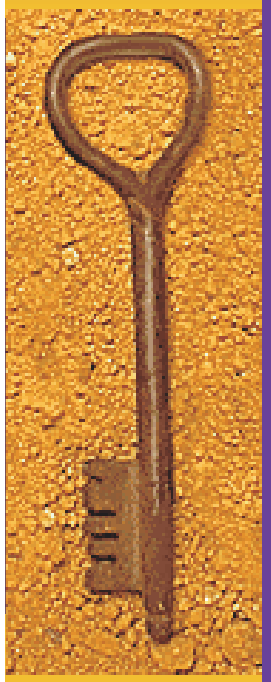
IPsec: Security Across the Protocol Stack

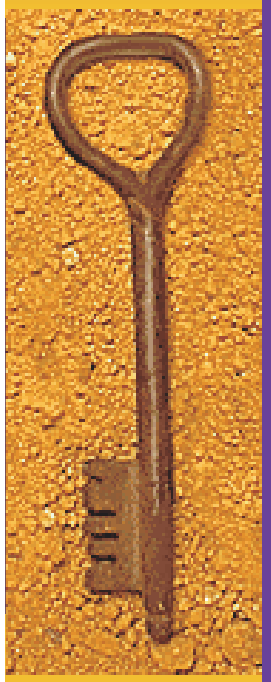
Brad Stephenson

CSCI NetProg



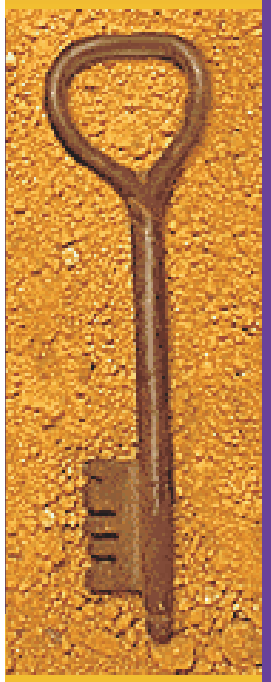
Network Security

- There are application specific security mechanisms (eg. S/MIME, PGP, Kerberos, SSL/HTTPS)
- But there are security concerns that cut across protocol layers
- Can we implement security in the network for all applications?



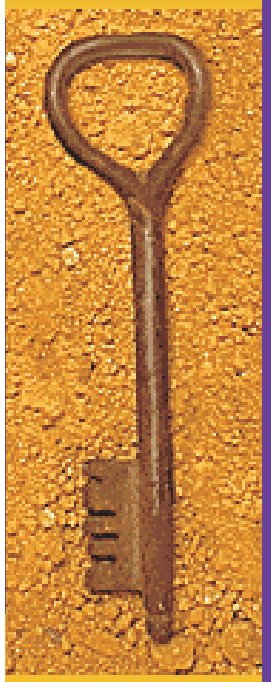
What is IPsec?

- A collection of tools and algorithms (protocols)
- General IP security mechanisms
- It provides
  - authentication
  - confidentiality
  - key management



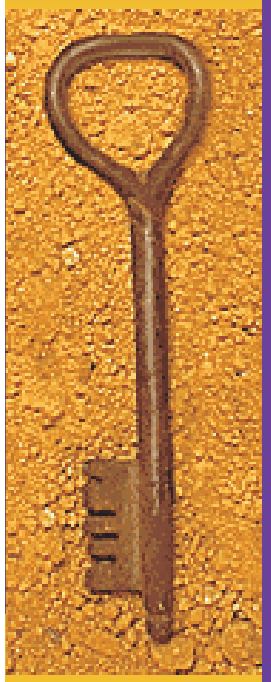
Services Provided by IPsec

- **Authentication** – ensure the identity of an entity
- **Confidentiality** – protection of data from unauthorized disclosure
- **Key Management** – generation, exchange, storage, safeguarding, etc. of keys in a public key cryptosystem



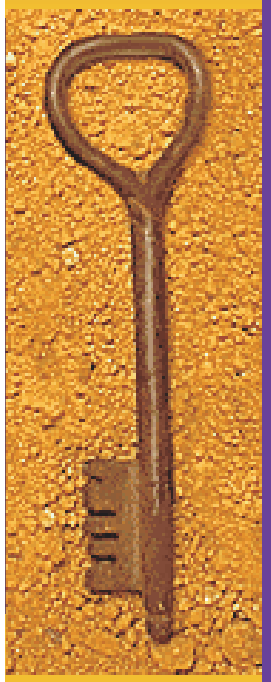
IPsec Services (detailed)

