IPSec

IPSec Presentation

- Introduction: IPv4
  - IP was originally introduced for
    - its simplicity, its scalability, and its openness to transmit data
  - Its use has increased thanks to Internet, Extranet, Company Networks
  - Currently, lot of causes affect all networks
    - virus, denial of services, …
    - malicious internal or external action, industrial spying ….
  - To ensure future (mobile & fix) networks
    - security is needed
    - design of new IP protocols becomes necessary
- IPSec = IP security Protocol
  - Standard developed by the IETF
  - First RFC in 1995 without management key (static view)
  - Second version in November 1998 with management key (IKE)
  - Common use between IPv4 and IPv6 (mandatory in IPv6)
- IPSec is based on a set of mechanism which protect the exchange of data on the network
- Implementation of IPSec
  - Native implementation (in the IP stack with IPSec native)
  - BITS (Bump in the Stack) : additional software
  - BITW (Bump in the Wire) : outboard cryptographic processor

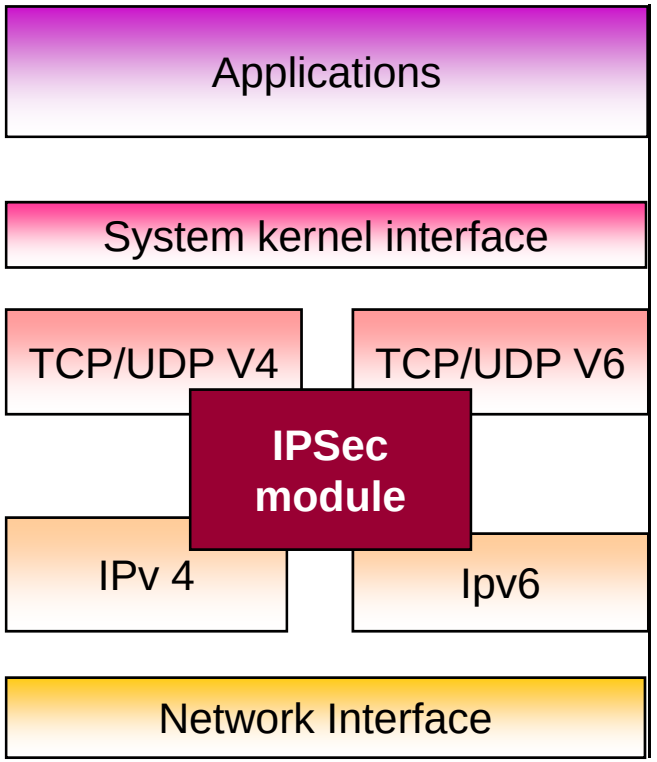
IPSec

- IPSec (IP Security) is integrated in IPv6
  - IPv6 : the protocols that define the operation of the “next generation” Internet
  - Features: Larger addresses (128 bytes), Secure communications (IPSec), QoS, Mobility
  - 6bone world-wide experimental IPv6 network
  - Migration strategies being debated
- IPSec
  - Network layer encryption and authentication
  - Open standards ensuring secure private communications
  - Provides necessary component of standards-based, flexible solution for deploying a network-wide security policy
- IPSec Status
  - Most RFCs well defined
  - Multiple implementations (Nortel, Redcreek, Sun Solaris, Microsoft, Cisco, HP, 6Wind, others)
  - PKI work underway in IETF, industry, government
  - Periodic interoperability/conformance testing using reference implementations
- Benefits of IPSec
  - Standard for privacy, integrity and authenticity for networked commerce
  - Implemented transparently in the network infrastructure
  - End-to-end security solution including routers, firewalls, PCs and servers

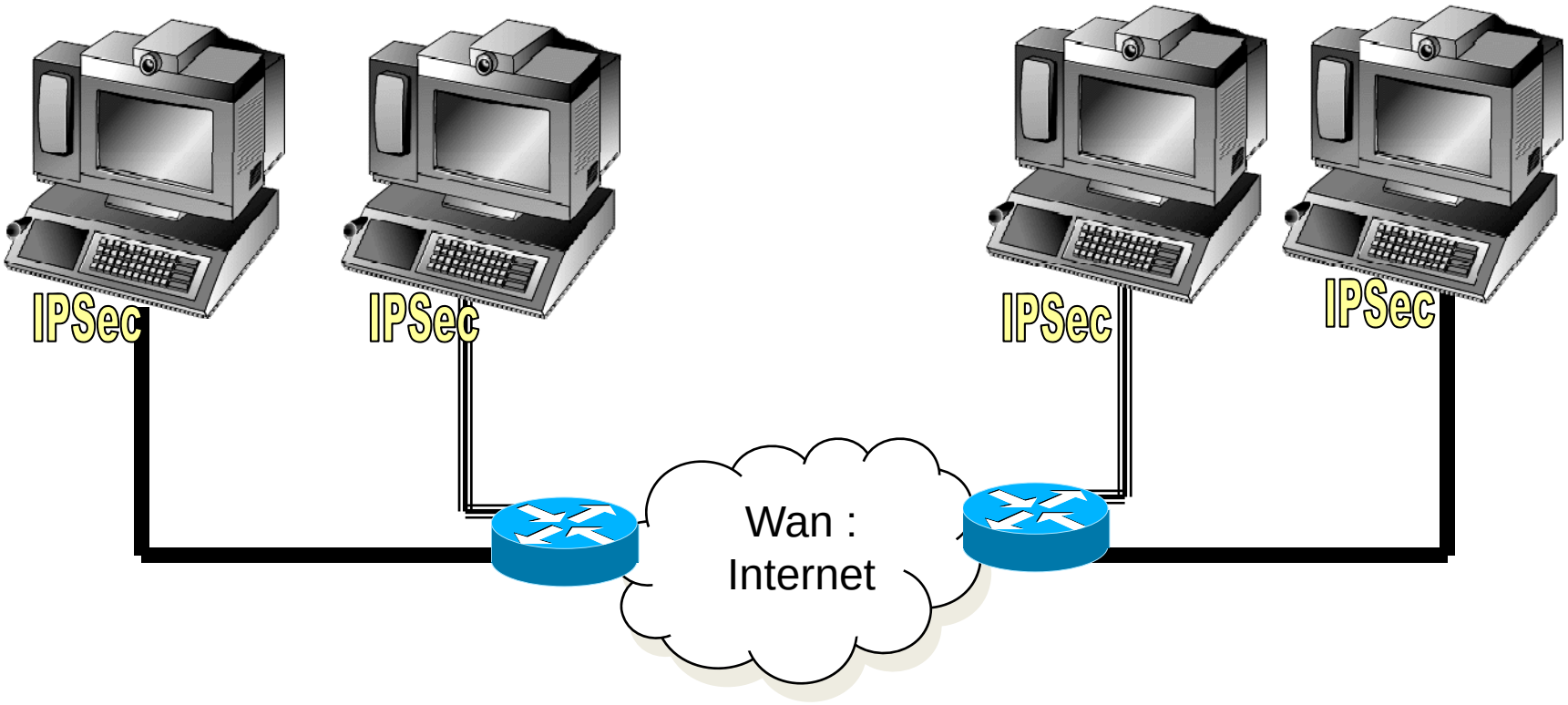
IPSec implementation : native

- Requires access to IP source code

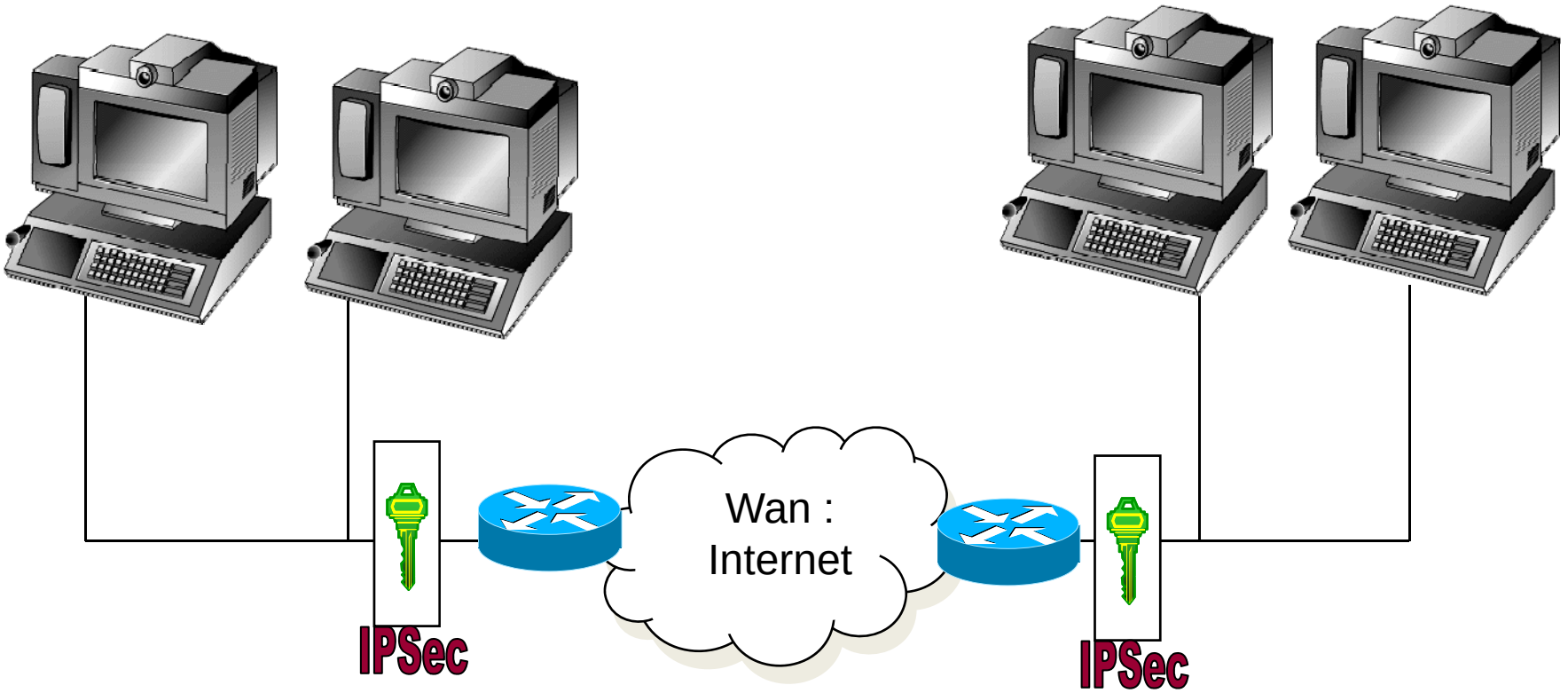
- Applicable in hosts and security gateway



