelor cheii pentru *child – SA*

Dacă în *CREATE – child – SA* se generează un nou *child – SA*, componentele cheii sunt generate astfel:

Generarea componentelor cheii pentru *child − SA*

Dacă în *CREATE − child − SA* se generează un nou *child − SA*, componentele cheii sunt generate astfel:

$$KEYCOMP = prf^{+}(SK_d, N_A \| N_B)$$

Generarea componentelor cheii pentru *child − SA*

Dacă în *CREATE − child − SA* se generează un nou *child − SA*, componentele cheii sunt generate astfel:

$$KEYCOMP = prf^{+}(SK_d, N_A \| N_B)$$

sau

$$KEYCOMP = prf^{+}(SK_d, g^{ab} \| N_A \| N_B)$$

Generarea componentelor cheii pentru *child − SA*

Dacă în *CREATE − child − SA* se generează un nou *child − SA*, componentele cheii sunt generate astfel:

$$KEYCOMP = prf^{+}(SK_d, N_A \| N_B)$$

sau

$$KEYCOMP = prf^{+}(SK_d, g^{ab} \| N_A \| N_B)$$

În primul caz, N_A și N_B sunt nonce-urile generate în comunicarea *CREATE − child − SA*.

Generarea componentelor cheii pentru $child - SA$

Dacă în $CREATE - child - SA$ se generează un nou $child - SA$, componentele cheii sunt generate astfel:

$$KEYCOMP = prf^{+}(SK_d, N_A \| N_B)$$

sau

$$KEYCOMP = prf^{+}(SK_d, g^{ab} \| N_A \| N_B)$$

În primul caz, N_A și N_B sunt nonce-urile generate în comunicarea $CREATE - child - SA$.

În al doilea caz, apare și cheia partajată g^{ab} obținută din același schimb de informații, componentă legată de Diffie-Hellman.

Integritatea și autentificarea în *IKE*

Pentru autentifcarea datelor curente, *AUTH* dispune de: metoda de autentificare folosită și datele de autentificat.

Integritatea și autentificarea în *IKE*

Pentru autentifcarea datelor curente, *AUTH* dispune de: metoda de autentificare folosită și datele de autentificat.
IKEv2 folosește drept metode de autentificare: semnătura digitală *RSA*, semnătura digitală *DSS* și codul de integritate a mesajelor bazat pe partajare.

Integritatea și autentificarea în *IKE*

Pentru autentifcarea datelor curente, *AUTH* dispune de: metoda de autentificare folosită și datele de autentificat.
IKEv2 folosește drept metode de autentificare: semnătura digitală *RSA*, semnătura digitală *DSS* și codul de integritate a mesajelor bazat pe partajare.
Mesajele din *IKE* – *AUTH* pot include un certificat sau o autoritate de certificare *CA* care legalizează semnăturile digitale folosite de parteneri. *IKEv2* permite lui *Alice* să indice autoritățile de certificare și tipurile de certificate pe care le folosește (*X*.509, *PKCS* #7, *PGP*, *DNS SIG* etc).

Integritatea și autentificarea în *IKE*

Pentru autentifcarea datelor curente, *AUTH* dispune de: metoda de autentificare folosită și datele de autentificat.
IKEv2 folosește drept metode de autentificare: semnătura digitală *RSA*, semnătura digitală *DSS* și codul de integritate a mesajelor bazat pe partajare.
Mesajele din *IKE* – *AUTH* pot include un certificat sau o autoritate de certificare *CA* care legalizează semnăturile digitale folosite de parteneri. *IKEv2* permite lui *Alice* să indice autoritățile de certificare și tipurile de certificate pe care le folosește (*X*.509, *PKCS* #7, *PGP*, *DNS SIG* etc).
După selectarea unui *CA*, protocolul acceptă autentificările acestuia asupra metodelor folosite.

Dacă *Alice* preferă o autentificare extinsă, ea nu va include la Pasul 3 datele *AUTH*. Atunci *Bob* va include – la Pasul 4 – un câmp *EAP* și trimite SA_B2, TS_A, TS_B până când autentificarea este completă.

Dacă *Alice* preferă o autentificare extinsă, ea nu va include la Pasul 3 datele *AUTH*. Atunci *Bob* va include – la Pasul 4 – un câmp *EAP* și trimite SA_B2, TS_A, TS_B până când autentificarea este completă.
În această variantă, $IKE - INIT$ și $IKE - AUTH$ vor avea forma:

1. $Alice \longrightarrow Bob: \quad HDR\|SA_A1\|KE_A\|N_A.$

Dacă *Alice* preferă o autentificare extinsă, ea nu va include la Pasul 3 datele *AUTH*. Atunci *Bob* va include – la Pasul 4 – un câmp *EAP* și trimite SA_B2, TS_A, TS_B până când autentificarea este completă.
În această variantă, $IKE - INIT$ și $IKE - AUTH$ vor avea forma:

1. $Alice \longrightarrow Bob: \quad HDR\|SA_A1\|KE_A\|N_A.$
2. $Bob \longrightarrow Alice: \quad HDR\|SA_B1\|KE_B\|N_B\|[CERTREQ]$

Dacă *Alice* preferă o autentificare extinsă, ea nu va include la Pasul 3 datele *AUTH*. Atunci *Bob* va include – la Pasul 4 – un câmp *EAP* și trimite SA_B2, TS_A, TS_B până când autentificarea este completă.
În această variantă, $IKE - INIT$ și $IKE - AUTH$ vor avea forma:

1. $Alice \longrightarrow Bob: \quad HDR\|SA_A1\|KE_A\|N_A.$
2. $Bob \longrightarrow Alice: \quad HDR\|SA_B1\|KE_B\|N_B\|[CERTREQ]$
3. $Alice \longrightarrow Bob:$
   $HDR\|SK\{ID_A, [CERTREQ], [ID_B], SA_A2, TS_A, TS_B\}$

Dacă *Alice* preferă o autentificare extinsă, ea nu va include la Pasul 3 datele *AUTH*. Atunci *Bob* va include – la Pasul 4 – un câmp *EAP* și trimite SA_B2, TS_A, TS_B până când autentificarea este completă.
În această variantă, $IKE - INIT$ și $IKE - AUTH$ vor avea forma:

1. $Alice \longrightarrow Bob: \quad HDR\|SA_A1\|KE_A\|N_A.$
2. $Bob \longrightarrow Alice: \quad HDR\|SA_B1\|KE_B\|N_B\|[CERTREQ]$
3. $Alice \longrightarrow Bob:$
   $HDR\|SK\{ID_A, [CERTREQ], [ID_B], SA_A2, TS_A, TS_B\}$
4. $Bob \longrightarrow Alice: \quad HDR\|SK\{ID_B, [CERT], AUTH, EAP\}$

Dacă *Alice* preferă o autentificare extinsă, ea nu va include la Pasul 3 datele *AUTH*. Atunci *Bob* va include – la Pasul 4 – un câmp *EAP* și trimite SA_B2, TS_A, TS_B până când autentificarea este completă.
În această variantă, $IKE - INIT$ și $IKE - AUTH$ vor avea forma:

1. $Alice \longrightarrow Bob: \quad HDR\|SA_A1\|KE_A\|N_A.$
2. $Bob \longrightarrow Alice: \quad HDR\|SA_B1\|KE_B\|N_B\|[CERTREQ]$
3. $Alice \longrightarrow Bob:$
   $HDR\|SK\{ID_A, [CERTREQ], [ID_B], SA_A2, TS_A, TS_B\}$
4. $Bob \longrightarrow Alice: \quad HDR\|SK\{ID_B, [CERT], AUTH, EAP\}$
5. $Alice \longrightarrow Bob: \quad HDR\|SK\{EAP\}$

Dacă *Alice* preferă o autentificare extinsă, ea nu va include la Pasul 3 datele *AUTH*. Atunci *Bob* va include – la Pasul 4 – un câmp *EAP* și trimite SA_B2, TS_A, TS_B până când autentificarea este completă.
În această variantă, $IKE - INIT$ și $IKE - AUTH$ vor avea forma:

1. $Alice \longrightarrow Bob: \quad HDR\|SA_A1\|KE_A\|N_A.$
2. $Bob \longrightarrow Alice: \quad HDR\|SA_B1\|KE_B\|N_B\|[CERTREQ]$
3. $Alice \longrightarrow Bob:$
   $HDR\|SK\{ID_A, [CERTREQ], [ID_B], SA_A2, TS_A, TS_B\}$
4. $Bob \longrightarrow Alice: \quad HDR\|SK\{ID_B, [CERT], AUTH, EAP\}$
5. $Alice \longrightarrow Bob: \quad HDR\|SK\{EAP\}$
6. $Bob \longrightarrow Alice: \quad HDR\|SK\{EAP\ (succes)\}$

Dacă *Alice* preferă o autentificare extinsă, ea nu va include la Pasul 3 datele *AUTH*. Atunci *Bob* va include – la Pasul 4 – un câmp *EAP* și trimite SA_B2, TS_A, TS_B până când autentificarea este completă.
În această variantă, $IKE - INIT$ și $IKE - AUTH$ vor avea forma:

1. $Alice \longrightarrow Bob: \quad HDR \| SA_A1 \| KE_A \| N_A.$
2. $Bob \longrightarrow Alice: \quad HDR \| SA_B1 \| KE_B \| N_B \| [CERTREQ]$
3. $Alice \longrightarrow Bob:$
   $HDR \| SK\{ID_A, [CERTREQ], [ID_B], SA_A2, TS_A, TS_B\}$
4. $Bob \longrightarrow Alice: \quad HDR \| SK\{ID_B, [CERT], AUTH, EAP\}$
5. $Alice \longrightarrow Bob: \quad HDR \| SK\{EAP\}$
6. $Bob \longrightarrow Alice: \quad HDR \| SK\{EAP\ (succes)\}$
7. $Alice \longrightarrow Bob: \quad HDR \| SK\{AUTH\}$

Dacă *Alice* preferă o autentificare extinsă, ea nu va include la Pasul 3 datele *AUTH*. Atunci *Bob* va include – la Pasul 4 – un câmp *EAP* și trimite SA_B2, TS_A, TS_B până când autentificarea este completă.
În această variantă, $IKE - INIT$ și $IKE - AUTH$ vor avea forma:

1. $Alice \longrightarrow Bob: \quad HDR \| SA_A1 \| KE_A \| N_A.$
2. $Bob \longrightarrow Alice: \quad HDR \| SA_B1 \| KE_B \| N_B \| [CERTREQ]$
3. $Alice \longrightarrow Bob:$
   $HDR \| SK\{ID_A, [CERTREQ], [ID_B], SA_A2, TS_A, TS_B\}$
4. $Bob \longrightarrow Alice: \quad HDR \| SK\{ID_B, [CERT], AUTH, EAP\}$
5. $Alice \longrightarrow Bob: \quad HDR \| SK\{EAP\}$
6. $Bob \longrightarrow Alice: \quad HDR \| SK\{EAP\ (succes)\}$
7. $Alice \longrightarrow Bob: \quad HDR \| SK\{AUTH\}$
8. $Bob \longrightarrow Alice: \quad HDR \| SK\{AUTH, SA_B2, TS_A, TS_B\}$

Grupul de descriptori Diffie-Hellman

Schimbul de chei Diffie-Hellman este folosit în *IKE* pentru a genera componentele cheilor. După închiderea conexiunii, ambii parteneri "uită" nu numai cheile folosite, dar și secretele care au stat la baza calculelor acestora.

Grupul de descriptori Diffie-Hellman

Schimbul de chei Diffie-Hellman este folosit în *IKE* pentru a genera componentele cheilor. După închiderea conexiunii, ambii parteneri "uită" nu numai cheile folosite, dar și secretele care au stat la baza calculelor acestora.

IKE folosește trei reprezentări de grupuri:

- de exponențiere modulară (numite *MODP*),

Grupul de descriptori Diffie-Hellman

Schimbul de chei Diffie-Hellman este folosit în *IKE* pentru a genera componentele cheilor. După închiderea conexiunii, ambii parteneri "uită" nu numai cheile folosite, dar și secretele care au stat la baza calculelor acestora.

IKE folosește trei reprezentări de grupuri:

- de exponențiere modulară (numite *MODP*),
- grupuri pe curbe eliptice peste $GF(2^n)$ (numite *EC2N*),

Grupul de descriptori Diffie-Hellman

Schimbul de chei Diffie-Hellman este folosit în *IKE* pentru a genera componentele cheilor. După închiderea conexiunii, ambii parteneri "uită" nu numai cheile folosite, dar și secretele care au stat la baza calculelor acestora.

IKE folosește trei reprezentări de grupuri:

- de exponențiere modulară (numite *MODP*),
- grupuri pe curbe eliptice peste $GF(2^n)$ (numite $EC2N$),
- grupuri pe curbe eliptice peste $GP[P]$ (numite ECP).

Grupul de descriptori Diffie-Hellman

Schimbul de chei Diffie-Hellman este folosit în *IKE* pentru a genera componentele cheilor. După închiderea conexiunii, ambii parteneri "uită" nu numai cheile folosite, dar și secretele care au stat la baza calculelor acestora.

IKE folosește trei reprezentări de grupuri:

- de exponențiere modulară (numite *MODP*),
- grupuri pe curbe eliptice peste $GF(2^n)$ (numite $EC2N$),
- grupuri pe curbe eliptice peste $GP[P]$ (numite *ECP*).

Pentru fiecare reprezentare sunt posibile diverse implementări, în funcție de parametrii selectați.

Specificații *IKEv1* și *IKEv2*

- **Grup** 2: Grup de exponențiere modulară cu un modul de 1024 biți.

Specificații *IKEv1* și *IKEv2*

- **Grup** 2: Grup de exponențiere modulară cu un modul de 1024 biți.

Este definit de:
- generatorul $g = 2$,

Specificații *IKEv1* și *IKEv2*

- **Grup** 2: Grup de exponențiere modulară cu un modul de 1024 biți.

Este definit de:
- generatorul $g = 2$,
- modulul $p = 2^{1536} - 2^{1472} - 1 + 2^{64} \cdot \{[2^{1406}\pi] + 741804\}$.

Specificații *IKEv1* și *IKEv2*

- **Grup** 2: Grup de exponențiere modulară cu un modul de 1024 biți.

Este definit de:
- generatorul $g = 2$,
- modulul $p = 2^{1536} - 2^{1472} - 1 + 2^{64} \cdot \{[2^{1406}\pi] + 741804\}$.

Valoarea lui în hexazecimal:

FFFFFFFF	*FFFFFFFF*	*C90FDAA2*	*2168C234*	*C4C6628B*	*80DC1CD1*
29024E08	*8A67CC74*	*020BBEA6*	*3B139B22*	*514A0879*	*8E3404DD*
EF9519B3	*CD3A431B*	*302B0A6D*	*F25F1437*	*4FE1356D*	*6D51C245*
E485B576	*625E7EC6*	*F44C42E9*	*A637ED6B*	*0BFF5CB6*	*F406B7ED*
EE386BFB	*5A899FA5*	*AE9F2411*	*7C4B1FE6*	*49286651*	*ECE45B3D*
C2007CB8	*A163BF05*	*98DA4836*	*1C55D39A*	*69163FA8*	*FD24CF5F*
83655D23	*DCA3AD96*	*1C62F356*	*208552BB*	*9ED52907*	*7096966D*
670C354E	*4ABC9804*	*F1746C08*	*CA237327*	*FFFFFFFF*	*FFFFFFFF*

- **Grup** 14: Grup de exponențiere modulară cu un modul de 2048 biți.