- Access control
- Connectionless integrity
- Data origin authentication
- Rejection of replayed packets
- Confidentiality (via encryption)
- Some traffic flow confidentiality (firewall to firewall)



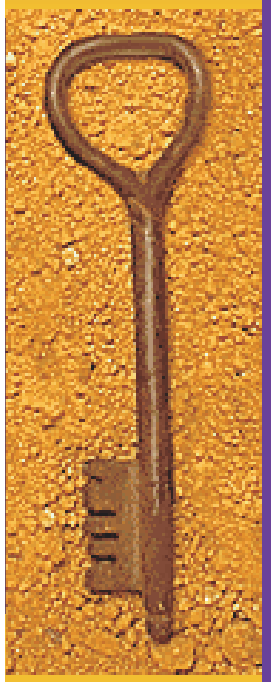
Benefits of IPsec

- If implemented in a firewall or router, provides strong security to all traffic crossing the perimeter
- Resides below the transport layer, hence transparent to application layer
- Can be transparent to end users
- *Note:* Mandatory for IPv6 implementations



AH and ESP

- Authentication Header (AH) provides:
  - Data integrity
  - Authentication of IP packets
  - Prevents replay attacks
- Encapsulating Security Payload (ESP):
  - Data confidentiality
  - Some traffic flow confidentiality
  - Authentication services of AH (optional)



Authentication Header (AH)

- Provides support for data integrity & authentication of IP packets
  - end system/router can authenticate user/app
  - prevents address spoofing attacks by tracking sequence numbers
- Based on use of a MAC
  - HMAC-MD5-96 or HMAC-SHA-1-96
- Parties must share a secret key