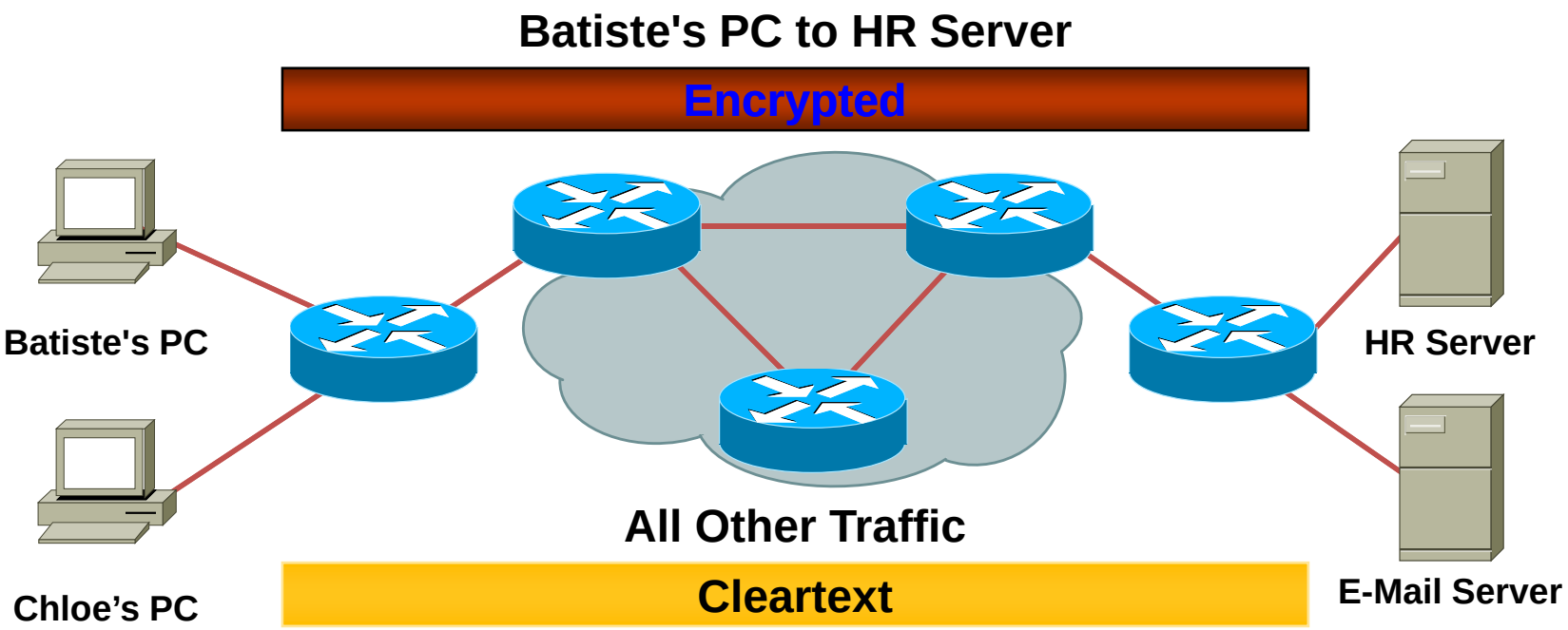
IPSec implementation : BITS



IPSec implementation : BITW



Network Layer Security



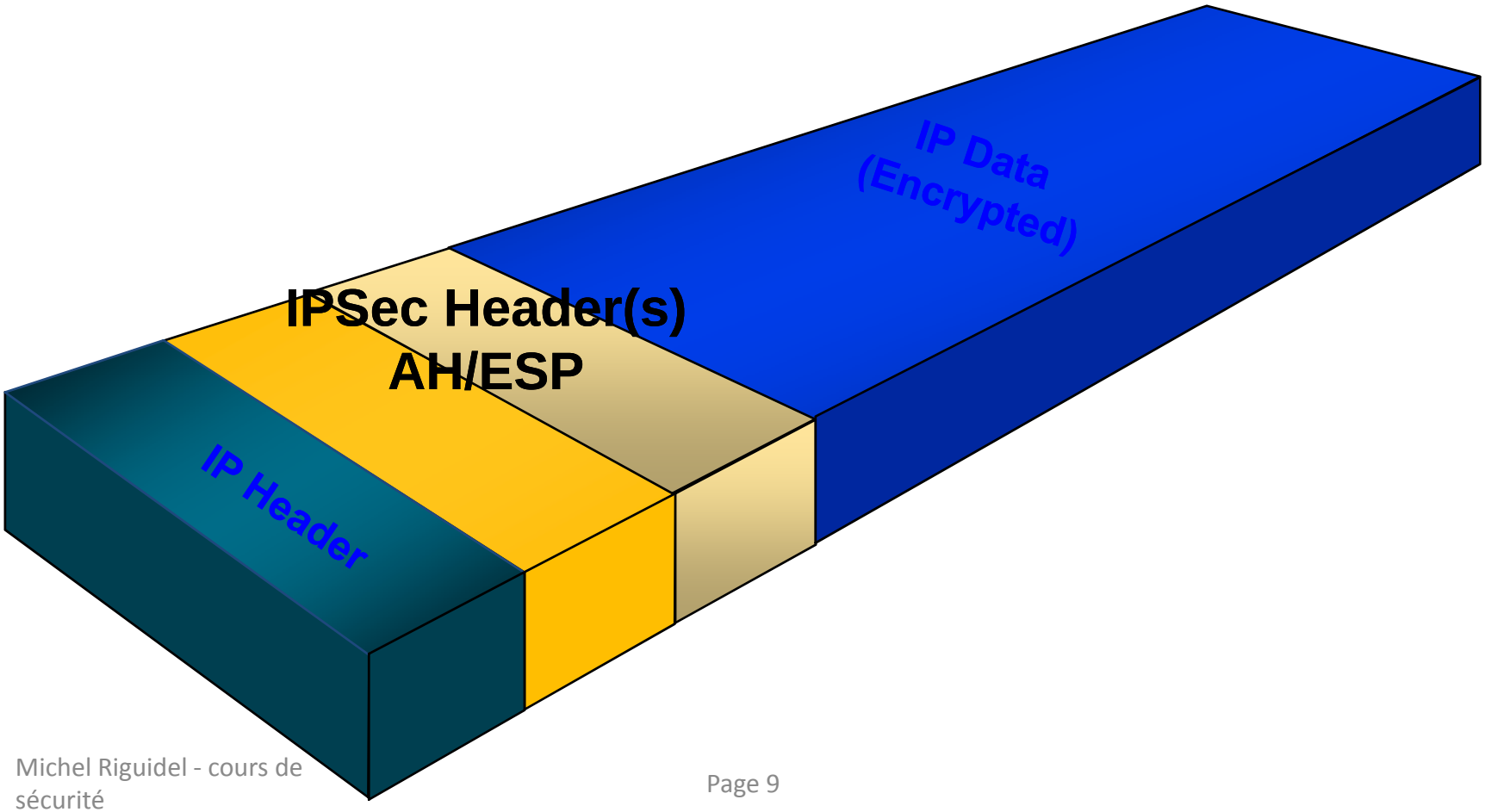
- Traffic protected on a flow-by-flow basis between specific hosts or subnets
- Media and interface independent
- Transparent to intermediate network devices
- Topology independent

Security Services provided

- Data Confidentiality
  - The transferred data arrive at their destination without being taped
- Data Integrity
  - The data received are identical to the data sent (but they could be spying)
- Data origin authentication
  - Assure that the data have been issued by the expeditor
- Access control & limited traffic flow confidentiality
  - Allow to filter the stream and to define a level of security
- Replay prevention
  - An information already sent can’t be retransmit, notably by a third party

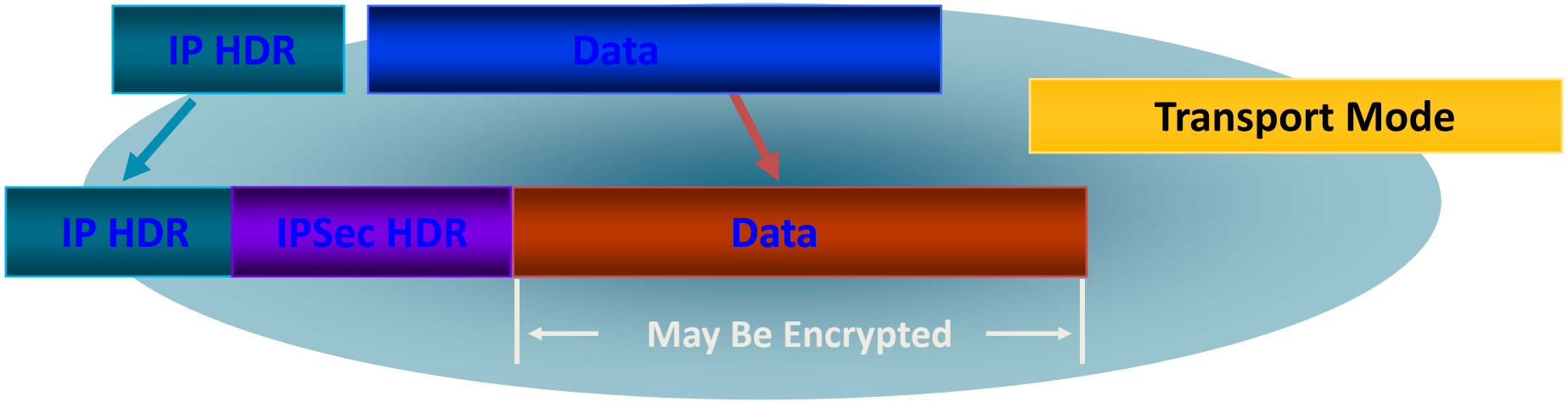
IPSec description

Interoperable Authentication, Integrity and Encryption

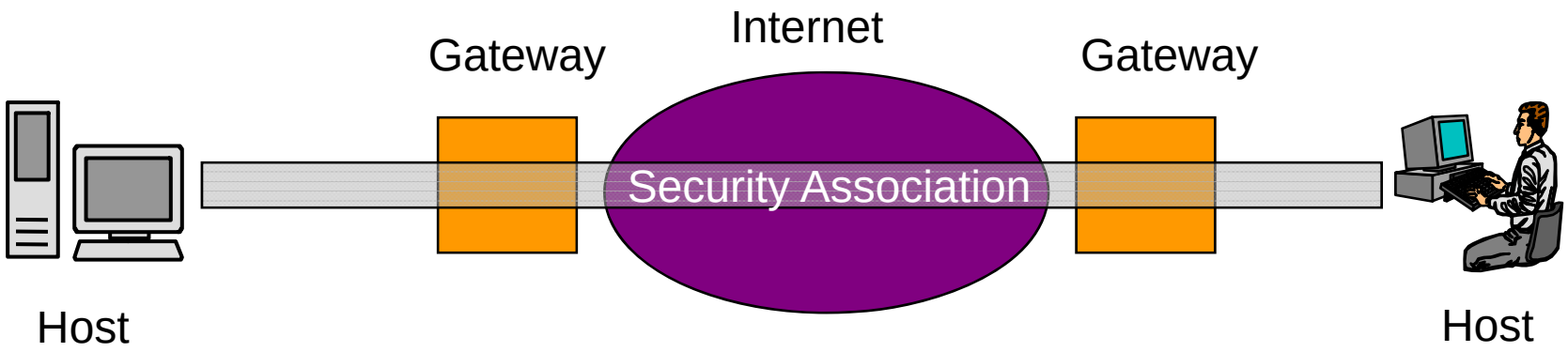


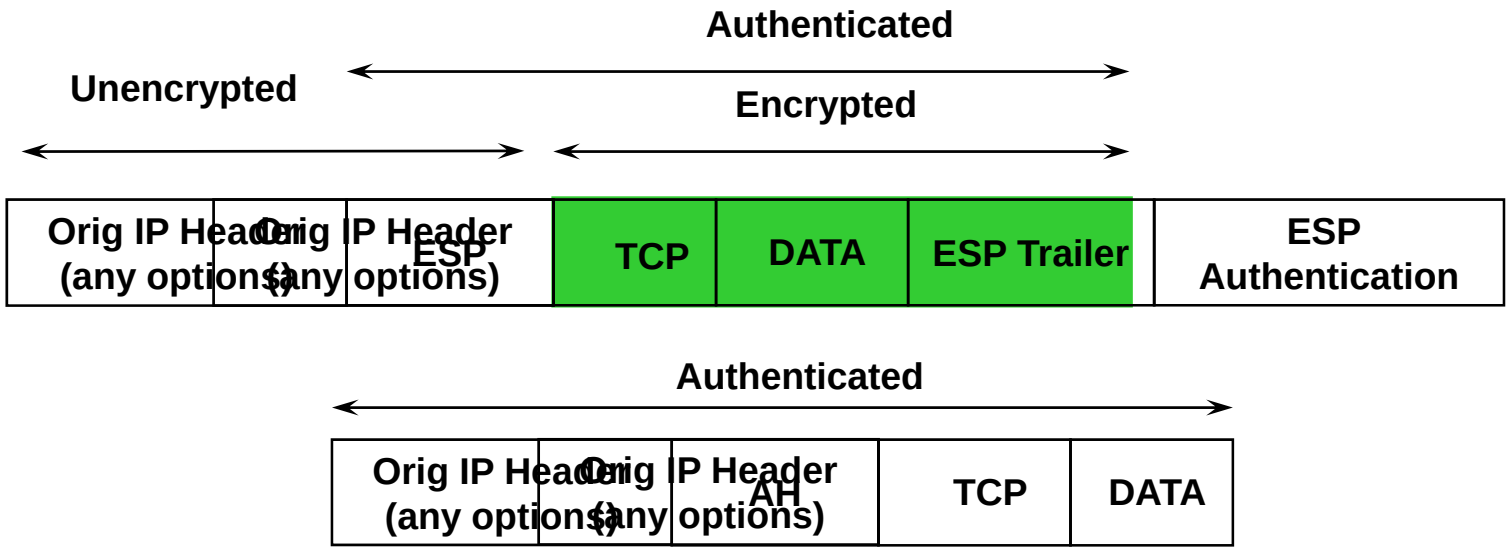
IPSec Mode : Transport

- Transport mode
  - Only the payload is encrypted
  - Implementation over IP
  - Special processing (like QoS, Multicast) enabled
  - Useful for tunneling protocol (like L2TP)