- **Grup** 14: Grup de exponențiere modulară cu un modul de 2048 biți.
- **Grup** 3: Grup pe o curbă eliptică peste $GF[2^{155}]$, extensie generată de polinomul

$$p(X) = X^{155} + X^{62} + 1.$$

- **Grup** 14: Grup de exponențiere modulară cu un modul de 2048 biți.
- **Grup** 3: Grup pe o curbă eliptică peste $GF[2^{155}]$, extensie generată de polinomul
$$p(X) = X^{155} + X^{62} + 1.$$
Ecuația curbei este $y^2 + xy = x^3 + 471951$, iar generatorul grupului este punctul $P = (x, y) = (123, 456)$.

- **Grup** 14: Grup de exponențiere modulară cu un modul de 2048 biți.
- **Grup** 3: Grup pe o curbă eliptică peste $GF[2^{155}]$, extensie generată de polinomul
$$p(X) = X^{155} + X^{62} + 1.$$
Ecuația curbei este $y^2 + xy = x^3 + 471951$, iar generatorul grupului este punctul $P = (x, y) = (123, 456)$.
Ordinul acestui grup este
4567192616659071619386556591434463519676923731 6
care se descompune în factorii primi

$$2^2 \cdot 3 \cdot 380599384721589301615546382619538626639743644 3$$

- **Grup** 14: Grup de exponențiere modulară cu un modul de 2048 biți.
- **Grup** 3: Grup pe o curbă eliptică peste $GF[2^{155}]$, extensie generată de polinomul
$$p(X) = X^{155} + X^{62} + 1.$$
Ecuația curbei este $y^2 + xy = x^3 + 471951$, iar generatorul grupului este punctul $P = (x, y) = (123, 456)$.
Ordinul acestui grup este
4567192616659071619386556591434463519676923731 6
care se descompune în factorii primi

$$2^2 \cdot 3 \cdot 380599384721589301615546382619538626639743644 3$$

- **Grup** 4: Grup pe o curbă eliptică peste $GF[2^{185}]$.

Standardele ulterioare (*RFC* 3526) specifică grupuri Diffie-Hellman mai puternice, echivalente ca nivel de securitate cu sistemul de criptare simetric *AES*.

Standardele ulterioare (*RFC* 3526) specifică grupuri Diffie-Hellman mai puternice, echivalente ca nivel de securitate cu sistemul de criptare simetric *AES*.

- **Grup** 5: Grup de exponențiere modulară cu un modul de 1536 biți.

Standardele ulterioare (*RFC* 3526) specifică grupuri Diffie-Hellman mai puternice, echivalente ca nivel de securitate cu sistemul de criptare simetric *AES*.

- **Grup** 5: Grup de exponențiere modulară cu un modul de 1536 biți.
- **Grup** 15: Grup de exponențiere modulară cu un modul de 3072 biți.

Standardele ulterioare (*RFC* 3526) specifică grupuri Diffie-Hellman mai puternice, echivalente ca nivel de securitate cu sistemul de criptare simetric *AES*.

- **Grup** 5: Grup de exponențiere modulară cu un modul de 1536 biți.
- **Grup** 15: Grup de exponențiere modulară cu un modul de 3072 biți.
- **Grup** 16: Grup de exponențiere modulară cu un modul de 4096 biți.

Standardele ulterioare (*RFC* 3526) specifică grupuri Diffie-Hellman mai puternice, echivalente ca nivel de securitate cu sistemul de criptare simetric *AES*.

- **Grup** 5: Grup de exponențiere modulară cu un modul de 1536 biți.
- **Grup** 15: Grup de exponențiere modulară cu un modul de 3072 biți.
- **Grup** 16: Grup de exponențiere modulară cu un modul de 4096 biți.
- **Grup** 17: Grup de exponențiere modulară cu un modul de 6144 biți.

Standardele ulterioare (*RFC* 3526) specifică grupuri Diffie-Hellman mai puternice, echivalente ca nivel de securitate cu sistemul de criptare simetric *AES*.

- **Grup** 5: Grup de exponențiere modulară cu un modul de 1536 biți.
- **Grup** 15: Grup de exponențiere modulară cu un modul de 3072 biți.
- **Grup** 16: Grup de exponențiere modulară cu un modul de 4096 biți.
- **Grup** 17: Grup de exponențiere modulară cu un modul de 6144 biți.
- **Grup** 18: Grup de exponențiere modulară cu un modul de 8192 biți.

Performanțe *IPSec*

Când se folosește *IPSec*, dimensiunea pachetului *IP* crește din cauza adăugării antetelor specifice (*ESP*, *AH*, noul antet *IP* în cazul modului tunel).

Performanțe *IPSec*

Când se folosește *IPSec*, dimensiunea pachetului *IP* crește din cauza adăugării antetelor specifice (*ESP*, *AH*, noul antet *IP* în cazul modului tunel).
Acest lucru duce la creșterea raportului între dimensiunea antetelor și cea a datelor, reducând banda efectivă de comunicare.

Performanțe *IPSec*

Când se folosește *IPSec*, dimensiunea pachetului *IP* crește din cauza adăugării antetelor specifice (*ESP*, *AH*, noul antet *IP* în cazul modului tunel).
Acest lucru duce la creșterea raportului între dimensiunea antetelor și cea a datelor, reducând banda efectivă de comunicare.

În plus, timpul necesar pentru construcția antetelor și aplicarea algoritmilor criptografici conduce la o întârziere suplimentară în transmisia pachetelor.

Acest dezavantaj devine supărător mai ales când capacitățile de procesare sunt mici.

Complexitate *IPSec*

- *IPSec* prezintă dezavantajul unei complexități extrem de mari; este prețul plătit pentru numărul mare de opțiuni și de flexibilitatea sa mult crescută.

Complexitate *IPSec*

- *IPSec* prezintă dezavantajul unei complexități extrem de mari; este prețul plătit pentru numărul mare de opțiuni și de flexibilitatea sa mult crescută.
  Algoritmii complecși conduc la erori greu de remediat. În plus, un sistem complex este mult mai dificil și mai costisitor de realizat și întreținut.

Complexitate *IPSec*

- *IPSec* prezintă dezavantajul unei complexități extrem de mari; este prețul plătit pentru numărul mare de opțiuni și de flexibilitatea sa mult crescută.
  Algoritmii complecși conduc la erori greu de remediat. În plus, un sistem complex este mult mai dificil și mai costisitor de realizat și întreținut.
- De multe ori există mai multe posibilități de a implementa facilități identice sau similare.

Complexitate *IPSec*

- *IPSec* prezintă dezavantajul unei complexități extrem de mari; este prețul plătit pentru numărul mare de opțiuni și de flexibilitatea sa mult crescută.
  Algoritmii complecși conduc la erori greu de remediat. În plus, un sistem complex este mult mai dificil și mai costisitor de realizat și întreținut.
- De multe ori există mai multe posibilități de a implementa facilități identice sau similare.
- În mod normal, un soft este realizat în urma unei metodologii de tip "încearcă-și-repară". Rezultatul constituie un produs mai puțin funcțional decât cel așteptat inițial.

Complexitate *IPSec*

- *IPSec* prezintă dezavantajul unei complexități extrem de mari; este prețul plătit pentru numărul mare de opțiuni și de flexibilitatea sa mult crescută.
  Algoritmii complecși conduc la erori greu de remediat. În plus, un sistem complex este mult mai dificil și mai costisitor de realizat și întreținut.
- De multe ori există mai multe posibilități de a implementa facilități identice sau similare.
- În mod normal, un soft este realizat în urma unei metodologii de tip "încearcă-și-repară". Rezultatul constituie un produs mai puțin funcțional decât cel așteptat inițial.
  Securitatea sa nu poate fi testată ușor, iar dacă părțile sistemului sunt considerate independente, analiza este și mai dificilă.

Singura modalitate de a testa securitatea unui sistem este aceea de a realiza analize care necesită mult timp și efort.

Singura modalitate de a testa securitatea unui sistem este aceea de a realiza analize care necesită mult timp și efort.
Dacă se consideră un sistem cu n opțiuni diferite, fiecare cu 2 posibilități de alegere, atunci sunt $n(n-1)/2 = \mathcal{O}(n^2)$ perechi diferite de opțiuni care pot interacționa în moduri diferite și 2^n configurații posibile.
Fiecare dintre acestea este posibil să conducă la o falie de securitate.

Singura modalitate de a testa securitatea unui sistem este aceea de a realiza analize care necesită mult timp și efort.
Dacă se consideră un sistem cu n opțiuni diferite, fiecare cu 2 posibilități de alegere, atunci sunt $n(n-1)/2 = \mathcal{O}(n^2)$ perechi diferite de opțiuni care pot interacționa în moduri diferite și 2^n configurații posibile.
Fiecare dintre acestea este posibil să conducă la o falie de securitate.
Deci numărul de puncte slabe în securitate sistemului poate crește foarte rapid odată cu complexitatea.

Singura modalitate de a testa securitatea unui sistem este aceea de a realiza analize care necesită mult timp și efort.
Dacă se consideră un sistem cu n opțiuni diferite, fiecare cu 2 posibilități de alegere, atunci sunt $n(n-1)/2 = \mathcal{O}(n^2)$ perechi diferite de opțiuni care pot interacționa în moduri diferite și 2^n configurații posibile.
Fiecare dintre acestea este posibil să conducă la o falie de securitate.
Deci numărul de puncte slabe în securitate sistemului poate crește foarte rapid odată cu complexitatea.

În concluzie, sistemele de securitate trebuie construite cât mai simplu cu putință.
IPSec este mult prea complex pentru a fi considerat cu adevărat sigur.

Documentație *IPSec*

Documentația *IPSec* este stufoasă și foarte dificil de înțeles.

Documentație *IPSec*

Documentația *IPSec* este stufoasă și foarte dificil de înțeles.
O lipsă majoră a documentației este nespecificarea clară a scopurilor *IPSec*.

Documentație *IPSec*

Documentația *IPSec* este stufoasă și foarte dificil de înțeles.
O lipsă majoră a documentației este nespecificarea clară a scopurilor *IPSec*.Un designer care dorește să utilizeze *IPSec* și nu cunoaște exact funcționalitățile sale, poate realiza un sistem care să nu atingă nivelul de securitate considerat.

Documentație *IPSec*

Documentația *IPSec* este stufoasă și foarte dificil de înțeles.
O lipsă majoră a documentației este nespecificarea clară a scopurilor *IPSec*.Un designer care dorește să utilizeze *IPSec* și nu cunoaște exact funcționalitățile sale, poate realiza un sistem care să nu atingă nivelul de securitate considerat.
Afirmația că *IPSec* oferă securitate la nivel *IP* poate conduce la o folosire neadecvată a sa.

Documentație *IPSec*

Documentația *IPSec* este stufoasă și foarte dificil de înțeles.
O lipsă majoră a documentației este nespecificarea clară a scopurilor *IPSec*.Un designer care dorește să utilizeze *IPSec* și nu cunoaște exact funcționalitățile sale, poate realiza un sistem care să nu atingă nivelul de securitate considerat.
Afirmația că *IPSec* oferă securitate la nivel *IP* poate conduce la o folosire neadecvată a sa.
Se poate considera – în mod eronat – că el este util ca sistem de securitate la nivel aplicație (de exemplu la autentificarea unui user pentru citirea e-mail-ului). *IPSec* autentifică pachetele ca provenind de la cineva care cunoaște o anumită cheie, chiar dacă se poate crede că autentifică o anumită adresă *IP*.

Documentație *IPSec*

Documentația *IPSec* este stufoasă și foarte dificil de înțeles.
O lipsă majoră a documentației este nespecificarea clară a scopurilor *IPSec*.Un designer care dorește să utilizeze *IPSec* și nu cunoaște exact funcționalitățile sale, poate realiza un sistem care să nu atingă nivelul de securitate considerat.
Afirmația că *IPSec* oferă securitate la nivel *IP* poate conduce la o folosire neadecvată a sa.
Se poate considera – în mod eronat – că el este util ca sistem de securitate la nivel aplicație (de exemplu la autentificarea unui user pentru citirea e-mail-ului). *IPSec* autentifică pachetele ca provenind de la cineva care cunoaște o anumită cheie, chiar dacă se poate crede că autentifică o anumită adresă *IP*.
Astfel de interpretări greșite ale *IPSec* pot conduce la erori.

Eliminarea funcționalităților duplicate

IPSec are două moduri de operare (*transport* și *tunel*) și două protocoale (*AH* și *ESP*), fapt ce conduce la o complexitate ridicată.
Diferența dintre acestea este minoră, atât din punct de vedere al funcționalității cât și din punct de vedere al performanței.

Eliminarea funcționalităților duplicate

IPSec are două moduri de operare (*transport* și *tunel*) și două protocoale (*AH* și *ESP*), fapt ce conduce la o complexitate ridicată.
Diferența dintre acestea este minoră, atât din punct de vedere al funcționalității cât și din punct de vedere al performanței.

Există propunerea a renunța la modul transport, fapt care ar conduce eliminarea diferențelor dintre echipamentele de rețea în host-uri și porți de securitate (*security gateways*).
Principala diferență dintre acestea este că porțile de securitate nu pot opera în modul transport.

Protocolul *AH* poate fi eliminat.

Protocolul *AH* poate fi eliminat.

Funcționalitățile oferite de *AH* și *ESP* sunt similare. În modul transport, *AH* oferă o autentificare mai puternică întrucât autentifică și câmpuri ale antetului *IP*. Dar – utilizat în mod tunel – *ESP* oferă același nivel de autentificare.

Protocolul *AH* poate fi eliminat.

Funcționalitățile oferite de *AH* și *ESP* sunt similare. În modul transport, *AH* oferă o autentificare mai puternică întrucât autentifică și câmpuri ale antetului *IP*. Dar – utilizat în mod tunel – *ESP* oferă același nivel de autentificare.

Protocolul *AH* autentifică indirect și antetele de la nivelele inferioare, fapt care atrage după sine o multitudine de probleme, fiindcă există câmpuri care se pot modifica pe parcursul drumului de la sursă la destinație.

Deci protocolul *AH* trebuie să fie conștient de ce se întâmplă la nivelele inferioare, pentru a evita utilizarea acestor câmpuri.

Protocolul *AH* poate fi eliminat.

Funcționalitățile oferite de *AH* și *ESP* sunt similare. În modul transport, *AH* oferă o autentificare mai puternică întrucât autentifică și câmpuri ale antetului *IP*. Dar – utilizat în mod tunel – *ESP* oferă același nivel de autentificare.

Protocolul *AH* autentifică indirect și antetele de la nivelele inferioare, fapt care atrage după sine o multitudine de probleme, fiindcă există câmpuri care se pot modifica pe parcursul drumului de la sursă la destinație.

Deci protocolul *AH* trebuie să fie conștient de ce se întâmplă la nivelele inferioare, pentru a evita utilizarea acestor câmpuri.

Autentificarea *ESP* în mod tunel evită aceste probleme, cu prețul modificării lărgimii benzii de comunicare.

Utilizarea deficitară

Protocolul *ESP* permite ca atât autentificarea cât și criptarea să fie opționale.

Utilizarea deficitară

Protocolul *ESP* permite ca atât autentificarea cât și criptarea să fie opționale.

Exemplu

Un administrator de sistem poate să configureze prost IPSec: el este conștient că are nevoie de criptare pentru asigurarea confidențialității datelor, deci activează criptarea, însă lasă dezactivată opțiunea de autentificare.

Utilizarea deficitară

Protocolul *ESP* permite ca atât autentificarea cât și criptarea să fie opționale.

Exemplu

Un administrator de sistem poate să configureze prost IPSec: el este conștient că are nevoie de criptare pentru asigurarea confidențialității datelor, deci activează criptarea, însă lasă dezactivată opțiunea de autentificare.

Când sunt realizate atât autentificarea cât și criptarea, *IPSec* realizează întâi criptarea și apoi autentificarea textului criptat. Ordinea este exact inversă față de cea normală: mai întâi trebuie realizată autentificarea textului clar, și apoi se criptează textul autentificat.

Simetrizarea *SA*-urilor

Un *SA* poate fi folosit numai într-o direcție; deci pentru o comunicare bidirecțională sunt necesare două *SA*-uri.

Simetrizarea *SA*-urilor

Un *SA* poate fi folosit numai într-o direcție; deci pentru o comunicare bidirecțională sunt necesare două *SA*-uri.
Fiecare din ele implementează un singur mod de operare și un singur protocol. Dacă se folosesc două protocoale (*AH* și *ESP*) pentru un același pachet, atunci sunt necesare 2 *SA*-uri.
Utilizarea a două protocoale și a două moduri de operare conduce la creșterea complexității *SA*-urilor.

Simetrizarea *SA*-urilor

Un *SA* poate fi folosit numai într-o direcție; deci pentru o comunicare bidirecțională sunt necesare două *SA*-uri.
Fiecare din ele implementează un singur mod de operare și un singur protocol. Dacă se folosesc două protocoale (*AH* și *ESP*) pentru un același pachet, atunci sunt necesare 2 *SA*-uri.
Utilizarea a două protocoale și a două moduri de operare conduce la creșterea complexității *SA*-urilor.

În practică există foarte puține situații în care traficul este necesar să fie securizat într-o singură direcție. De aceea, *SA*-urile sunt negociate în pereche. Pare mult mai logic ca *SA*-urile să devină bidirecționale; astfel s-ar înjumătății numărul de *SA*-uri din sistem și s-ar evita de asemenea configurări asimetrice.

SPD (Security Policy Database)

Politicile de securitate sunt stocate în *SPD* (*Security Police Database*).

SPD (Security Policy Database)

Politicile de securitate sunt stocate în *SPD* (*Security Police Database*).
Pentru fiecare pachet de date transmis, se verifică cum trebuie procesat. *SPD*-ul poate specifica 3 acțiuni:

- poate respinge pachetul,

SPD (Security Policy Database)

Politicile de securitate sunt stocate în *SPD* (*Security Police Database*).
Pentru fiecare pachet de date transmis, se verifică cum trebuie procesat. *SPD*-ul poate specifica 3 acțiuni:

- poate respinge pachetul,
- poate lăsa pachetul să treacă fără să îi aplice *IPSec*,

SPD (Security Policy Database)

Politicile de securitate sunt stocate în *SPD* (*Security Police Database*).
Pentru fiecare pachet de date transmis, se verifică cum trebuie procesat. *SPD*-ul poate specifica 3 acțiuni:

- poate respinge pachetul,
- poate lăsa pachetul să treacă fără să îi aplice *IPSec*,
- poate să îi aplice *IPSec*.
  În acest caz se specifică și ce *SA* se folosește.

SPD (Security Policy Database)

Politicile de securitate sunt stocate în *SPD* (*Security Police Database*).
Pentru fiecare pachet de date transmis, se verifică cum trebuie procesat. *SPD*-ul poate specifica 3 acțiuni:

- poate respinge pachetul,
- poate lăsa pachetul să treacă fără să îi aplice *IPSec*,
- poate să îi aplice *IPSec*.
  În acest caz se specifică și ce *SA* se folosește.

Există un surplus de informație, ceea ce oferă destul de multă informație unui eventual atacator.
Se consideră că rolul *SPD*-ului este strict acela de a determina dacă un pachet trebuie aruncat, dacă trebuie transmis fără *IPSec*, dacă se autentifică sau dacă se autentifică și criptează.

Incompatibilități cu Internet Protocol

Există anumite incompatibilități între *IPSec* și *IP*.

Incompatibilități cu Internet Protocol

Există anumite incompatibilități între *IPSec* și *IP*.

Exemplu

Să considerăm două echipamente situate la o mare distanță unul de celălalt, comunică în modul tunel.

Incompatibilități cu Internet Protocol

Există anumite incompatibilități între *IPSec* și *IP*.

Exemplu

Să considerăm două echipamente situate la o mare distanță unul de celălalt, comunică în modul tunel.Între ele există routere care nu cunosc existența SA-urilor și care trebuie să transmită pachetele. Orice mesaj de tip ICMP este neautentificat și necriptat; deci el va fi eliminat, ceea ce conduce la pierderea funcționalității IP.

Incompatibilități cu Internet Protocol

Există anumite incompatibilități între *IPSec* și *IP*.

Exemplu

Să considerăm două echipamente situate la o mare distanță unul de celălalt, comunică în modul tunel.Între ele există routere care nu cunosc existența SA-urilor și care trebuie să transmită pachetele. Orice mesaj de tip ICMP este neautentificat și necriptat; deci el va fi eliminat, ceea ce conduce la pierderea funcționalității IP. Chiar dacă pe routere se implementează IPSec, tot nu se pot autentifica mesajele ICMP decât dacă acestea își setează SA cu capetele tunelului, cu scopul trimiterii pachetului ICMP.

Incompatibilități cu Internet Protocol

Există anumite incompatibilități între *IPSec* și *IP*.

Exemplu

Să considerăm două echipamente situate la o mare distanță unul de celălalt, comunică în modul tunel.Între ele există routere care nu cunosc existența SA-urilor și care trebuie să transmită pachetele. Orice mesaj de tip ICMP este neautentificat și necriptat; deci el va fi eliminat, ceea ce conduce la pierderea funcționalității IP.
Chiar dacă pe routere se implementează IPSec, tot nu se pot autentifica mesajele ICMP decât dacă acestea își setează SA cu capetele tunelului, cu scopul trimiterii pachetului ICMP.
Dar acesta trebuie să fie sigur că primește mesaje de la un router veritabil, ceea ce nu poate verifica decât eventual cu PKI, lucru inexistent la acest nivel.

Duplicarea mesajelor

În mod ideal, dacă se transmit mai multe copii ale aceluiași mesaj, atunci răspunsurile trebuie să fie identice.

Mai mult, sistemul ar trebui să recunoască faptul că primește copii identice și să replice primul răspuns transmis.

Duplicarea mesajelor

În mod ideal, dacă se transmit mai multe copii ale aceluiași mesaj, atunci răspunsurile trebuie să fie identice.

Mai mult, sistemul ar trebui să recunoască faptul că primește copii identice și să replice primul răspuns transmis.

O implementare mai puțin atentă poate crea falii de securitate.

Exemplu

Se transmit mai multe mesaje cu zona de date modificată.
Se presupune că implementarea Diffie - Hellman modulo p se realizează peste un subgrup de dimensiune q, unde $q|p-1$ și $p-1$ are mulți divizori mici.

Exemplu

Se transmit mai multe mesaje cu zona de date modificată.
Se presupune că implementarea Diffie - Hellman modulo p se realizează peste un subgrup de dimensiune q, unde $q|p-1$ și $p-1$ are mulți divizori mici.
Oscar poate determina cheia secretă prin repetarea datelor *KE* care conțin elemente de ordin mic și verificând care din setul de chei posibile este utilizat la criptare. Aceasta îi relevă ordinul elementului trimis ca date *KE*.

Exemplu

Se transmit mai multe mesaje cu zona de date modificată.
Se presupune că implementarea Diffie - Hellman modulo p se realizează peste un subgrup de dimensiune q, unde $q|p-1$ și $p-1$ are mulți divizori mici.
Oscar poate determina cheia secretă prin repetarea datelor *KE* care conțin elemente de ordin mic și verificând care din setul de chei posibile este utilizat la criptare. Aceasta îi relevă ordinul elementului trimis ca date *KE*.
Combinând mai multe rezultate, se determină cheia secretă Diffie - Hellman.

Concluzii

IPSec constituie cel mai bun protocol de securitate existent.

Concluzii

IPSec constituie cel mai bun protocol de securitate existent. Comparat cu alte protocoale de securitate, *IPSec* oferă multe avantaje din punct de vedere arhitectural și o foarte mare flexibilitate.

Concluzii

IPSec constituie cel mai bun protocol de securitate existent. Comparat cu alte protocoale de securitate, *IPSec* oferă multe avantaje din punct de vedere arhitectural și o foarte mare flexibilitate.
Detaliile securității rețelei sunt de obicei ascunse aplicațiilor. În plus, *IPSec* poate fi transparent și utilizatorilor finali, eliminând necesitatea instruirii persoanelor în vederea utilizării mecanismelor de securitate.

Concluzii

IPSec constituie cel mai bun protocol de securitate existent. Comparat cu alte protocoale de securitate, *IPSec* oferă multe avantaje din punct de vedere arhitectural și o foarte mare flexibilitate.
Detaliile securității rețelei sunt de obicei ascunse aplicațiilor. În plus, *IPSec* poate fi transparent și utilizatorilor finali, eliminând necesitatea instruirii persoanelor în vederea utilizării mecanismelor de securitate.
Un canal sigur poate fi configurat:

- de la un cap la altul: protejează traficul dintre două entități),

Concluzii

IPSec constituie cel mai bun protocol de securitate existent. Comparat cu alte protocoale de securitate, *IPSec* oferă multe avantaje din punct de vedere arhitectural și o foarte mare flexibilitate.
Detaliile securității rețelei sunt de obicei ascunse aplicațiilor. În plus, *IPSec* poate fi transparent și utilizatorilor finali, eliminând necesitatea instruirii persoanelor în vederea utilizării mecanismelor de securitate.
Un canal sigur poate fi configurat:

- de la un cap la altul: protejează traficul dintre două entități),
- route-to-route: protejează traficul care trece peste o mulțime particulară de legături).

Concluzii

IPSec constituie cel mai bun protocol de securitate existent. Comparat cu alte protocoale de securitate, *IPSec* oferă multe avantaje din punct de vedere arhitectural și o foarte mare flexibilitate.

Detaliile securității rețelei sunt de obicei ascunse aplicațiilor. În plus, *IPSec* poate fi transparent și utilizatorilor finali, eliminând necesitatea instruirii persoanelor în vederea utilizării mecanismelor de securitate.

Un canal sigur poate fi configurat:

- de la un cap la altul: protejează traficul dintre două entități),
- route-to-route: protejează traficul care trece peste o mulțime particulară de legături).

Deci *IPSec* poate realiza securizarea utilizatorilor individuali (dacă acest lucru este necesar).

Sfârșit