Authentication Header

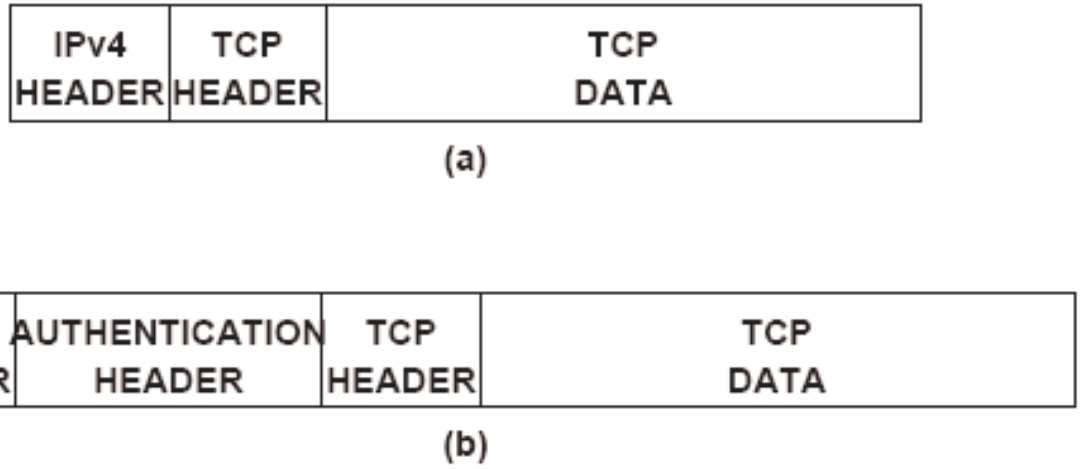


Figure 32.1, D. Comer

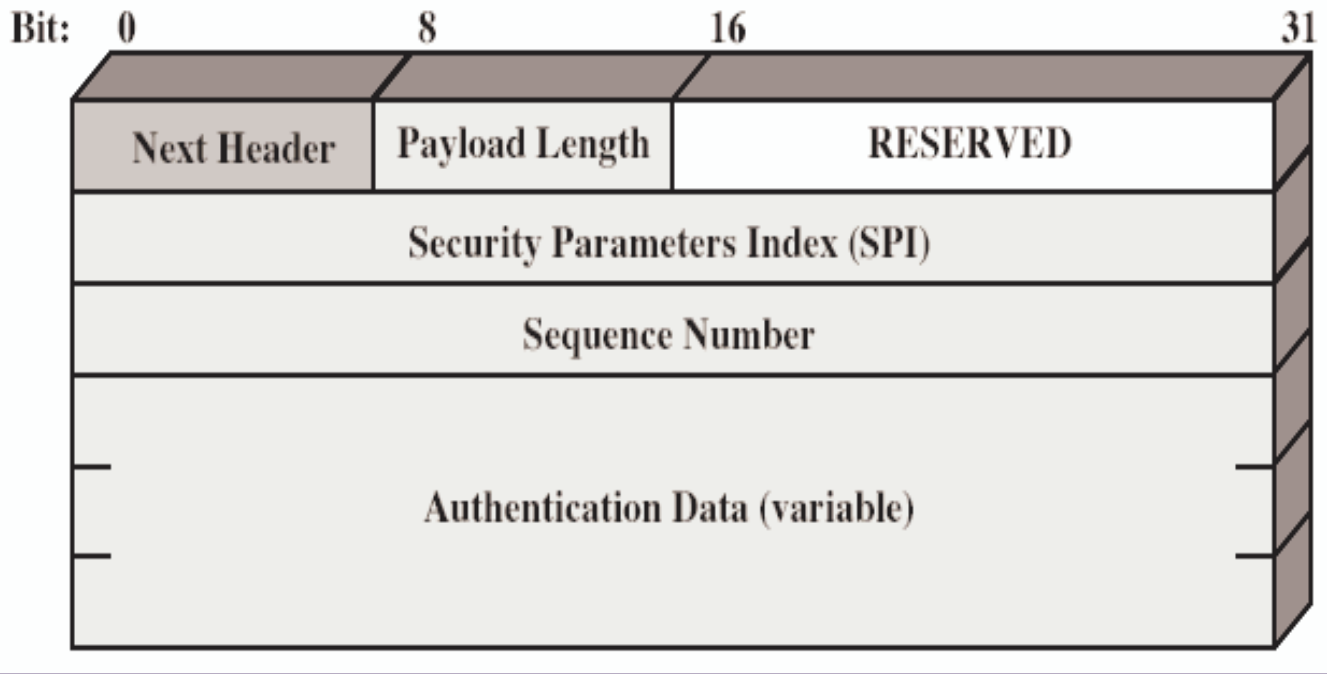
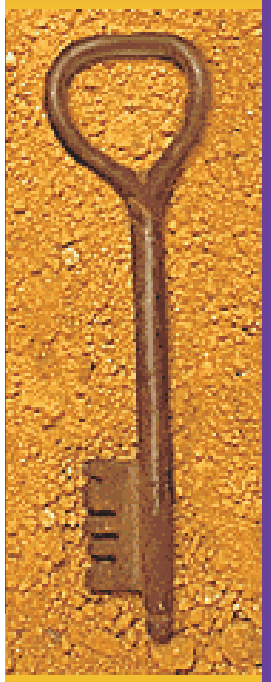


Figure 16-3, W. Stallings



Encapsulating Security Payload (ESP)

- Provides message content confidentiality & limited traffic flow confidentiality
- Can optionally provide the same authentication services as AH
- Supports many ciphers, modes, padding
  - DES, Triple-DES, RC5, IDEA, CAST, others