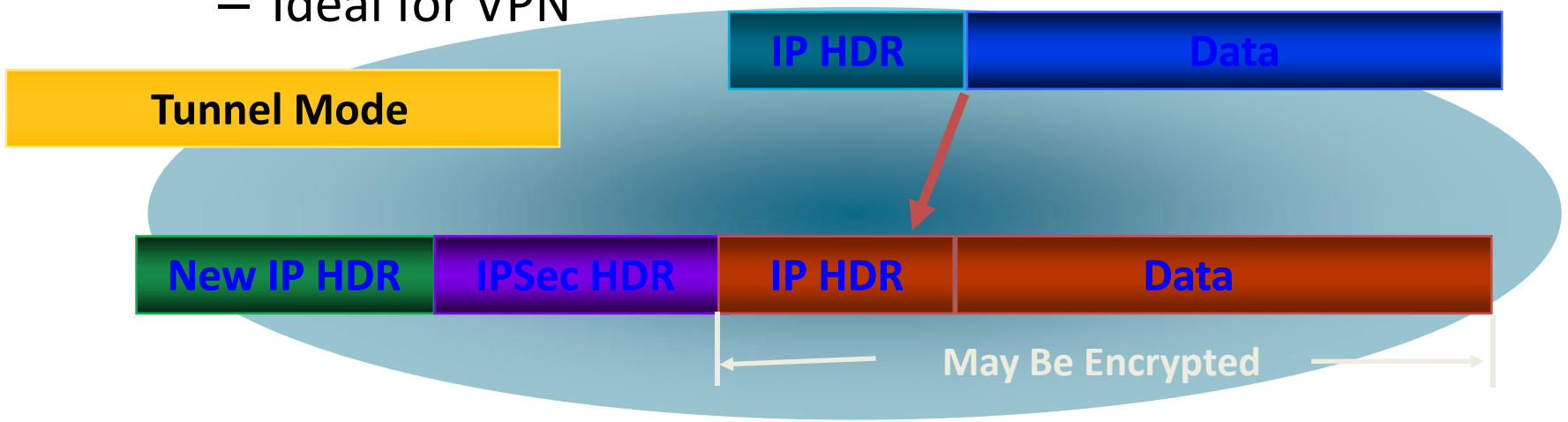
Architecture: IPSec Transport Mode



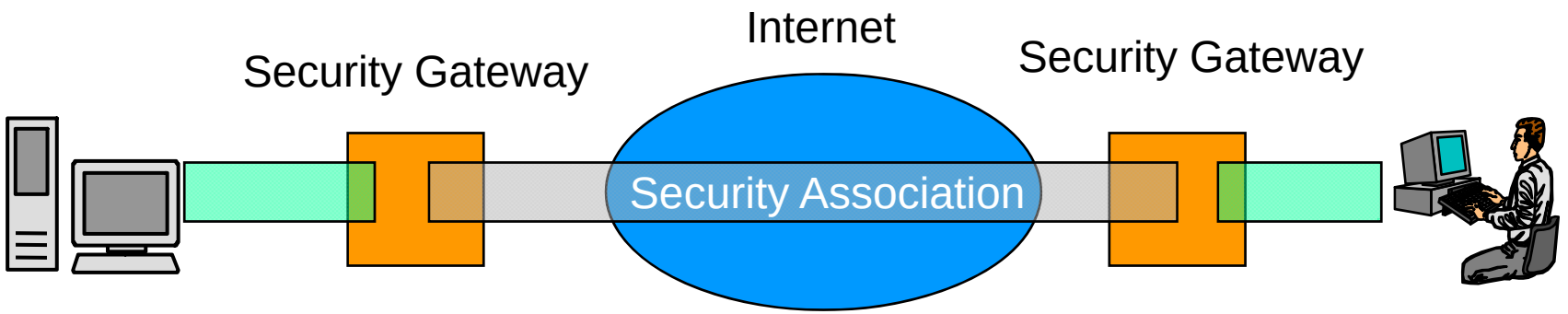


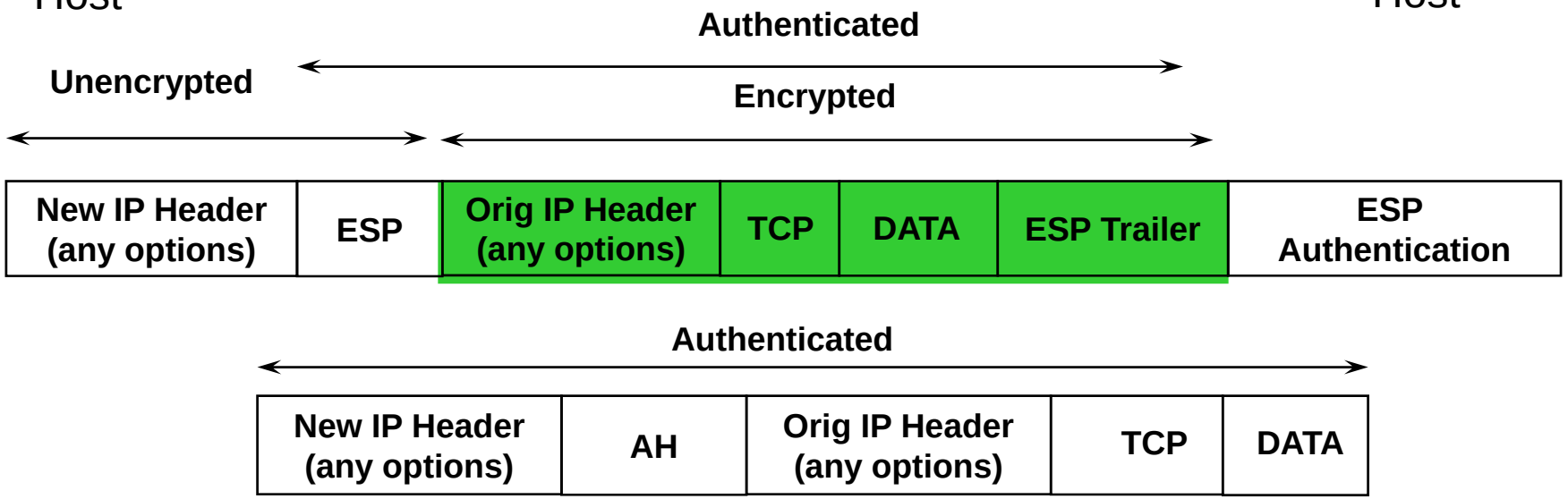
IPSec Mode : Tunnel

- Tunnel mode
  - All IP datagram are encrypted
  - Implementation above IP
  - ESP tunnel mode :
    - can provide more security
    - less complexity and cost
  - Ideal for VPN



Architecture: IPSec Tunnel Mode





How providing those services : AH & ESP Protocols

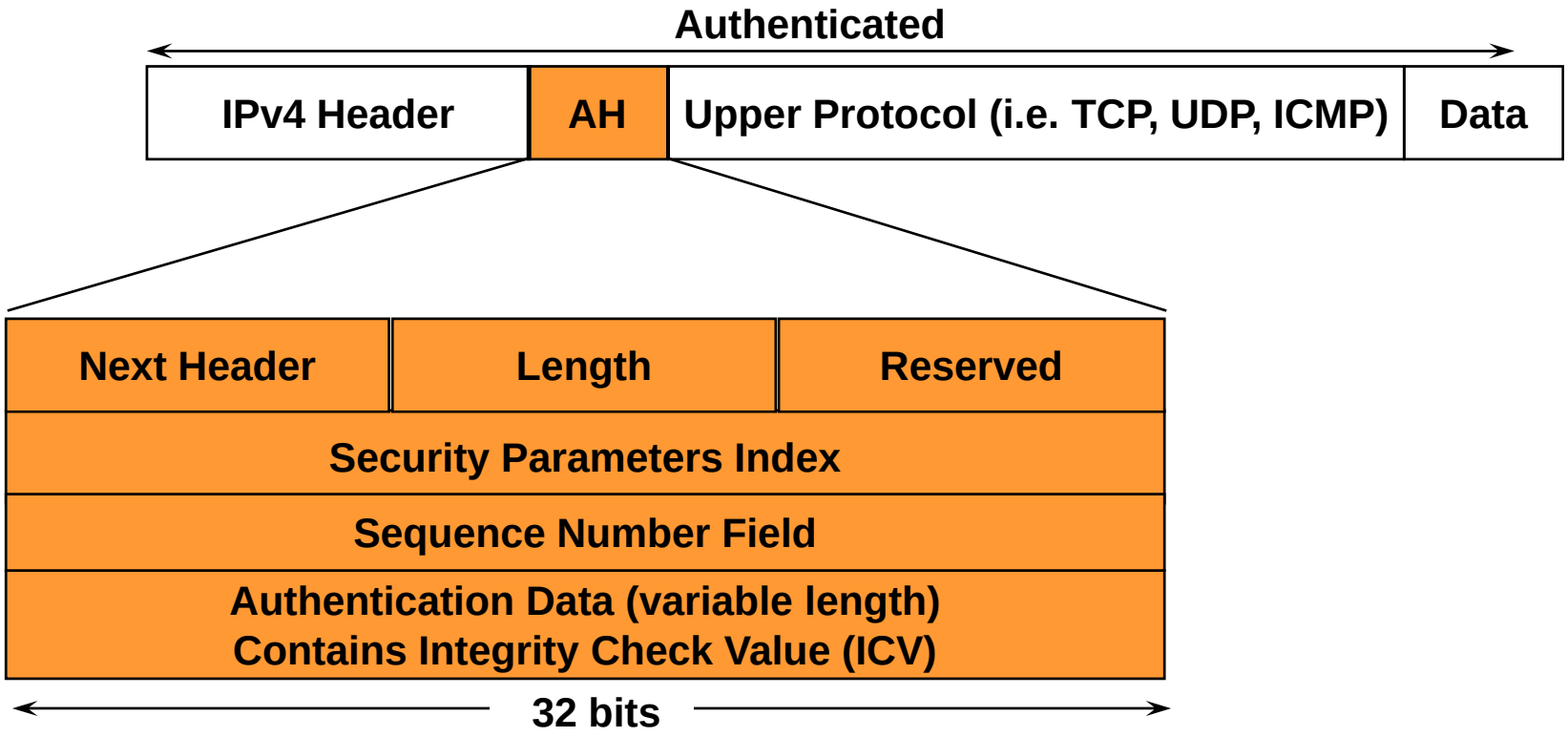
- They both offer integrity and authentication
- Based on cryptography algorithms
  - Bulk encryption algorithm (RC5, DES, 3DES, AES ..) for ESP
  - Keyed Hash algorithm (HMAC) combined with traditional hash algorithm (MD5, SHA-1) for authentication
- AH
  - Assure integrity and authentication by adding a new field in the IP datagram
- ESP
  - Assure integrity, authentication and/or confidentiality by adding a new field and encrypting the data
- AH & ESP default algorithms
  - HMAC based on MD5 or SHA-1 for authentication
  - DES-CBC (moving to AES) for confidentiality
  - Anti-replay sequence number receive window size:
    - Recommended size is 64
    - Minimum of 32 required
  - NULL encryption and NULL authentication algorithms for ESP (when encryption or authentication is not required)