Encapsulating Security Payload

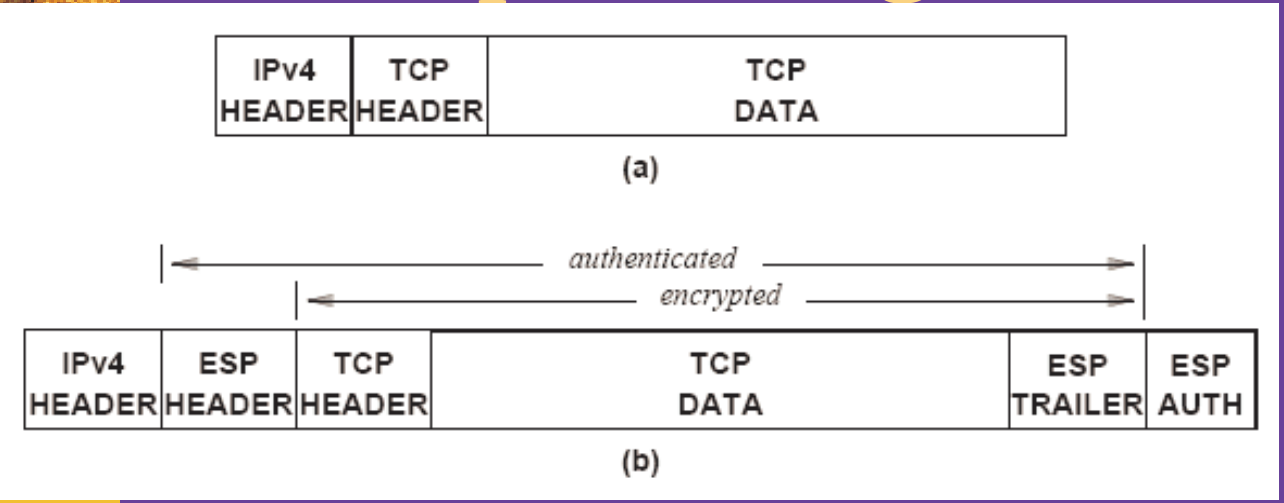


Figure 32.3, D. Comer

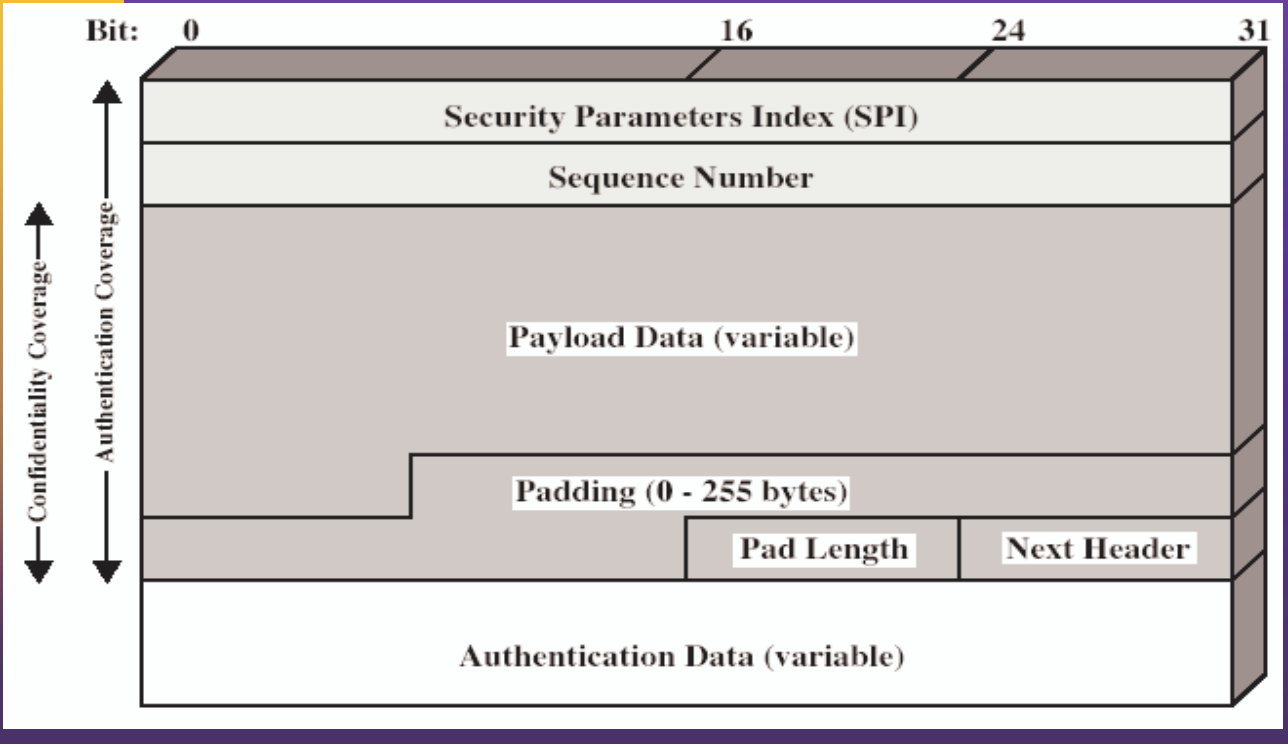
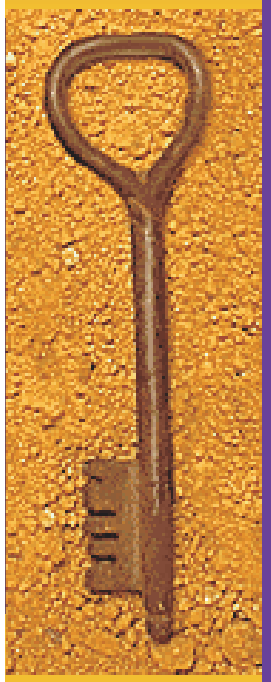
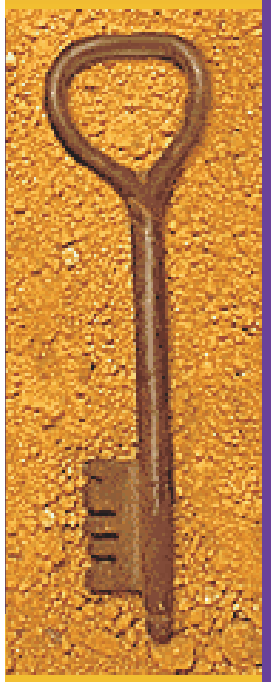