Authentication Header (AH)

- Data integrity
- Data origin authentication
- Anti-replay protection
- Protects the IP header
- No confidentiality

Protocols: AH
Authentication Header

- **Provides:**
  - Origin Authentication, Integrity, Anti-replay protection, does not provide encryption

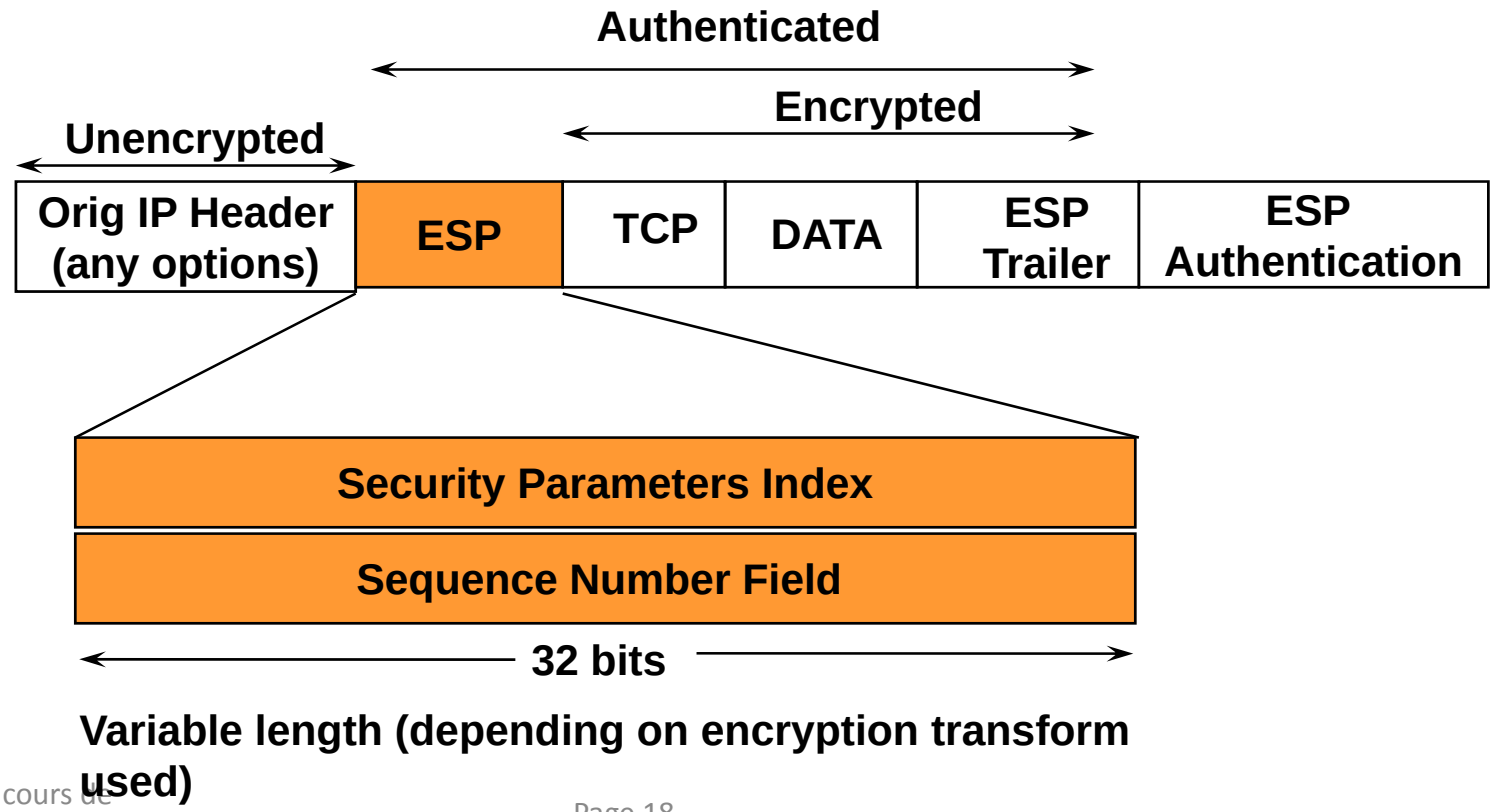


Encapsulating Security Payload (ESP)

- Data confidentiality
- Limited traffic flow confidentiality
- Data integrity
- Data origin authentication
- Anti-replay protection
- Does not protect IP Header

Protocols: ESP
Encapsulating Security Payload

- Provides:
  - Confidentiality (Encryption), Origin Authentication, Integrity, Anti-replay protection

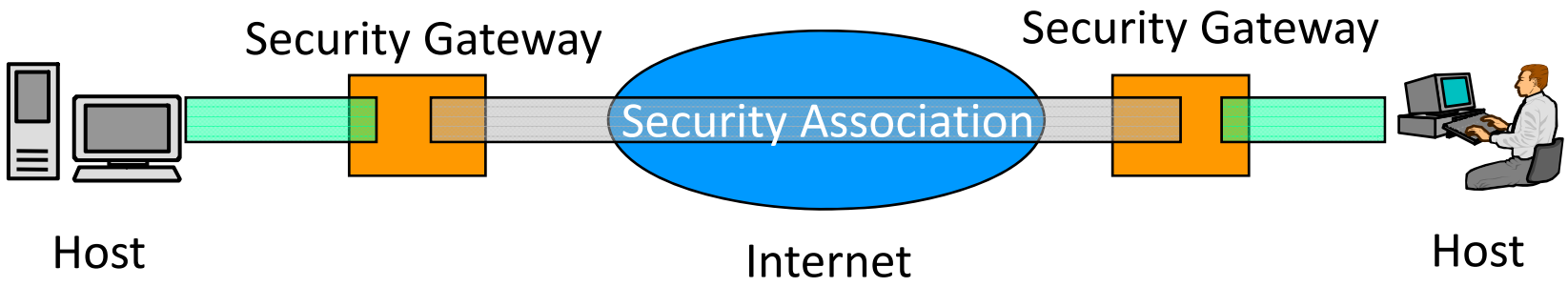


Security Association (SA)

- Defines a secure and unidirectional relationship
- Data structure containing the security parameters :
  - SPI (Security Parameter Index)
  - SNF (Sequence Number Field) used to avoid anti-replay
  - Anti-replay sequence number receive window
  - Authentication parameters (algorithms, keys, initialization vector)
  - Encryption parameters (algorithms, keys, length, initialization vector)
  - Key lifetime
  - SA lifetime
  - Protocol mode
- For a typical bi-directional communication, two SAs (one in each direction) are needed

Mechanisms: IPSec Security Associations

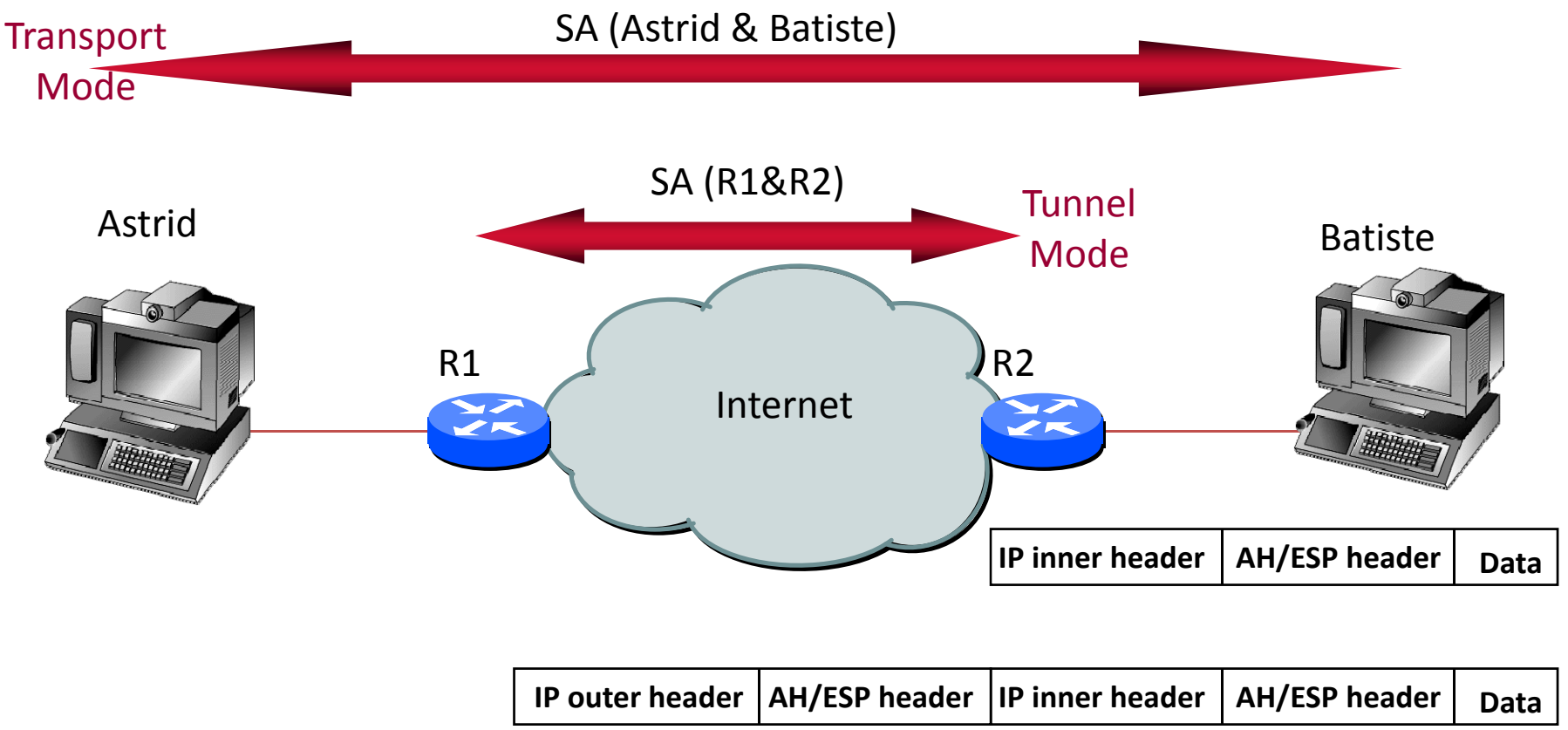
- A relationship between two or more entities that describes how the entities will use security services to communicate securely
- Simplex "connection" that affords security services to the traffic carried by it
- Bi-directional traffic requires one SA in each direction
- Security services provided by either AH or ESP
- If both AH and ESP required two SAs are formed
- Uniquely identified by
    - a SPI (Security Parameter Index)
    - IP destination address
    - Security Protocol Identifier (AH or ESP)



Security Association (SA)

- Agreement between two entities on a security policy, including:
  - Encryption algorithm
  - Authentication algorithm
  - Shared session keys
  - SA lifetime
- Unidirectional
  - Two-way communication consists of two Sas
- Key management
  - Manual mode
  - Automatic mode (via IKE)

Combining Security Associations



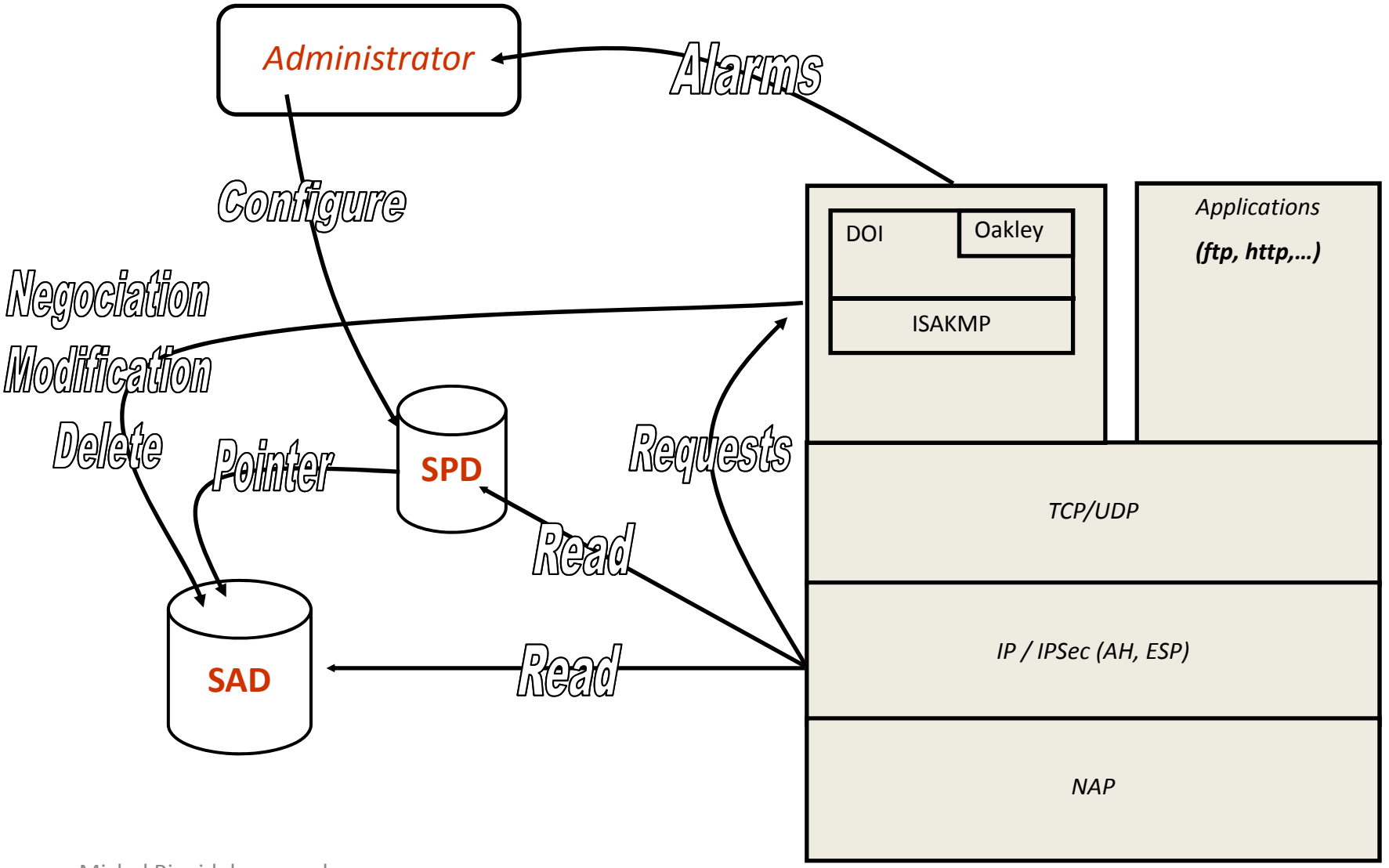
Security Policy Database (SPD)

- The SPD is the recipient for the system administrator's specification, of the security policies to be applied to outbound and inbound traffic
- The nominal form of the SPD includes for each entry :
  - The selectors that defines the traffic to which the policy should be applied
  - The security policy to be applied to the packet matching the associated selectors
- Each SPD entry specifies :
  - Drop
  - Bypass
  - Apply Ipsec (protocol & algorithm)
- Per interface, inbound and outbound SPDs

Security Association Database (SAD)

- The SAD contains the list of all inbound and outbound established SAs
- Each entry in the SAD defines the parameters associated with one SA. The entry is characterized by a set of values given to the field selectors. This defines the traffic flows to which the SA should be applied.
- For outbound processing, SAD entries are pointed to by entries in the SPD
- For inbound processing, each SAD entry is indexed by :
  - Outer header's destination IP address
  - IPSec protocol (AH or ESP) in the IP header (Protocol or Next Header fields)
  - SPI (Security Parameters Index) in the AH/ESP header : a 32-bit value used to distinguish among different SAs terminating at the same destination and using the same IPSec protocol

Mechanisms : Principle



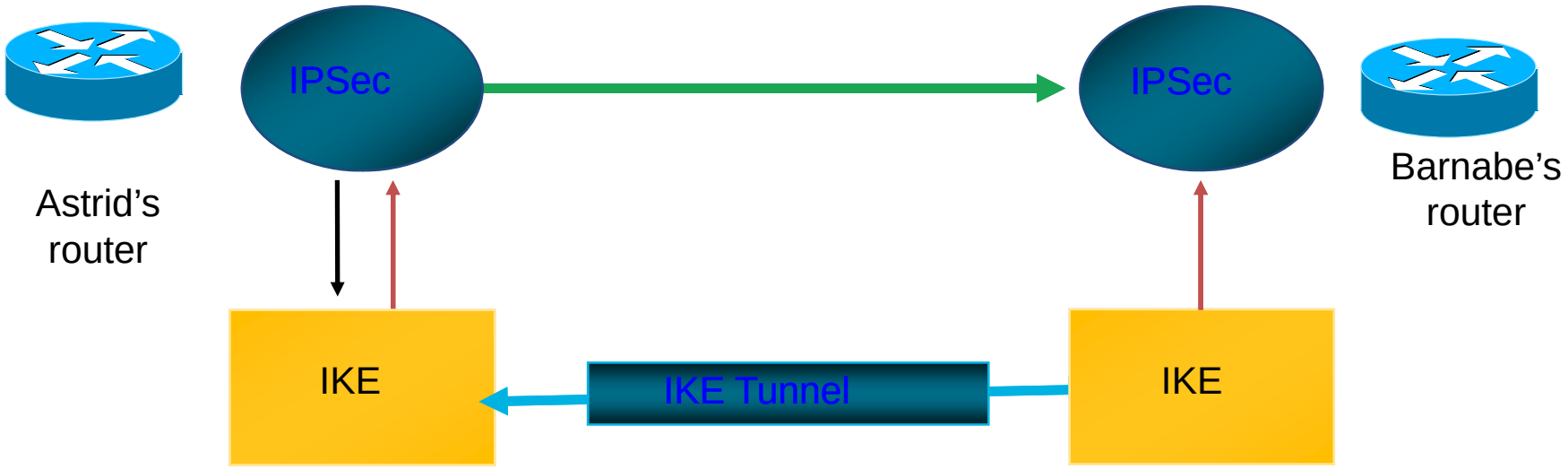
Internet Key Exchange (IKE)

- IKE protocol
  - Negotiates policy to protect communication
  - Authenticated Diffie-Hellman key exchange
  - Negotiates (possibly multiple) security associations for IPSec
  - Hybrid of three earlier protocols
    - ISAKMP (payload, syntax and encoding)
    - OAKLEY (based on Diffie-Hellman)
  - Objective : offer a secure and automated IPSec SA negotiation
  - Two phase
    - Establishment of a secure channel between the two peers
      - Called ISAKMP Security Association
      - Negotiation of the ISAKMP parameters (Authentication method, Algorithms used for encryption and authentication)
      - Key exchange
    - Ipsec negotiation inside the ISAKMP secure channel
      - Negotiation of the IPSec parameters : security protocols, algorithms and keys used for data authentication and encryption
- IKE Authentication
  - Signatures - Non-repudiable proof of communication
  - Encrypted nonce’s - Repudiable, deniable exchange
  - Pre-shared key

How IPSec Uses IKE

1. Outbound packet from Astrid to Barnabe. No IPSec SA

4. Packet is sent from Astrid to Barnabe protected by IPSec SA



2. Astrid's IKE begins negotiation with Barnabe's

3. Negotiation complete. Astrid and Barnabe now have complete set of SAs in place

IPSec and VPNs Architectures

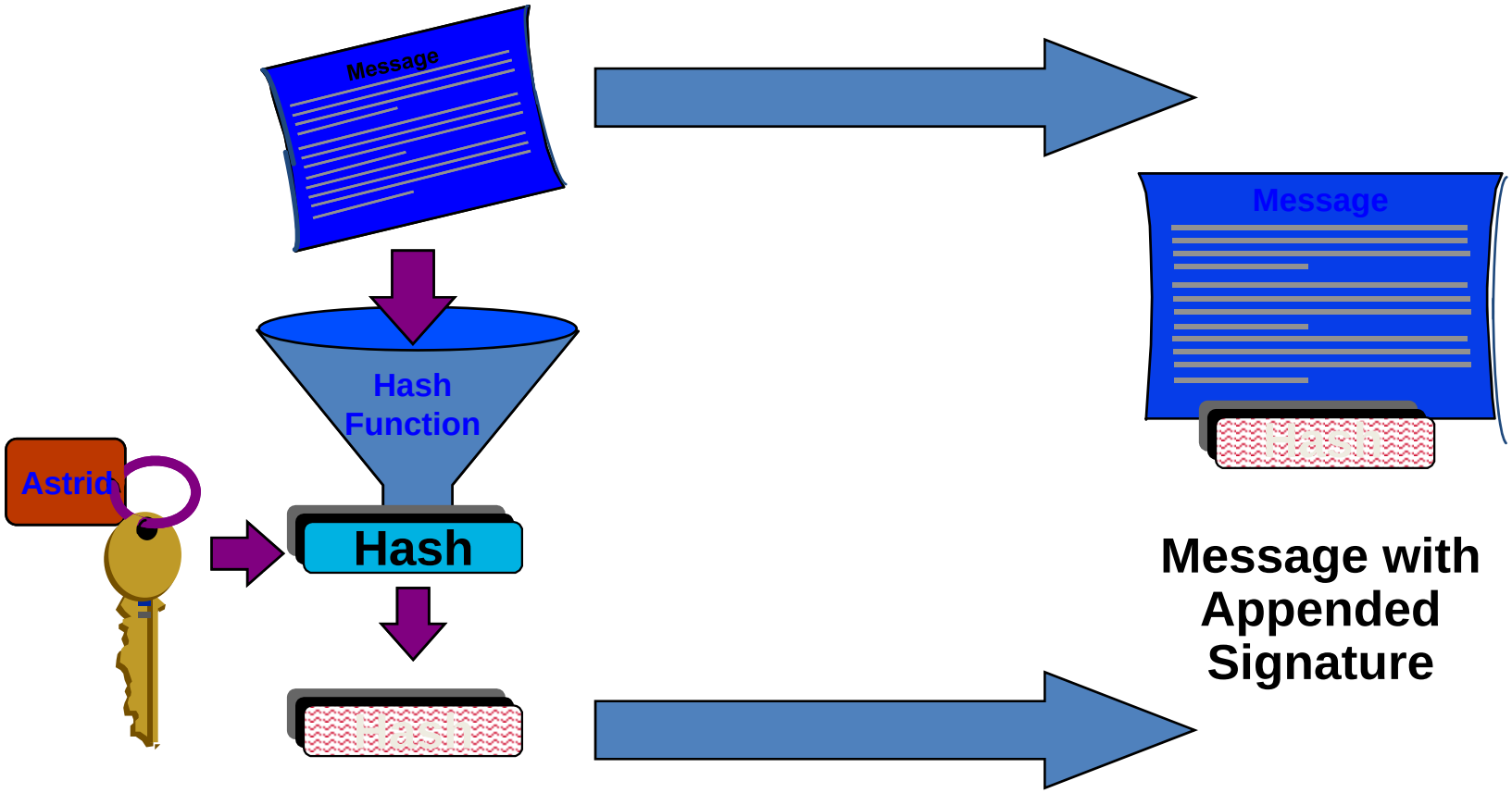
- Firewall to Firewall
  - Implement VPNs over the Internet.
  - Deployment already in progress; may some day largely replace private lines.
  - Caution: still vulnerable to denial of service attacks.
- Host to Firewall
  - Primary use: telecommuters dialing in.
  - Also usable for joint venture partners, clients, customers, etc.
  - But today's firewalls grant permissions based on IP addresses; they should use certificate names.
- Host to Host

  Attractive, but…
  - It's not widely available
  - Can we manage that many certificates?
  - Can servers afford it?
  - Can today's hosts protect their keys?