Figure 16-7, W. Stallings



Security Associations (SAs)

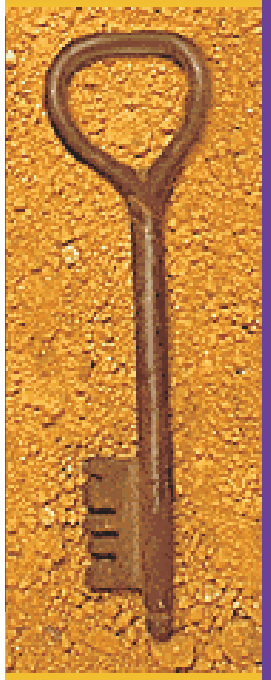
- A *one-way* relationship between sender & receiver that affords security for traffic flow
- Defined by 3 parameters:
  - Security Parameters Index (local identifier)
  - IP Destination Address
  - Security Protocol Identifier (AH or ESP)
- *Each implementation of IPsec must keep a database of SAs*

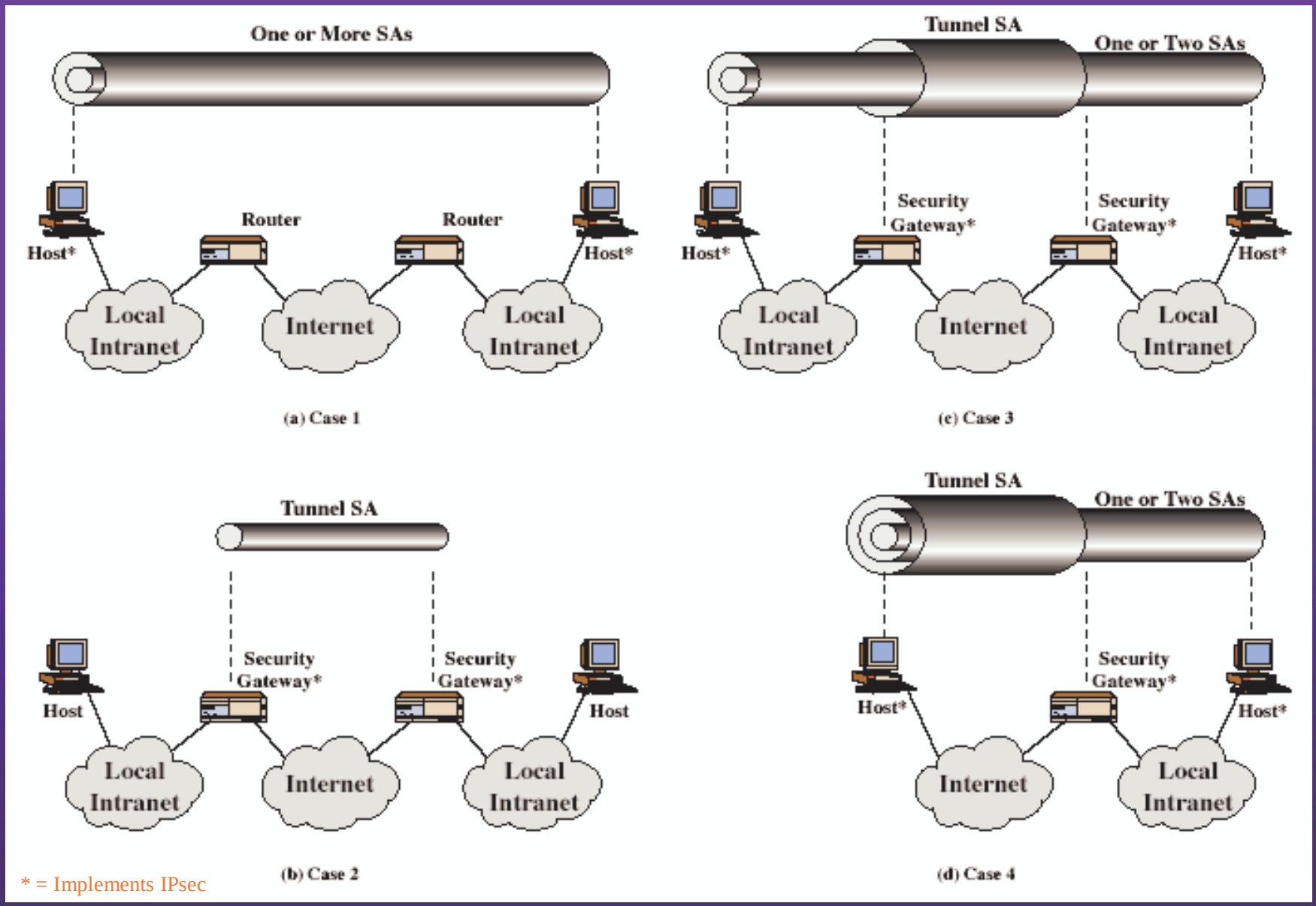


Combining Security Associations

- SAs can implement either AH or ESP
- To implement both need to combine SAs into a security bundle

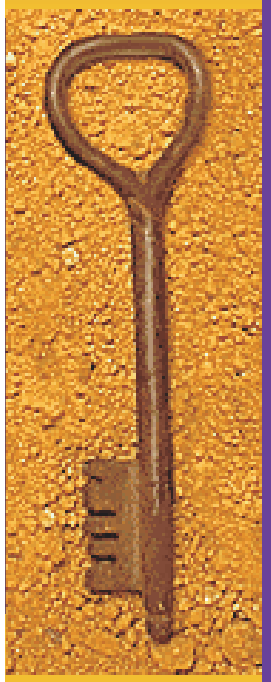
Combining Security Associations





* = Implements IPsec

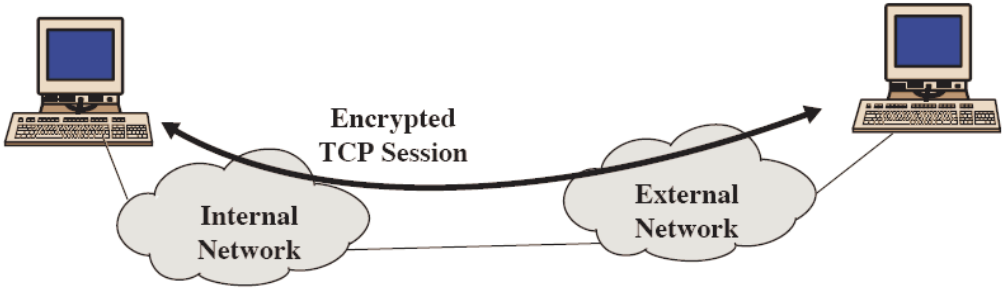
Figure 16-10, W. Stallings



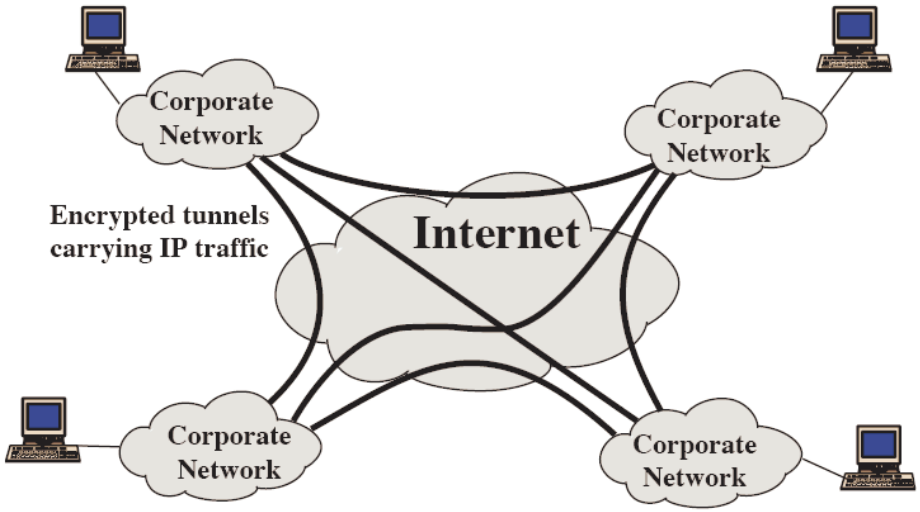
Transport vs. Tunnel Mode

- Transport mode
  - data protected but header left in clear
  - can do traffic analysis but is efficient
  - good for ESP host to host traffic
- Tunnel mode
  - add new header for next hop
  - hides end-host IP addresses through insecure networks
  - good for VPNs, gateway to gateway security

Transport & Tunnel Modes

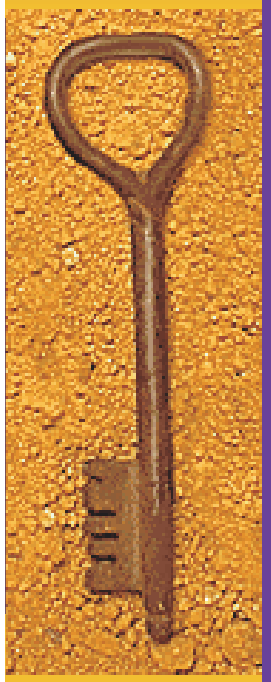


(a) Transport-level security



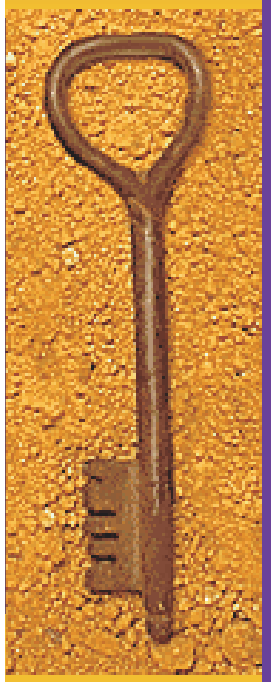