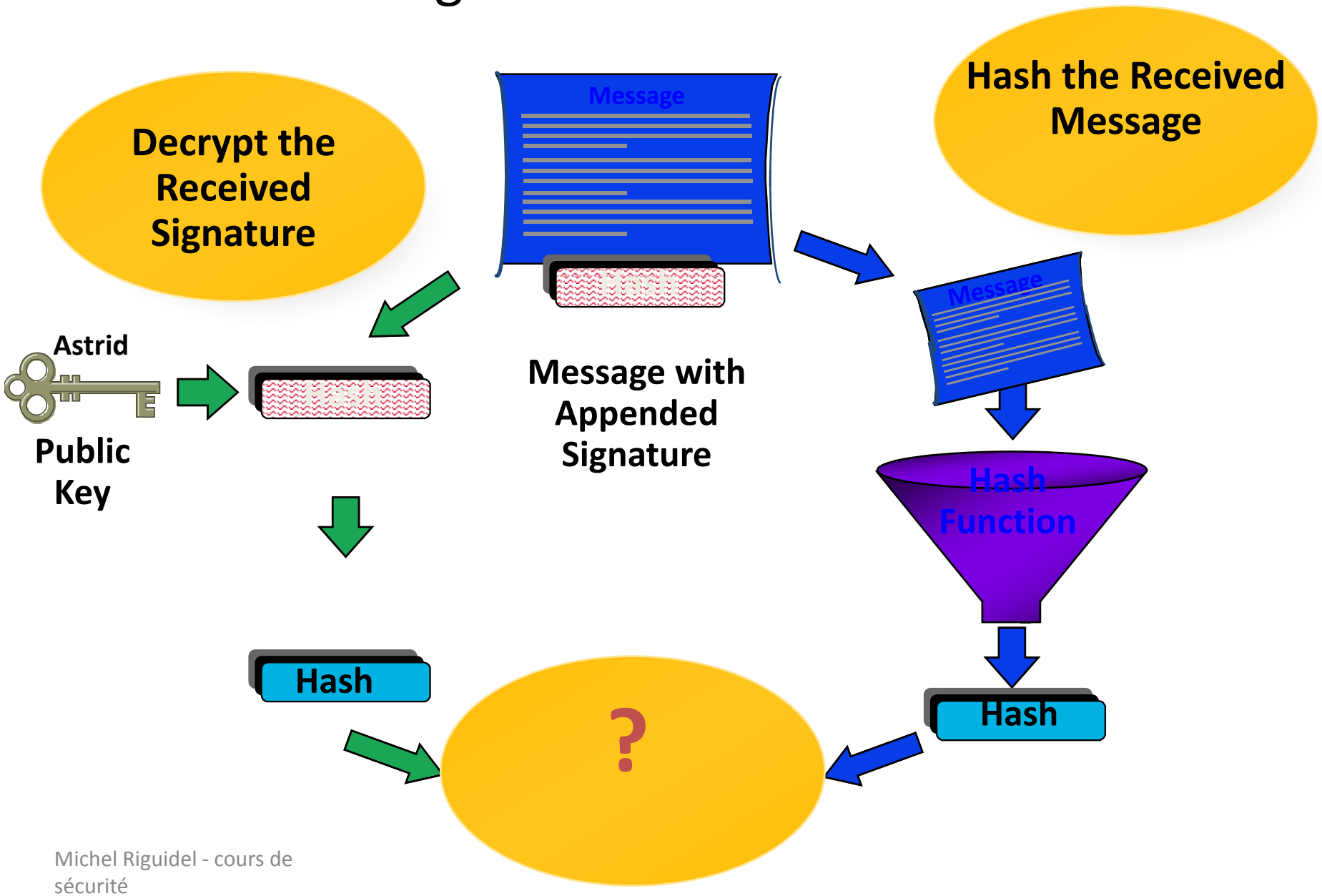
Conclusions

- Limits to IPSec
  - Encryption is not authentication; we must still control access.
    - Firewalls can't peek inside encrypted packets
  - Traffic engineers want to look inside packets, too.
  - New techniques for handling unusual links -- satellite hops, wireless LANs, constant bit rate ATM, etc. -- require examining, replaying, and tinkering with packets.
  - NAT boxes incompatible with end-to-end IPSec.
  - DHCP ?
  - Use key recovery technology?
- IPSec is a whole system which can answer needs of security and could be adapted in a lot of situations
- The implementation of IPSec in IPv6 and his efficient adaptation in IPv4 assures IPSec to become one of the major security solutions of the Internet and Intranet in the future
- but some improvements have to be done …
  - Treatment packet by packet
  - Interoperability
    - NAT, Dynamic allocation address, Multicast
    - all IPSec implementations
- Need of a centralized and dynamic management of security to be used at large scale

Digital Signature



Signature Verification



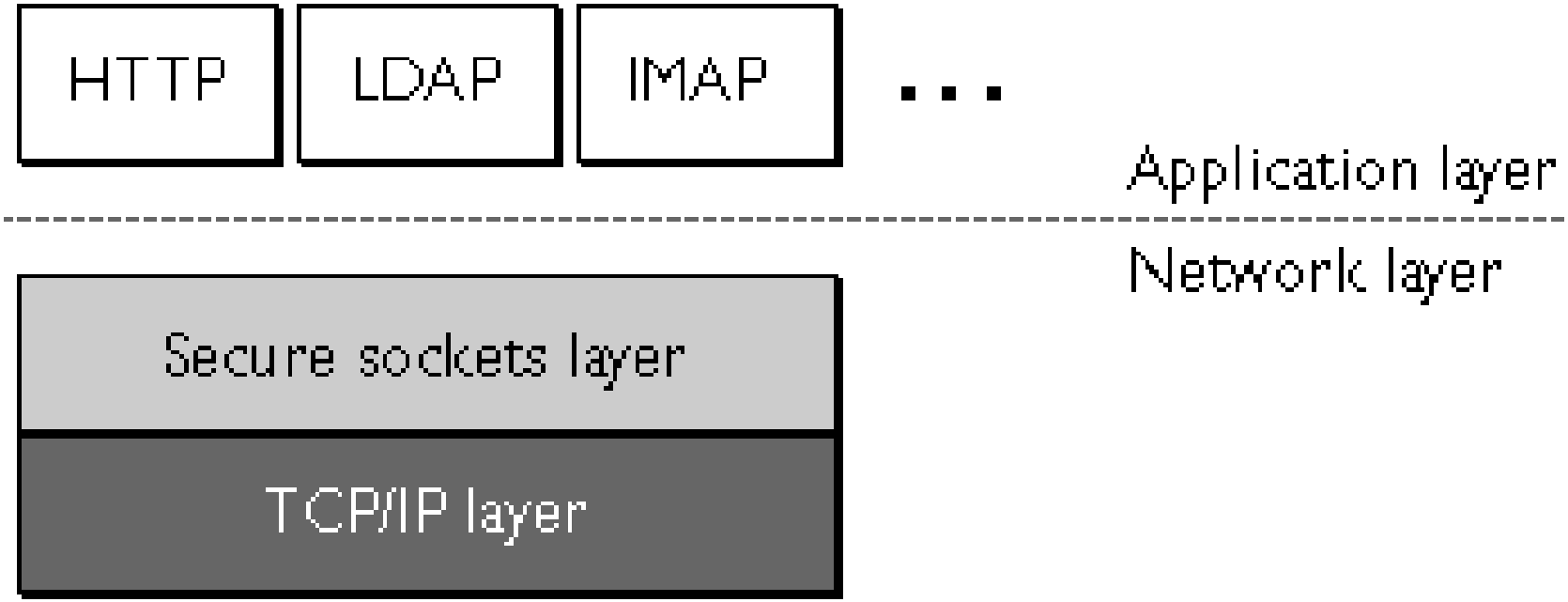
SSL : Présentation

- SSL : Secure Socket Layer
- Protocole de sécurité d'Internet pour les connexions point-à-point
- Développé par Netscape pour garantir la sécurité de la transmission de données sur Internet
- Version actuelle : SSL 3.0
- Fournit une connexion sécurisée entre le client et le serveur
- Protocole entre TCP et les protocoles applicatifs
- 2001 : l'IETF rachète le brevet de SSL à Netscape et le rebaptise TLS (Transport Level Security, RFC 2246) (version 1.0)

SSL : Services de sécurité

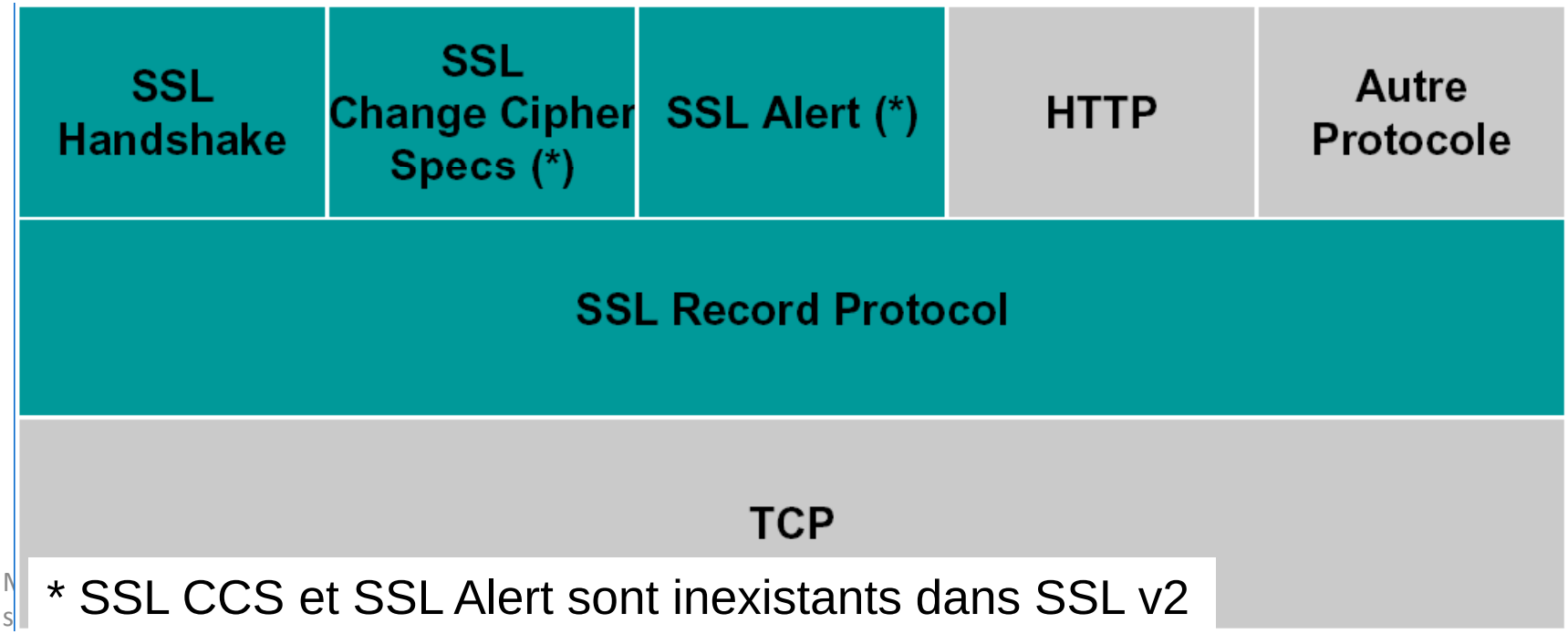
- Authentification :
  - Serveur (obligatoire) : pour confirmer son identité
  - Client (optionnel) : nécessaire quand par exemple le serveur est une banque
  - Utilisation de certificat X509 v3
  - Utilisation d'algorithme de chiffrement à clé publique pour vérifier le certificat
  - Se fait à l'établissement de la session
- Confidentialité :
  - Algorithme de chiffrement symétrique
  - Clé générée à l'établissement de la session
- Intégrité :
  - Fonction de hachage avec une clé secrète qui génère une empreinte appelée MAC (Message Authentication Code)
- Non rejeu :
  - Numéro de séquence