(b) A virtual private network via Tunnel Mode

Figure 16-8, W. Stallings



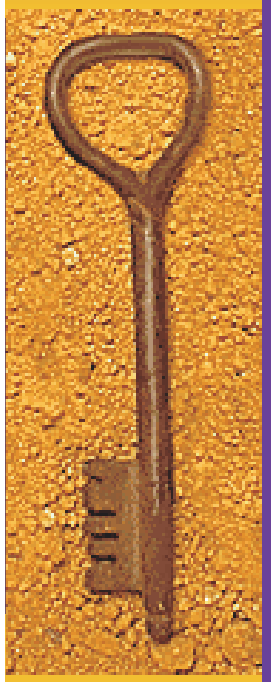
So you wanna try it?

- Implemented in OS kernel
- Non-trivial to understand



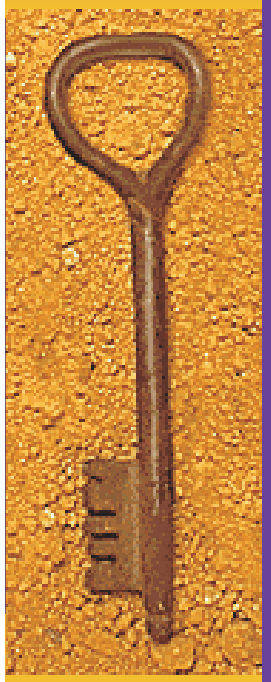
So you wanna try it?

- Linux
  - racoon
  - openswan (openswan.org)
  - Free S/WAN (freeswan.org)
- Unix
  - man **ipsec**
- Windows
  - mmc (Microsoft Management Console)



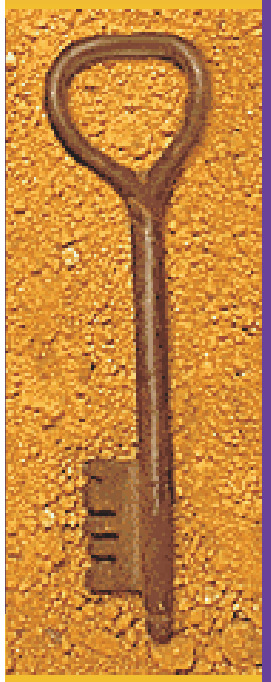
Linux

- Must specify a security policy in kernel
  - Who do you trust?
- racoon
  - Key management daemon
- Free S/WAN
  - IPsec implementation for Linux
- openswan
  - Another IPsec implementation for Linux