SSL : Architecture



SSL : protocoles
Pile protocolaire de SSL v3

- Consiste en deux niveaux de protocoles
- Le protocole Record
- 3 protocoles de niveau supérieur :
  - Handshake Protocol
  - Change Cipher Spec Protocol
  - Alert Protocol



* SSL CCS et SSL Alert sont inexistants dans SSL v2

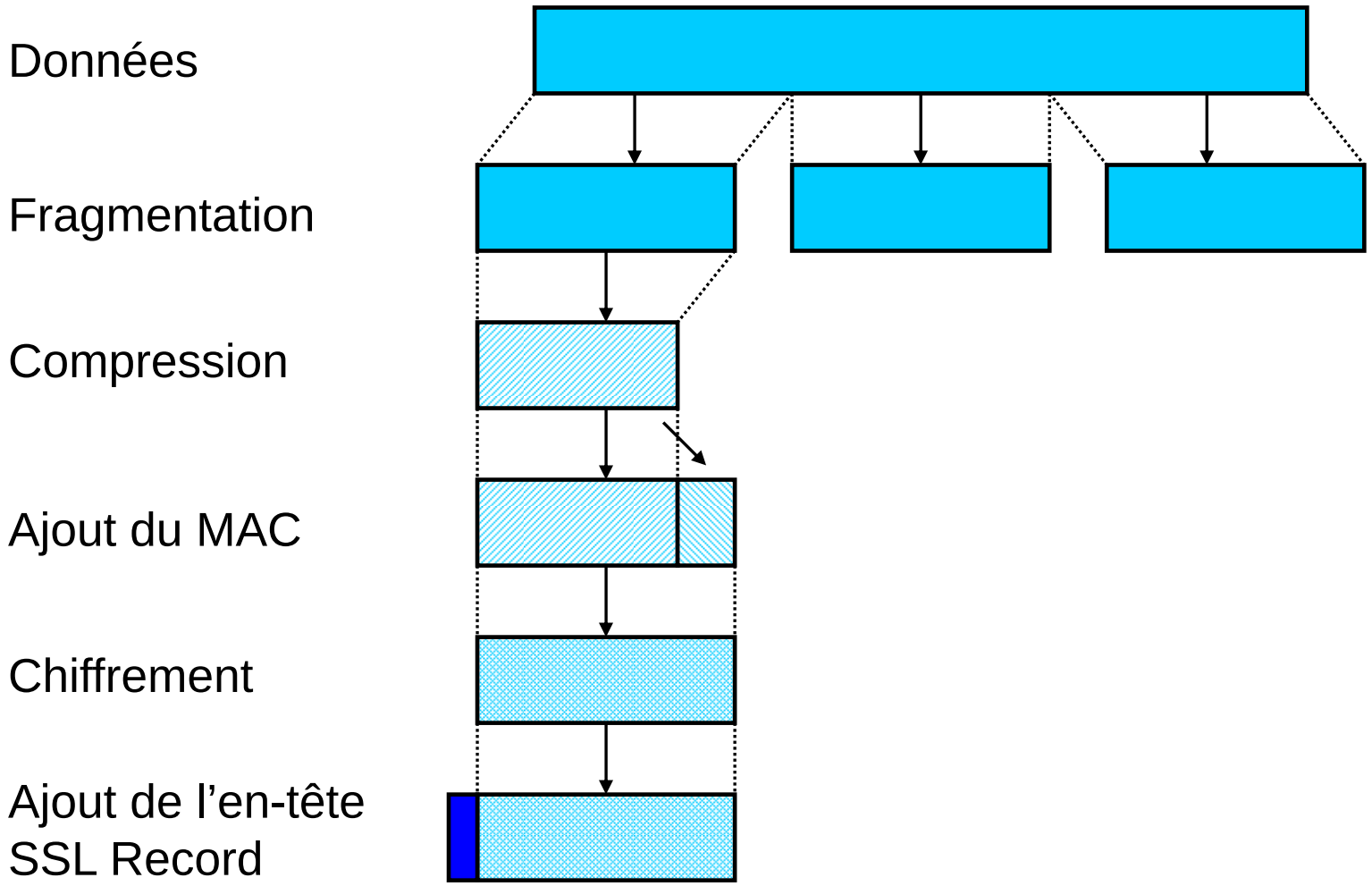
SSL : Session / Connexion

- Session :
  - Association entre un serveur et un client
  - Créée par le protocole Handshake
  - Définit un ensemble de paramètres cryptographiques de sécurité qui peuvent être utilisés pour plusieurs connexions (évite une négociation de paramètres de sécurité à chaque connexion)
- Connexion :
  - Relation de type point à point
  - Chaque connexion est associée à une session

SSL Record Protocol

- Ce protocole fournit 2 services à une connexion SSL :
  - Confidentialité : définit une clé secrète pour le chiffrement
  - Intégrité du message : définit une clé secrète pour le calcul de l'empreinte
- SSL Record Protocol : opérations
  - Fragmentation :
    - Le message est fragmenté en blocs de taille maximum 2^{14} octets
  - Compression :
    - Cette opération est prévue dans les spécifications mais non implémentée
  - Calcul du MAC :
    - Utilise la clé secrète
    - Utilise l'algorithme SHA-1 ou MD5
  - Chiffrement :
    - Le message + MAC sont chiffrés avec un chiffrement symétrique
  - Ajout de l'en-tête :
    - 5 octets, composée de longueur du message, version, etc.

SSL Record Protocol : fonctionnement



ChangeCipherSpec (CCS)

- CCS signale au Record toute modification des paramètres de sécurité

- Constitué d'un message (1 octet)

Alert Protocol

- Peut être invoqué par :
  - l'application (pour signaler la fin d'une connexion)
  - le protocole Handshake (suite à un problème survenu au cours de la négociation)

SSL : algorithmes de chiffrement symétriques

- DES
- 3DES
- DES-40
- Fortezza
- IDEA
- RC2-40
- RC4-40
- RC4-128

SSL Handshake Protocol

- Protocole de négociation
- Basé sur une série de messages échangés entre le client et le serveur qui permet :
  - d'authentifier chacun auprès de l'autre
  - de déterminer les spécifications de chiffrement ; i.e. les algorithmes de chiffrement, de hachage et d'échange de clés
  - d'échanger et vérifier les certificats du serveur et/ou du client
  - générer les clés : une clé de session maître et en dérive une clé de chiffrement pour le serveur, une pour le client et des clés pour sécuriser les MAC
- 4 phases pour les messages
- Chaque message a 3 champs

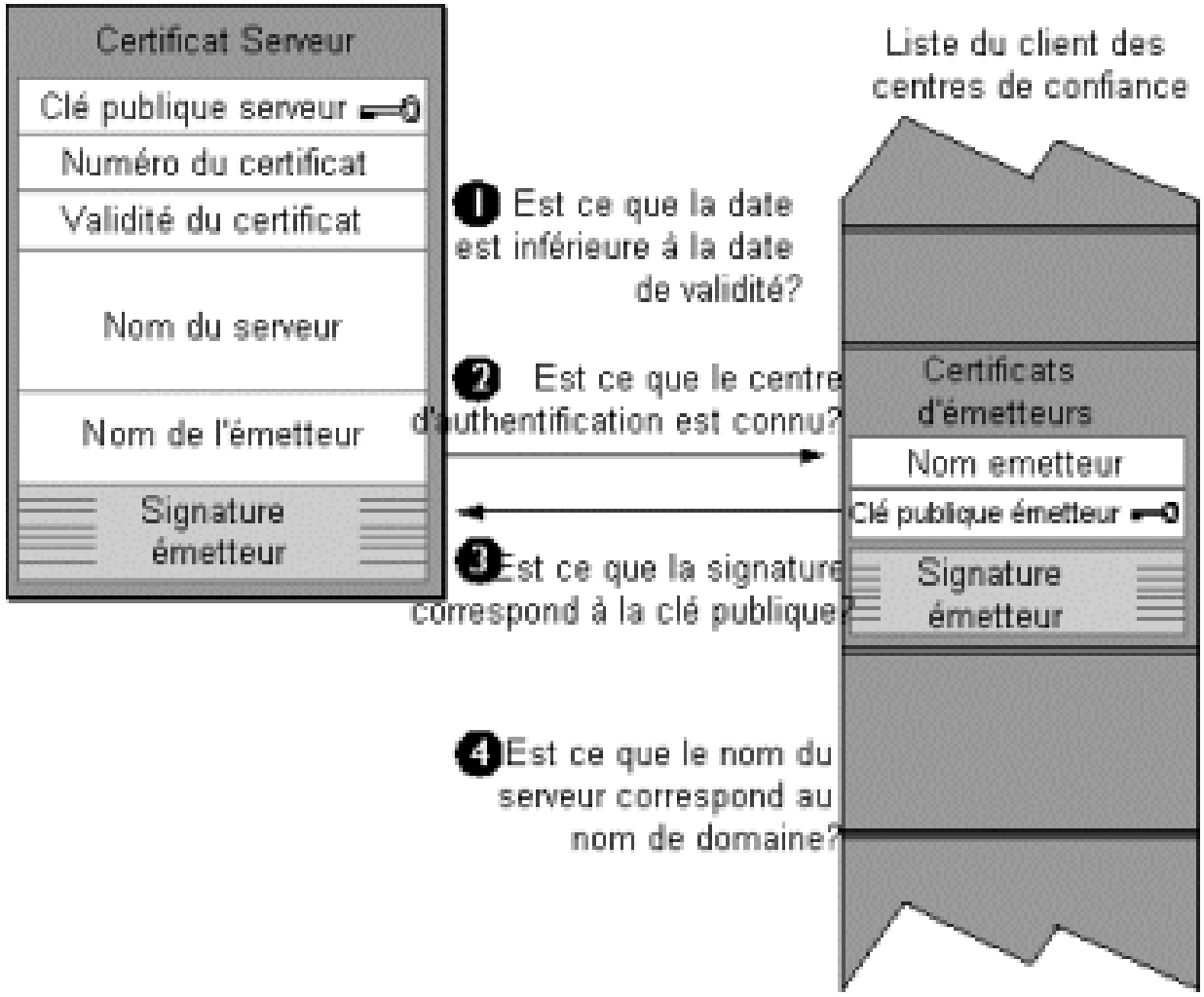
Les 4 phases

- Phase 1 : établir les paramètres de sécurité
  - Processus initié par le client et le serveur répond
  - Le client envoie des informations concernant la version SSL, l'identificateur de la session, une liste de algorithmes cryptographiques supportés
  - Négocie les paramètres de chiffrement :
    - La méthode d'échange de clés (RSA, Diffie-Hellman, Fortezza)
    - L'algorithme de chiffrement symétrique
    - La fonction de hachage pour créer le MAC (MD5, SHA-1)
- Phase 2 : authentification du serveur et échange de clés
  - Le serveur envoie au client :
    - son certificat
    - la clé secrète en utilisant un des algorithmes à clé publique
  - Les paramètres envoyés par le client sont inclus dans ce message pour contrer les attaques de rejeu
  - Le serveur peut demander un certificat au client
  - Les schémas de signature supportés sont :RSA, DSS, Fortezza
- Phase 3 : authentification du client et échange de clés
  - Le client envoie au serveur :
    - le certificat si le serveur l'a demandé
    - la clé secrète
  - Si c'est le RSA qui est utilisé pour l'échange de clés :
    - Le client génère une pré clé secrète (48 octets) et l'envoie au serveur
  - Si c'est Diffie-Hellman, les paramètres publics sont envoyés au serveur
- Phase 4 : fin de la négociation
  - Le client envoie un message final pour confirmer que l'échange de clé et le processus d'authentification sont réussis
  - Le serveur envoie également un message final

SSL Handshake Protocol

Change Cipher	*Handshake Protocol*	c ↔ s	*Informations échangées*
Specs	**ClientHello**	→	version, **nbre aléatoire (256 bits), ident session, liste de spécifications de chiffrement**
	ServerHello	←	version, nombre aléatoire (256 bits), ident session, spécification de chiffrement
	Certificate	←	certificat du serveur, chaîne de CA
	Certificate Request	←	types de certificats acceptables, liste de CA autorisés.
	ServerHeloDone	←	
	Certificate	→	certificat du client, chaîne de CA.
	ClientKeyExchange	→	version + 46 octets aléatoires chiffrés avec la clé publique du serveur
	CertificateVerify	→	empreinte clé maître + traces des échanges précédents.
Change →	**Finished**	→	empreinte clé maître + traces des échanges précédents.
Change ←	**Finished**	←	empreinte clé maître + traces des échanges précédents

Authentification du serveur



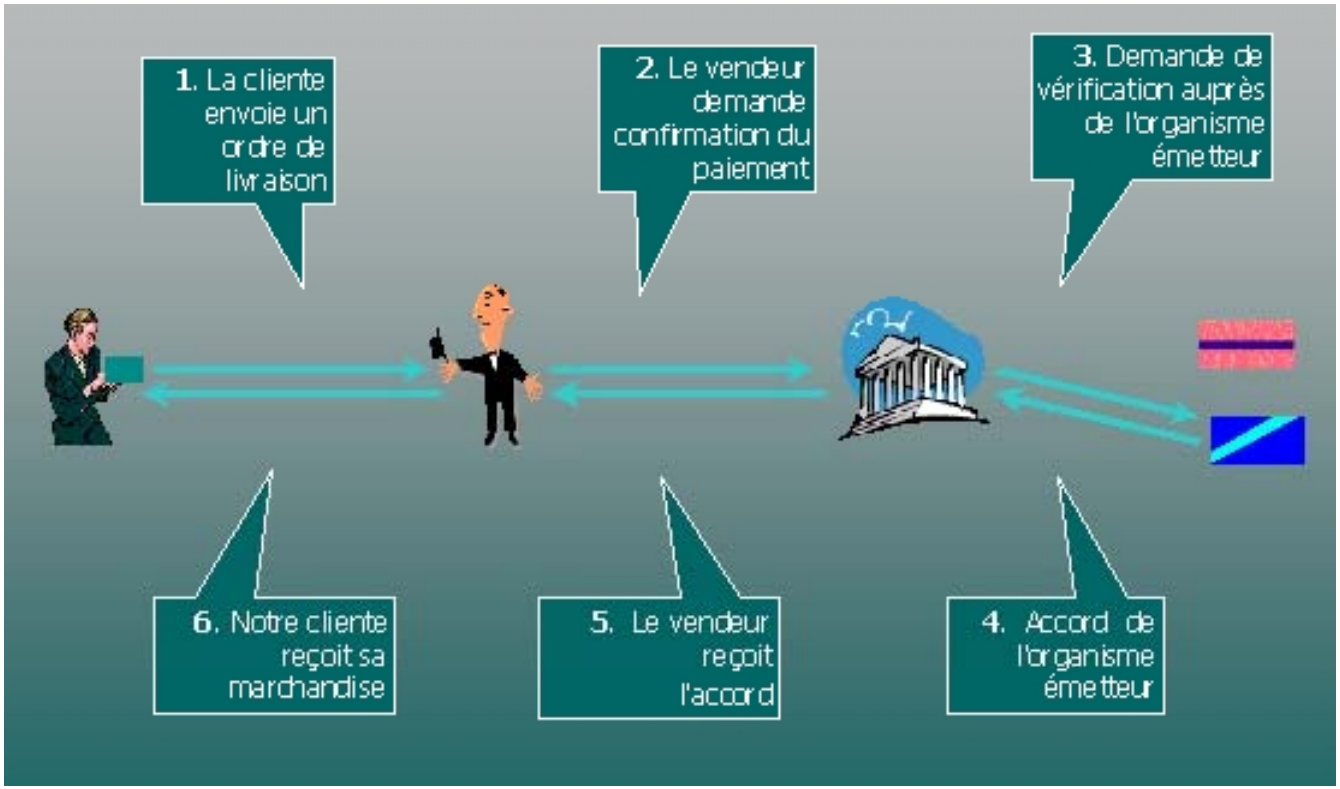
SET : présentation

- Secure Electronic Transaction
- Protocole de paiement développé par MasterCard et Visa (1997)
- SET propose un ensemble de protocoles de sécurité pour le paiement par carte bancaire sur Internet
- Il vérifie l'identité du client, du vendeur et de l'institution financière
- Il sécurise l'échange entre les différentes entités
- Il nécessite la possession d'une carte bancaire

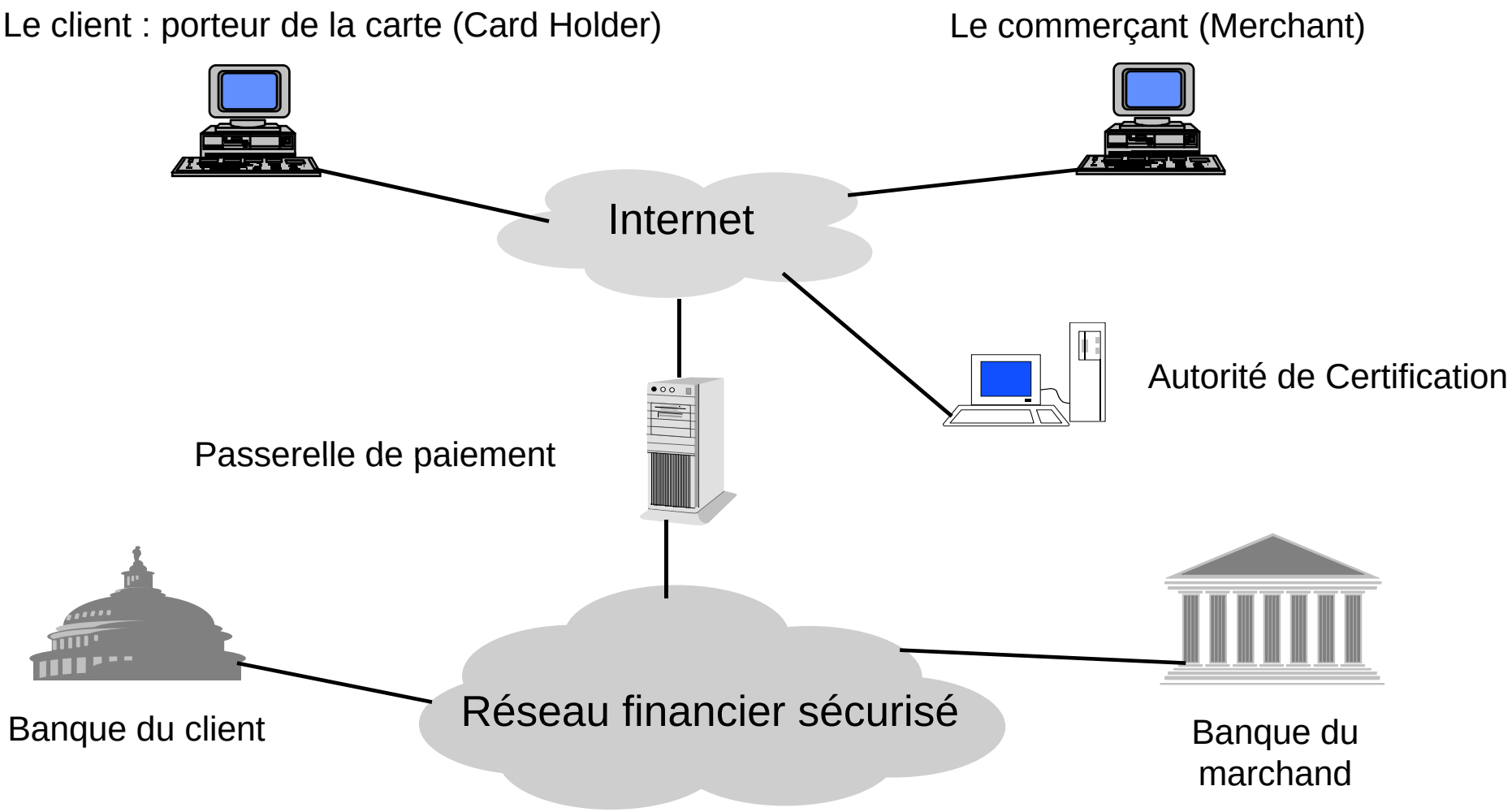
Les protocoles SET

- 3 protocoles essentiels
  - Le protocole d'achat
  - Le protocole d'autorisation de paiement
  - Le protocole de paiement

SET : principe



SET : architecture et acteurs



Séquence des opérations

- Le client obtient un compte et une carte de crédit
  - Certificat signé par l'institution émettrice
- Le marchand doit posséder
  - Deux certificats à clés publiques : signature et échange de clés
- Le client, après commande et réception du bon de commande, vérifie la légitimité du marchand
  - Le marchand envoie une copie de son certificat au client
- Le client envoie l'information de commande et l'information de paiement accompagnées de son certificat
- Le marchand demande l'autorisation de paiement
  - Vérification de la légitimité et de la solvabilité du client
- Le marchand confirme la commande au client
- Le marchand exécute la commande et fait une demande de paiement

Les services de SET

- Authentification :
  - Utilise les certificats numériques pour authentifier les deux parties effectuant la transaction
- Confidentialité :
  - Utilise la cryptographie à clé publique (1024 bits)
- Intégrité :
  - Signe les messages
- Non répudiation

Algorithmes SET

- DES : chiffrement /déchiffrement pour assurer la confidentialité
- RSA : clé de signature et chiffrement pour assurer l'authentification, l'identification et l'intégrité
- SHA-1 : génération d'une empreinte pour assurer l'intégrité
- HMAC-SHA-1 : générer d'une empreinte pour assurer l'intégrité

Comparaisons SET/SSL

- SET : Couche 7
  - Vérification lourde des certificats (algorithmes symétriques)
  - Calcul cryptographique lourd

- SSL : Couches au dessus du Transport (4)
  - Une authentification au début de la session
  - Certificat optionnel pour le client