Unix

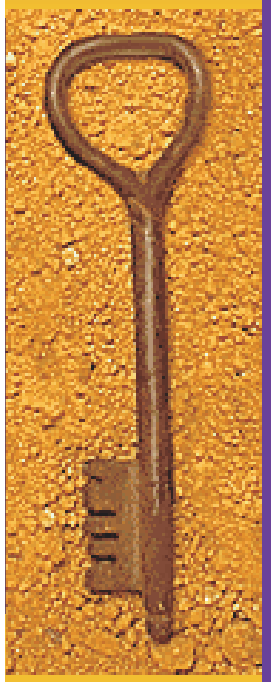
- IPsec policy is enforced in the **ip(7P)** driver for system-wide policy
- Use **ndd** to alter /dev/ip at the system level
- Or specify per-socket options



Unix Socket Options

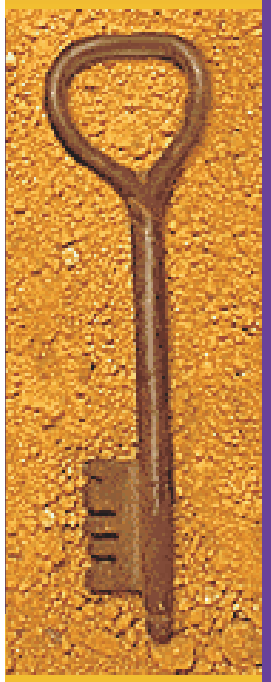
```
#include <sys/socket.h>
#include <netinet/in.h>
#include <net/pfkeyv2.h>

/* .... socket setup */
rc = setsockopt(sock, IPPROTO_IP, IP_SEC_OPT,
    (const char *)&ipsec_req, sizeof (ipsec_req_t));
```



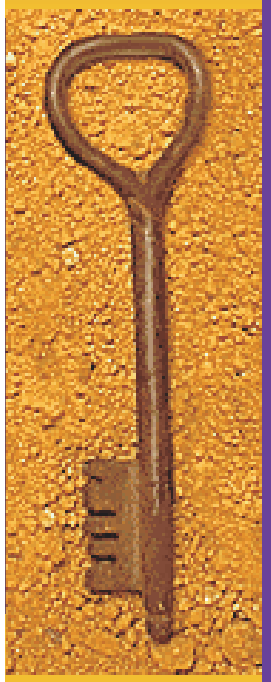
ipsec_req

```
typedef struct ipsec_req {
uint_t ipsr_ah_req; /* AH request */
uint_t ipsr_esp_req; /* ESP request */
uint_t ipsr_self_encap_req; /* Self-Encap request */
uint8_t ipsr_auth_alg; /* Auth algs for AH */
uint8_t ipsr_esp_alg; /* Encr algs for ESP */
uint8_t ipsr_esp_auth_alg; /* Auth algs for ESP */
} ipsec_req_t;
```



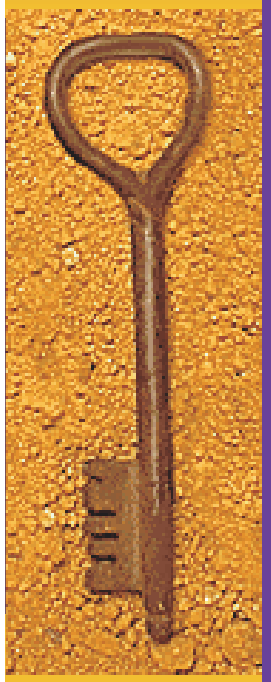
Windows XP

- Type **mmc** at a command line
- Add snap-in IPsec Policy
- Edit the policy as you see fit



Summary

- IPsec is a collection of protocols that provide low-level network security
- Last specification was in 1998, currently being revised as Internet Draft
- Required for IPv6
- Currently the most popular use is for implementing VPNs



References

- *RFC 2401* “Security Architecture for the Internet Protocol”
- *Internet Draft*, Dec 2004, “Security Architecture for the Internet Protocol”
- *Cryptography and Network Security*, W. Stallings, Chap. 16 “IP Security”
- *Internetworking with TCP/IP Vol. 1*, D. Comer, Chap. 32 “Internet Security”

ERROR: undefined
OFFENDING COMMAND:

STACK: