IPSec Overhead in Wireline and Wireless Networks

Hafeth Hourani

hafeth.hourani@nokia.com

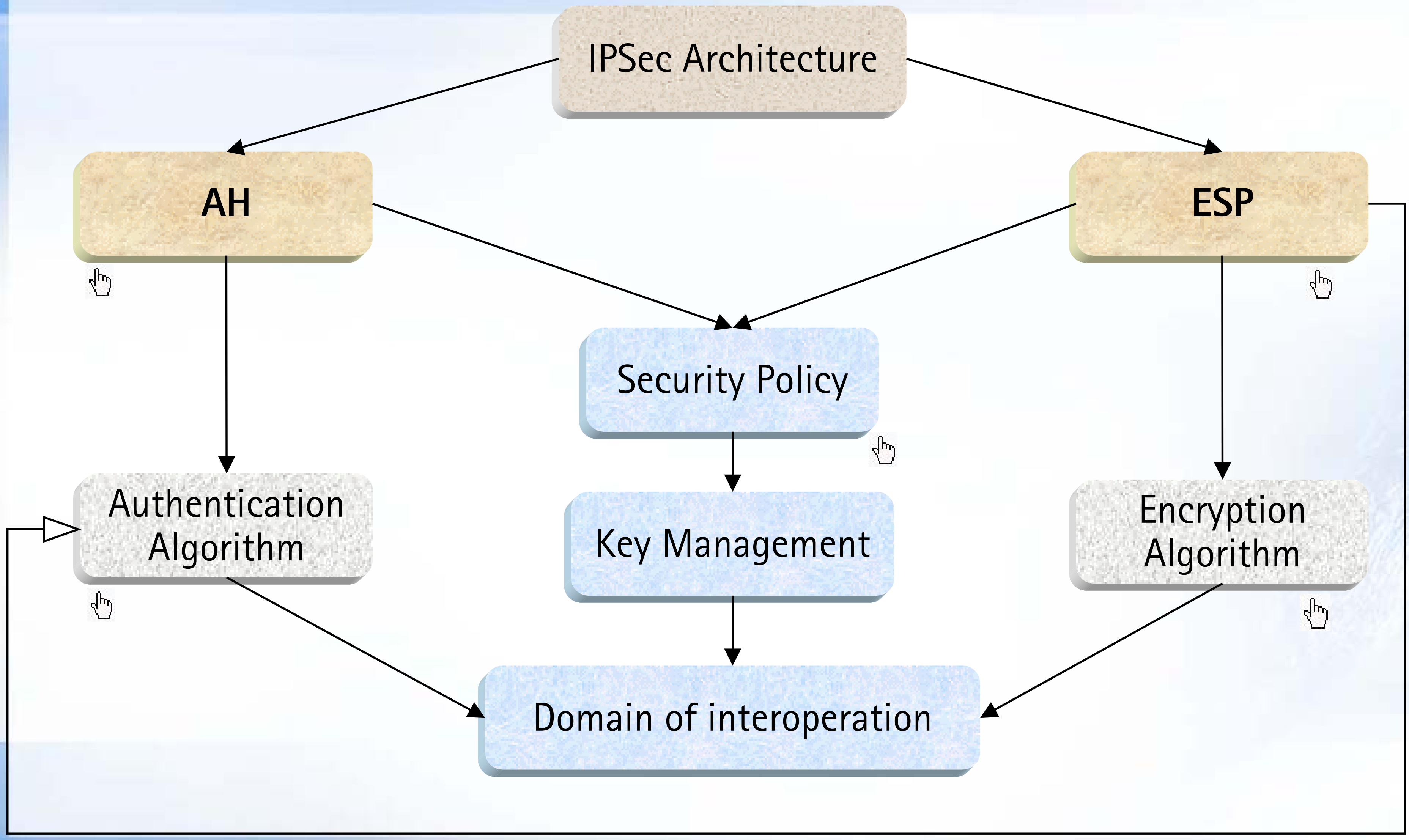
T-79.300 Postgraduate Course in Theoretical Computer Science

Contents

IPSec Overview

- Developed by IETF
- IPSec can be used for
  - Authentication
    - Make sure that the packets are from the indicated user
  - Confidentiality
    - Protect the payload content
  - Data Integrity
    - Protect the IP packet against accidental or deliberate modifications
  - Anti-Reply & Access Control
    - Prevent reply attacks
- Network layer security
  - IP packet level authentication/encryption (IP datagram protection)
  - No need for application layer security solutions (TLS, SSL, HTTPS, ..)
- IPSec is not a protocol, it is a "security solution" $\Rightarrow$ IPSec Suite

IPSec Architecture



IPSec Modes

- IPSec implementation
  - Transport Mode
    - ☞Protects the IP payload
    - ☞For end-to-end security
  - Tunnel Mode
    - ☞Protects the whole IP packet (header + payload)
    - ☞Used when the destination is not the end-point
- Considering the AH and ESP, four different combinations can exist:
  - AH Transport Mode
  - ESP Transport Mode
  - AH Tunnel Mode
  - ESP Tunnel Mode

Thesis Under Discussion

About the Thesis

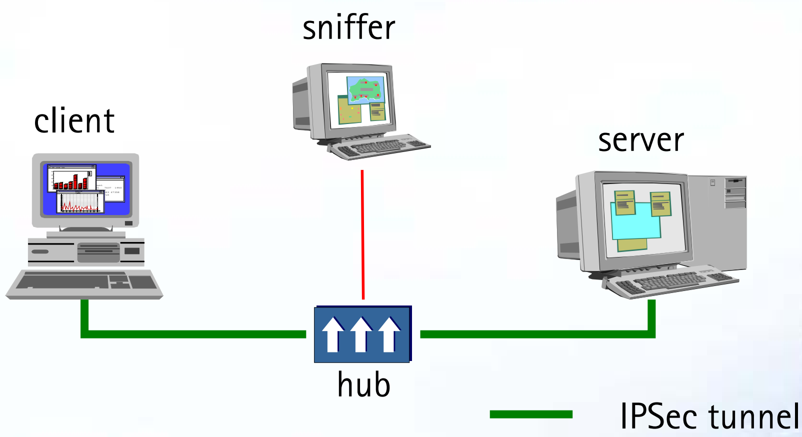
- Title:
  - "IPSec Overhead in Wireline and Wireless Networks for Web and Email Applications"
- By
  - George C. Hadjichristofi
- In
  - Faculty of Virginia Polytechnic Institute
- Date
  - 2001
- URL
  - http://scholar.lib.vt.edu/theses/available/etd-11152001-181815/

Research Goals

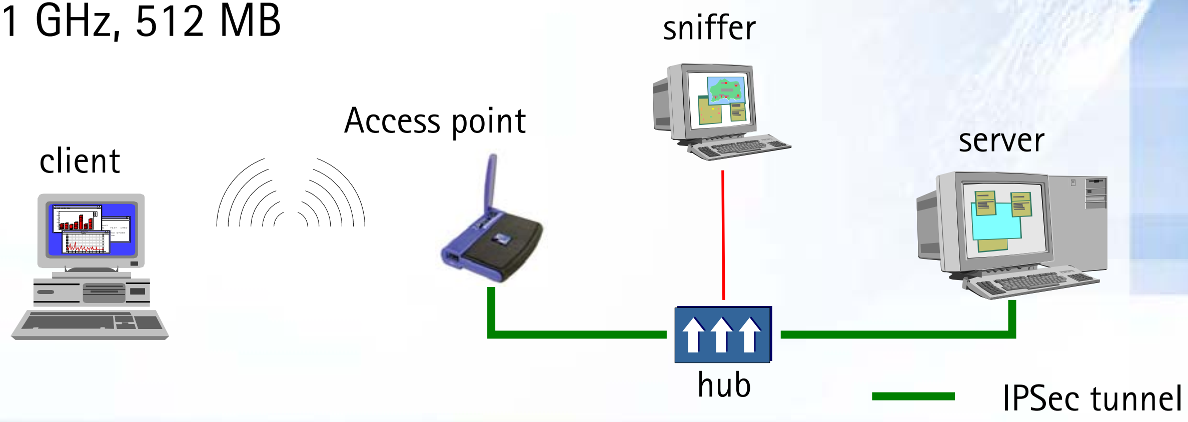
- The research focused on:
  - The IPSec overhead introduced by using different combinations of IPSec policies and configurations
    - ESP encryption and authentication vs. ESP encryption and AH authentication
    - ESP authentication vs. AH authentication
  - The overhead introduced by different authentication algorithms
    - HMAC-MD5 vs. HMAC-SHA1
  - The network load overhead
    - Compressed vs. Uncompressed files
  - HTTP vs. SMTP protocol security overhead
  - The IPSec overhead in Wireline vs. Wireless networks

Research Methodology: Configuration 1

- Wireline testbed configuration
  - 10 Mbps Ethernet LAN
  - Server: Intel Celeron, 566 MHz, 128 MB
  - Client: Pentium Pro, 200 MHz, 32 MB
  - OS: RedHat Linux 7.0, kernel 2.2.16
  - FreeS/WAN IPSec

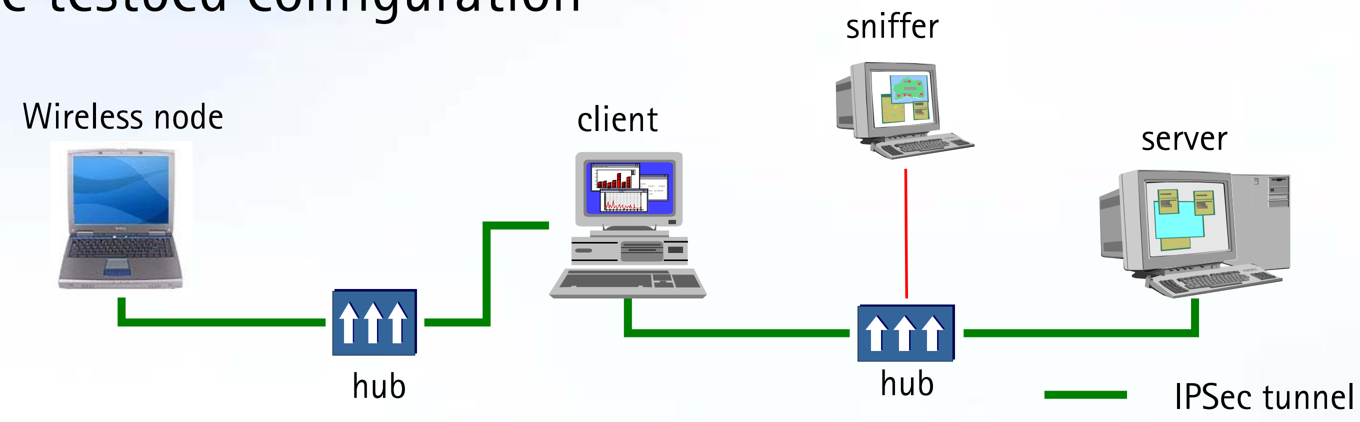


- Wireless testbed configuration
  - 10 Mbps Ethernet LAN
  - 2 Mbps IEEE 802.11 WLAN
  - Wireless node: Pentium III, 1 GHz, 512 MB

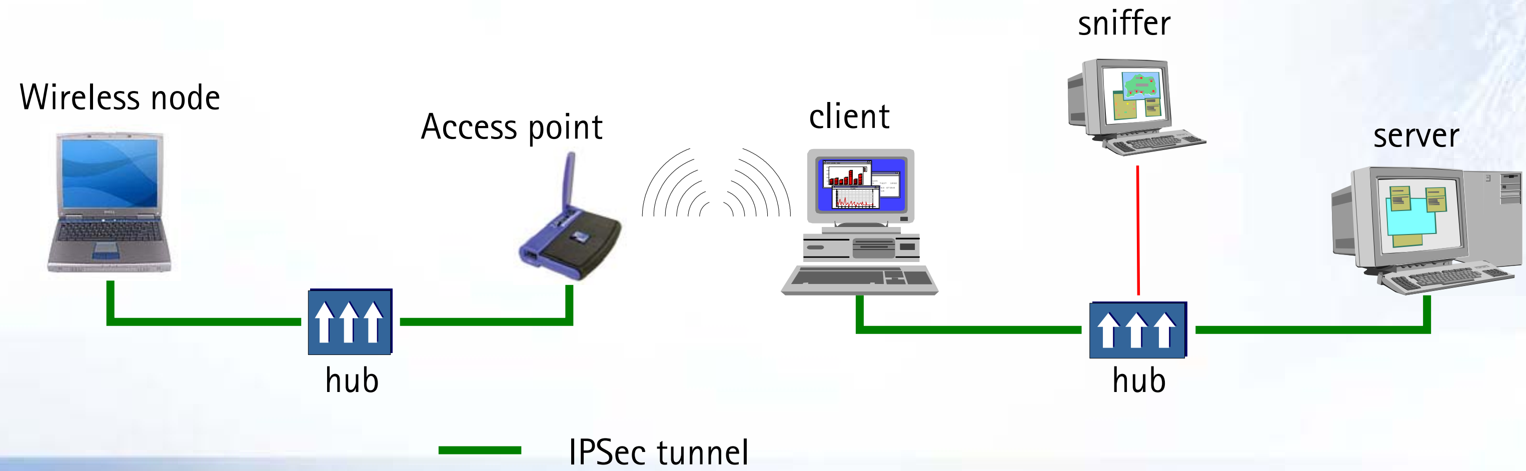


Research Methodology: Configuration 2

- Wireline testbed configuration

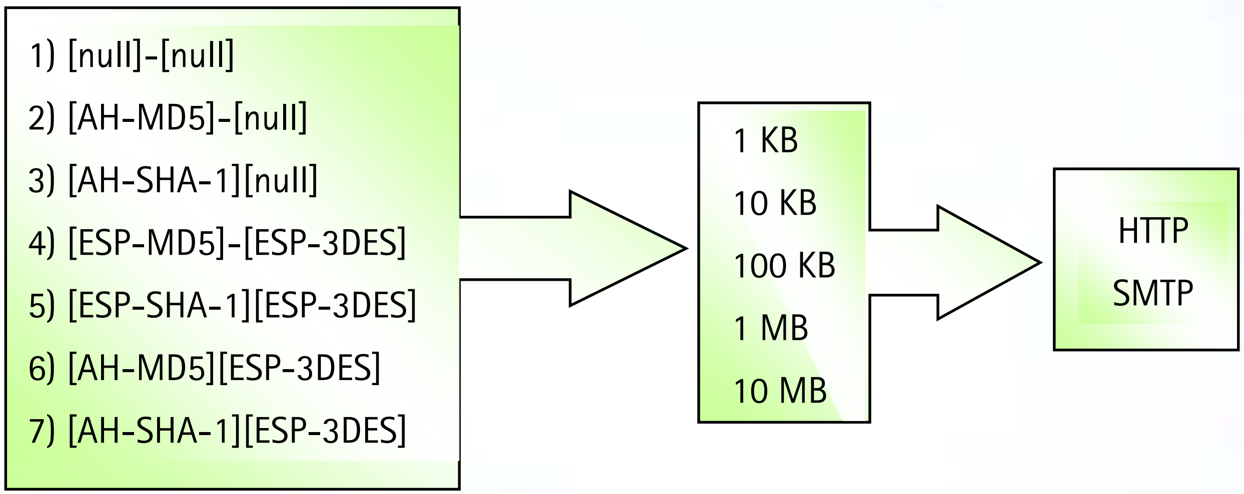


- Wireless testbed configuration



Research Methodology: Test Scenarios

- Seven test scenarios were performed
  - Combination of Authentication & Encryption with different algorithms
- Different file sizes
- Two protocols (HTTP & SMTP)

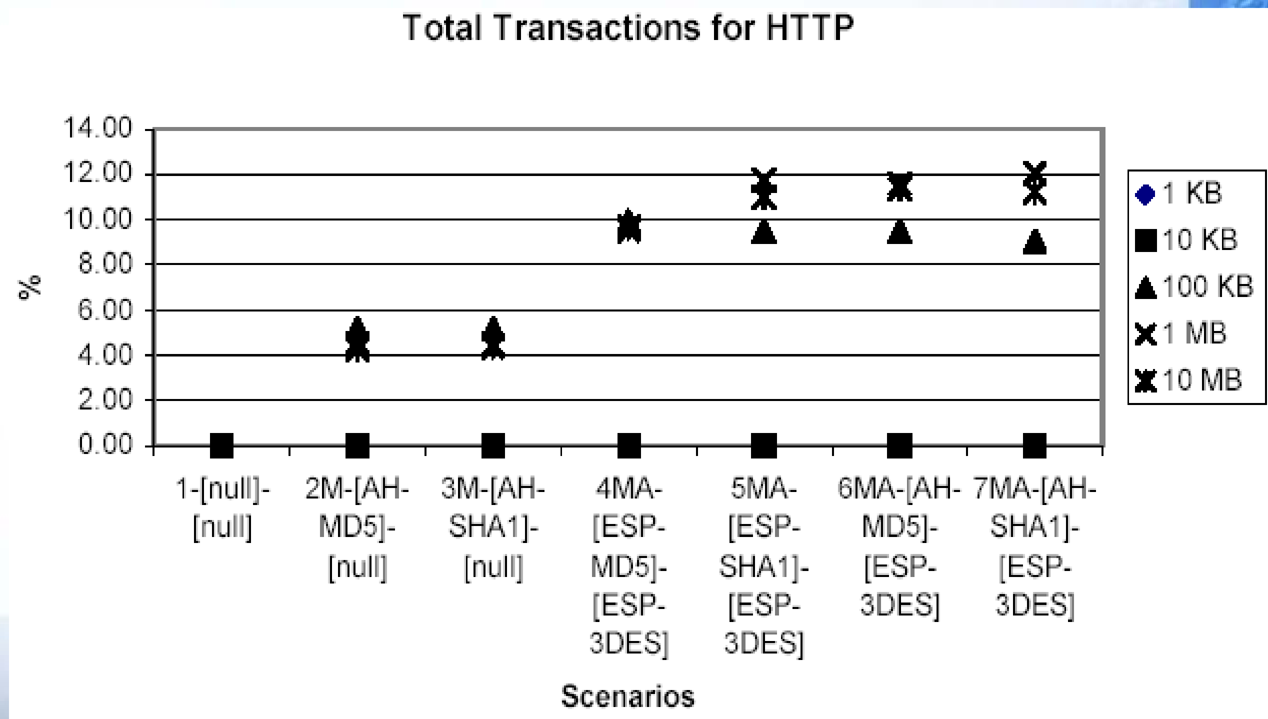


Recorded Metrics

- Number of Transactions
  - From Server to Client (S2C)
  - From Client to Server (C2S)
- Network Load => Throughput (bytes/sec)
  - From Server to Client (S2C)
  - From Client to Server (C2S)
- Transfer Time
  - The time needed to transfer a file from the server to the client

1) Number of Transactions: Results

- Total number of transactions
  - Increases as we go from [null][null] to [AH-SHA-1][ESP-3DES]
    - But not for small files (1 KB & 10 KB)
    - . . . Only for large files (100 KB, 1 MB, 10 MB)
  - Same trend with SMTP

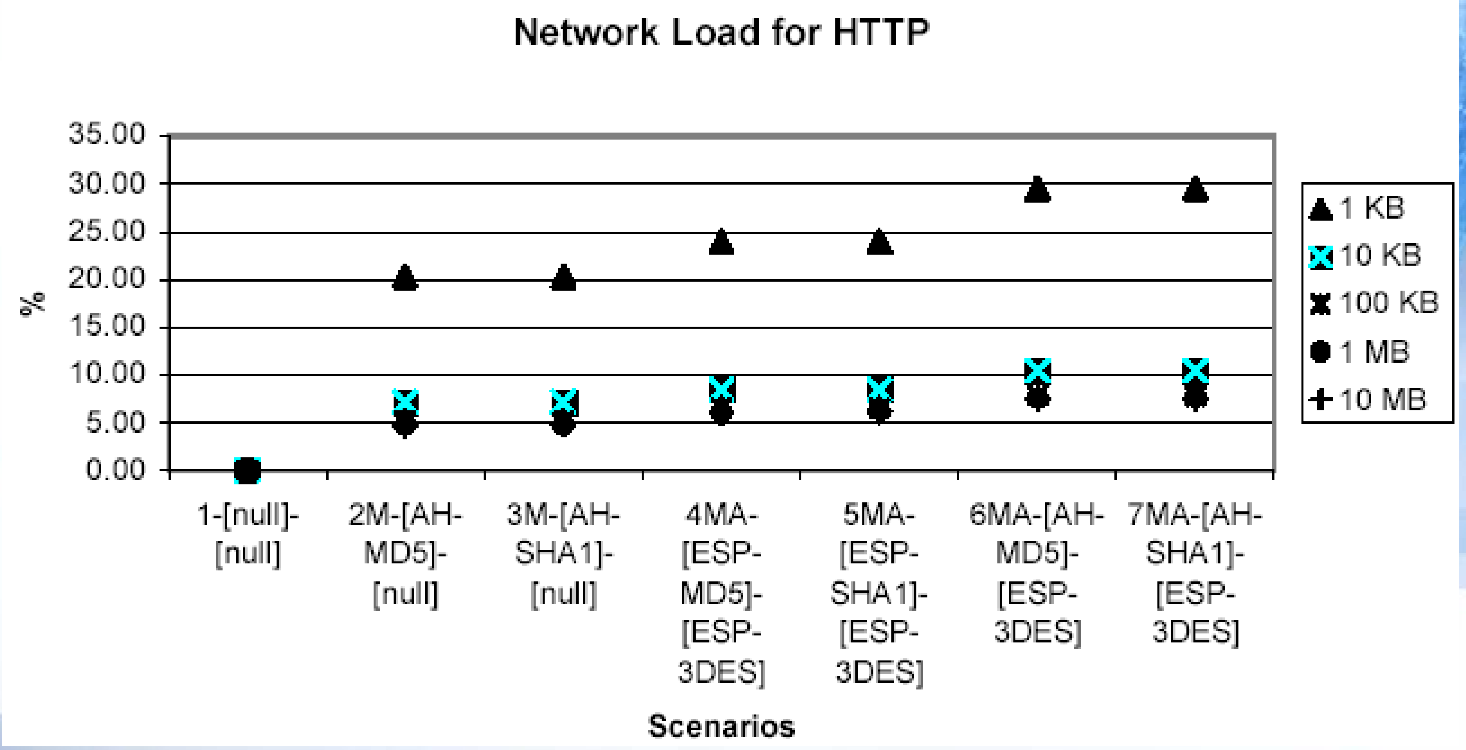


1) Number of Transactions: Analysis

- Facts:
  - The Server is faster than the Client
  - Connection-oriented scenarios
  - Large files saturates the TCP buffers more than the small files
- The client needed to “slow down” the server continuously
  - More control messages have been generated
  - => More transactions when sending large files

2) Network Load : Results

- Network Load
  - Network gets loaded more and more as we go from [null]-[null] to [AH-SHA-1][ESP-3DES]
    - Smaller files loads the network more than the bigger ones . . .
  - Same trend with SMTP

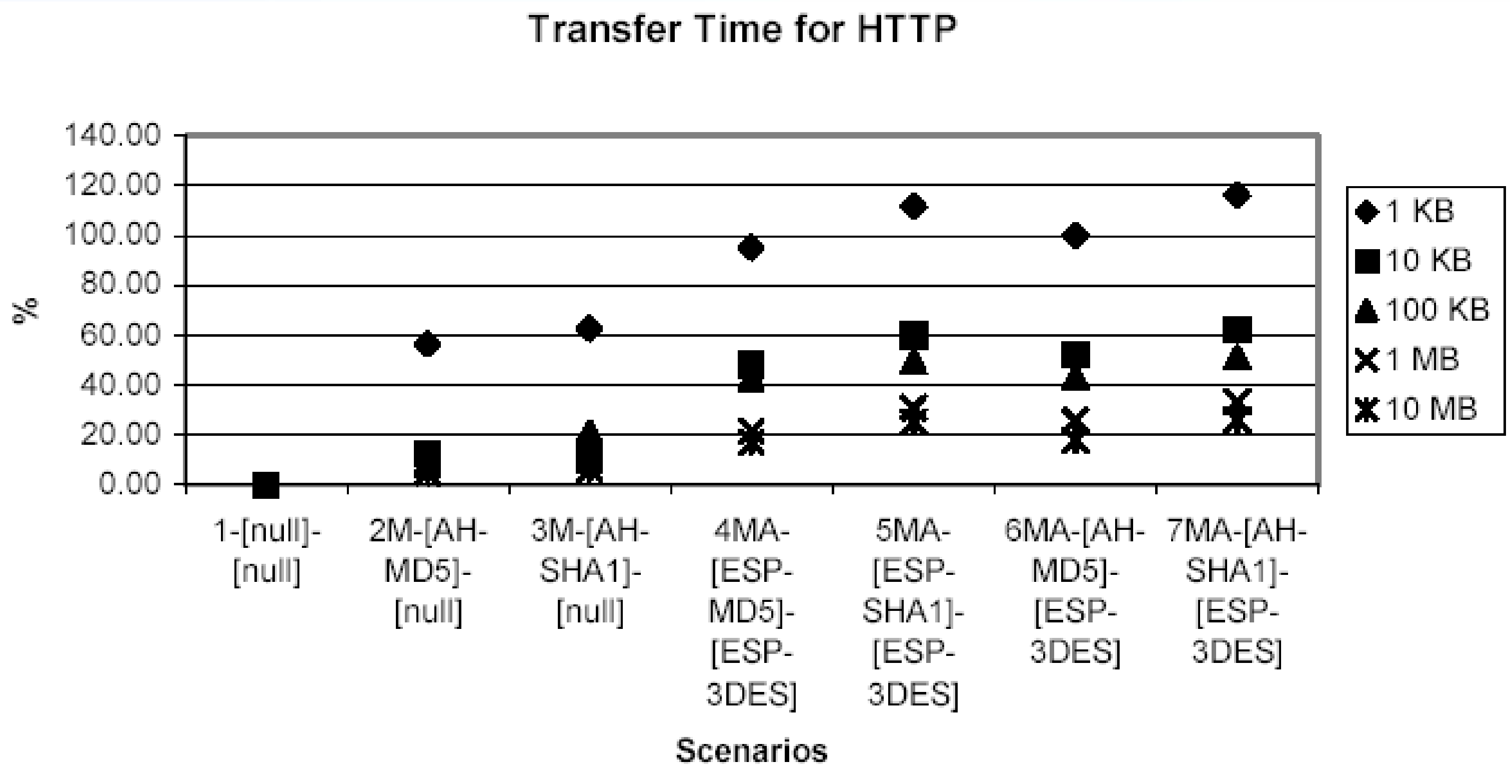


2) Network Load : Analysis

- The Client is slower than the Server
- More control messages are needed
- Network get loaded
- Larger number of ACKs are needed to control the traffic flow for small files

3) Transfer Time: Results

- Transfer Time



3) Transfer Time: Analysis

- Again . . .
  - The Client is slower than the Server
  - and
  - The transfer time increases because of the needed control message to control the data flow . . .

Wireless Case

- Almost same as wireline results

Thesis Conclusions

- [AH authentication & ESP encryption] results in a higher network load compared to [ESP authentication & ESP encryption]
- *Network Load* and *Number of Transactions* were the same for both the wireless and wireline environments
- In the wireless environment, the *Transfer Time* increases due to the additional overhead of the wireless link

Thesis Critic (1/2)

- Quite Vague metrics
  - Number of transactions
    - What does this include ?
    - What is the transaction size ?
  - Network Load
    - C2S vs. S2C
    - "Throughput" should have been used instead of vague "Network load" . . .
  - Transfer Time
- Most of the results can not be generalized
  - Many un-identified HW and SW bottlenecks
- The MTU and its effect in the results was not studied quite well . . .

Thesis Critic (2/2)

- The wireless transmission was not clear . . .
- No IPv6 consideration
- What about UDP . . .

- Personal Opinion: bad written thesis . . .

Other Studies and Results

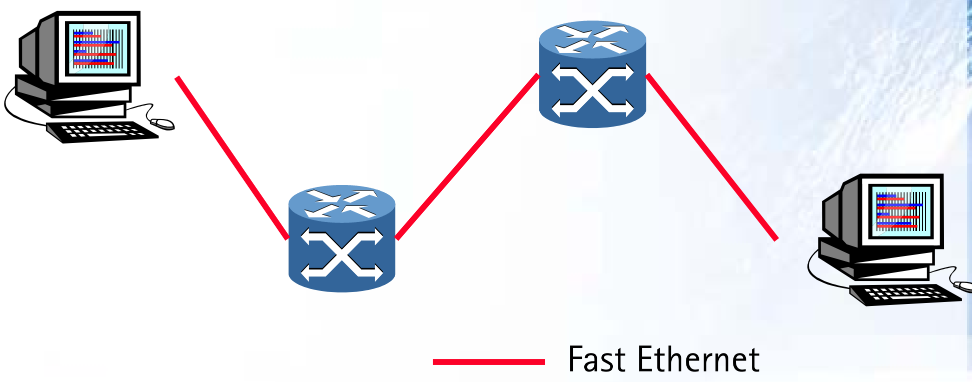
This section is based on some literature research

Performance Evaluation of Data Transmission Using IPSec over IPv6 Networks

Ref[3]

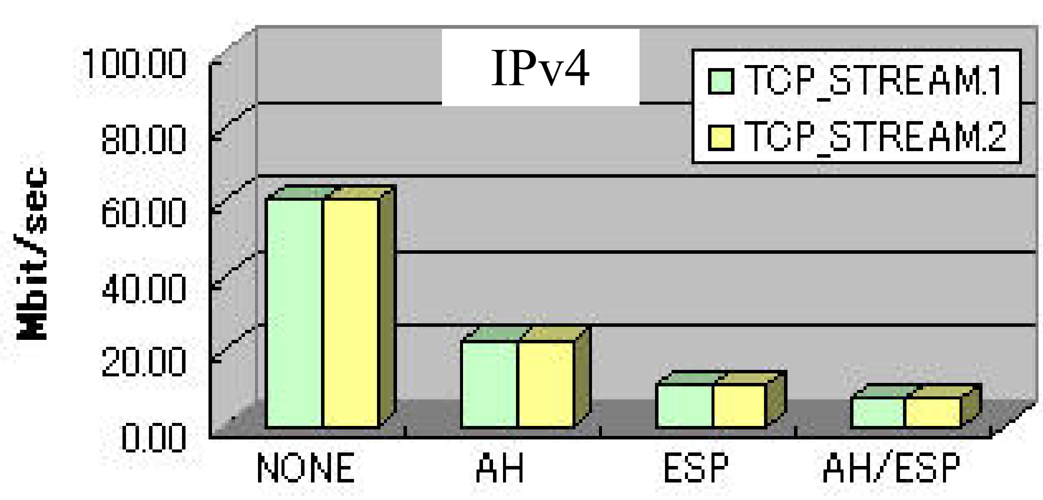
Overview and Test Configuration

- Evaluation system
  - Host:
    - Intel Pentium III 450 MHz, 128 MB
    - OS: FreeBSD 2.2.8
  - Router
    - Inter Pentium III 500 MHz, 256 MB
    - OS: FreeBSD 2.28
- The end-to-end throughput is evaluated in the following cases:
    - Without IPSec
    - Only with AH
    - Only with ESP (3DES)
    - With both AH and ESP
  - TCP & UDP / IPv4 & IPv6
- Test types
  - Stream data
    - For 60 minutes –streaming-
  - Request/Response
    - Ping message

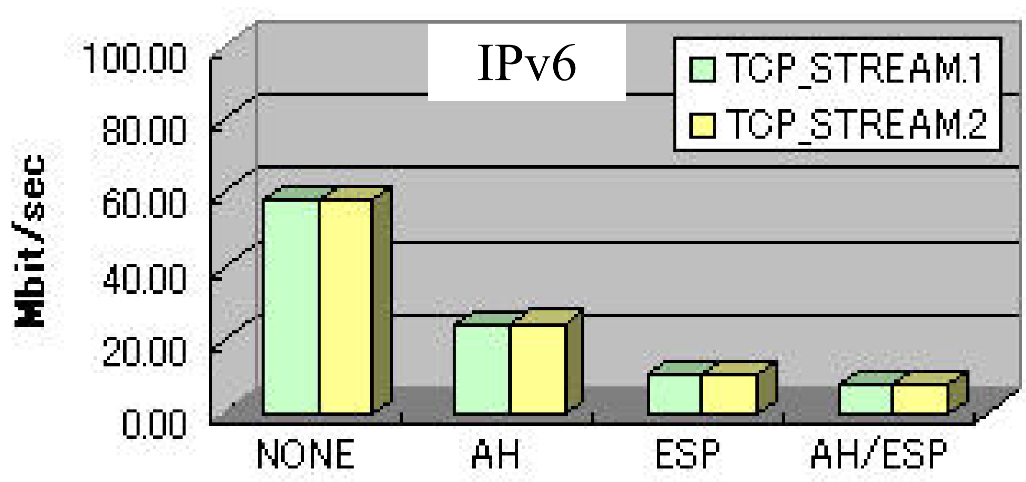


Stream Data: TCP/IPSec Results

- MTU = 4 KB
- TCP socket buffers
  - TCP_STREAM1 = 57,344 Bytes
  - TCP_STREAM2 = 32,769 Bytes

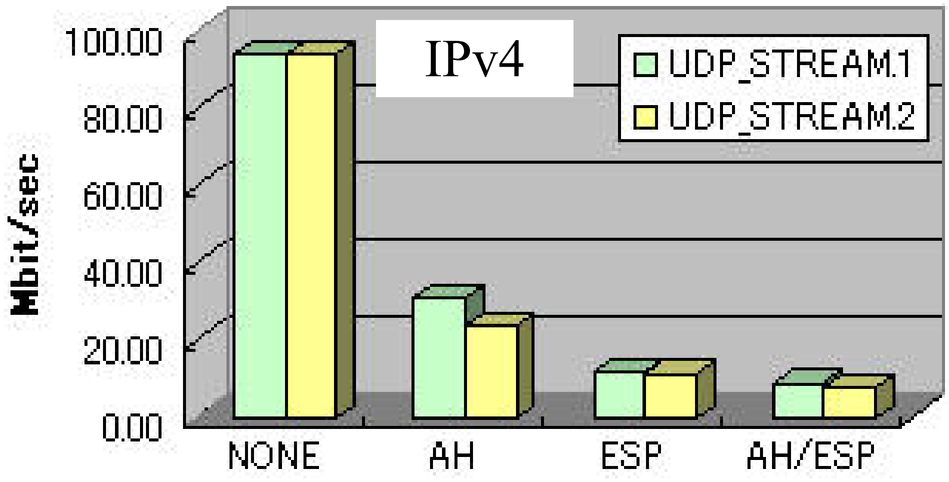


- End-to-End throughput
  - With AH
    - TP degrades ~ ½
  - With ESP
    - TP degrades ~ ¼
  - Both AH & ESP
    - TP degrades ~ < ¼

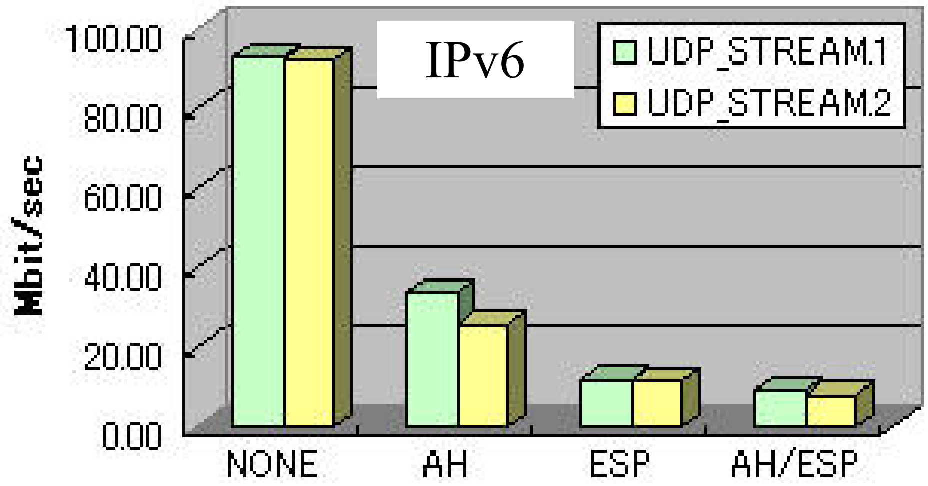


Stream Data: UDP/IPSec Results

- UDP socket buffer size = 32,768 bytes
- MTU
  - UDP_STREAM1 = 4,096 bytes
  - UDP_STREAM2 = 1,024 bytes



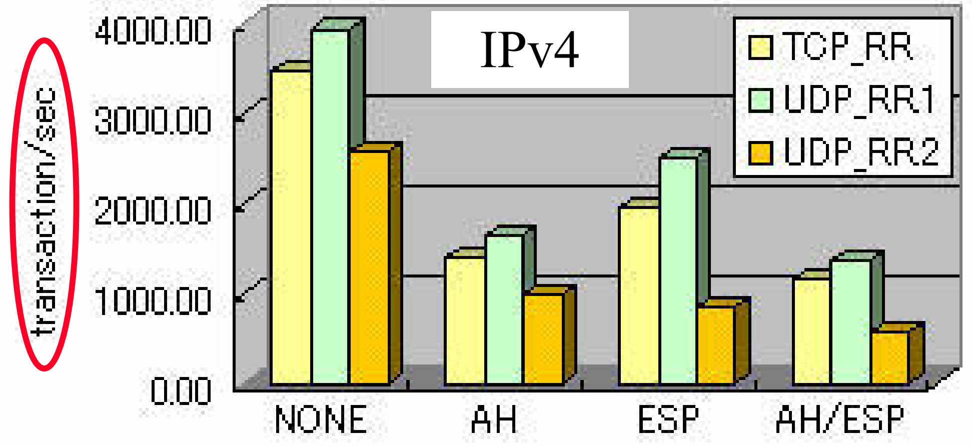
- End-to-End throughput
  - With AH
    - TP degrades ~ 1/3
  - With ESP
    - TP degrades ~ 1/9
  - Both AH & ESP
    - TP degrades ~ < 1/9
- When MTU is larger, the TP is better

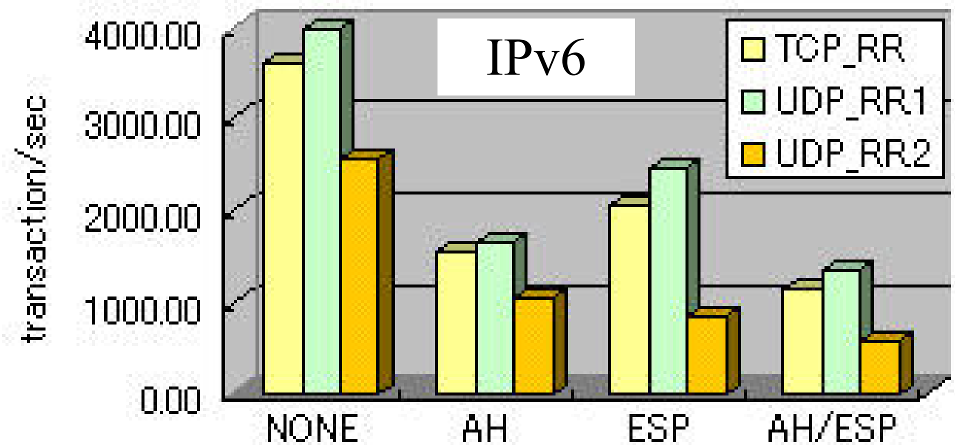


Request/Response : TCP & UDP Results

- Request/Response
  - TCP_RR
    - Request message = 1 byte
    - Response message = 1 byte
  - UDP_RR.1
    - Request message = 1 byte
    - Response message = 1 byte
  - UDP_RR.2
    - Request message = 512 bytes
    - Response message = 4 bytes

- End-to-End throughput
  - ESP TP > AH TP !!
    - @ TCP_RR & UDP_RR.1





Results Analysis: Stream Data

- TCP vs. UDP
    - Without IPSec: TCP throughput is less than UDP throughput
    - With IPSec: TCP throughput ≈ UDP throughput
  - => The IPSec processing is much larger than both TCP and UDP
    - The IPSec overhead dominates the difference between TCP and UDP
- AH vs. ESP
  - AH throughput ≈ twice ESP throughput
    - In stream data, the packet size is larger than the header length (basic IP + AH)
    - The required processing for ESP is far larger than that for AH
- ESP vs. (AH/ESP)
  - The processing of ESP dominates that of AH
- MTU
  - Larger MTU ⇒ less throughput

Result Analysis: Request/Response

- Stream Data vs. Request/Response
  - By applying IPSec;
    - The throughput degradation with Request/Response data is less than Stream data

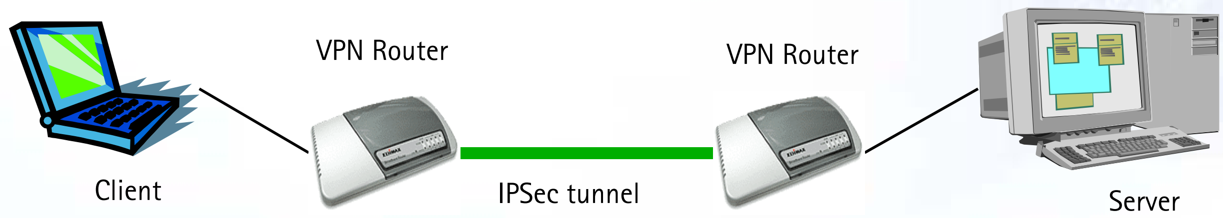
- The processing overhead of Request messages is far less than that of stream data

Evaluation of IPSec-based Wireline VPN

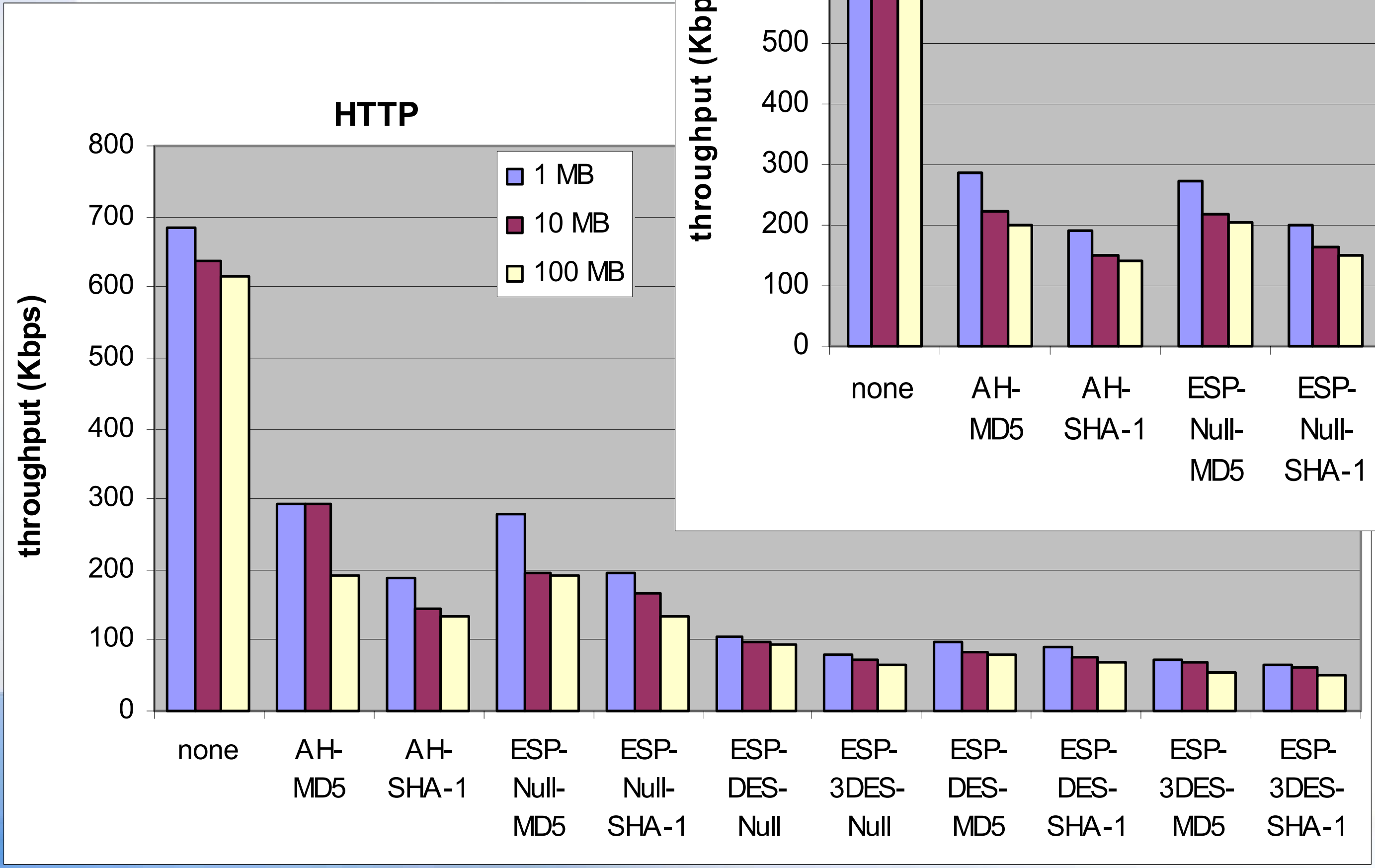
Ref[4]

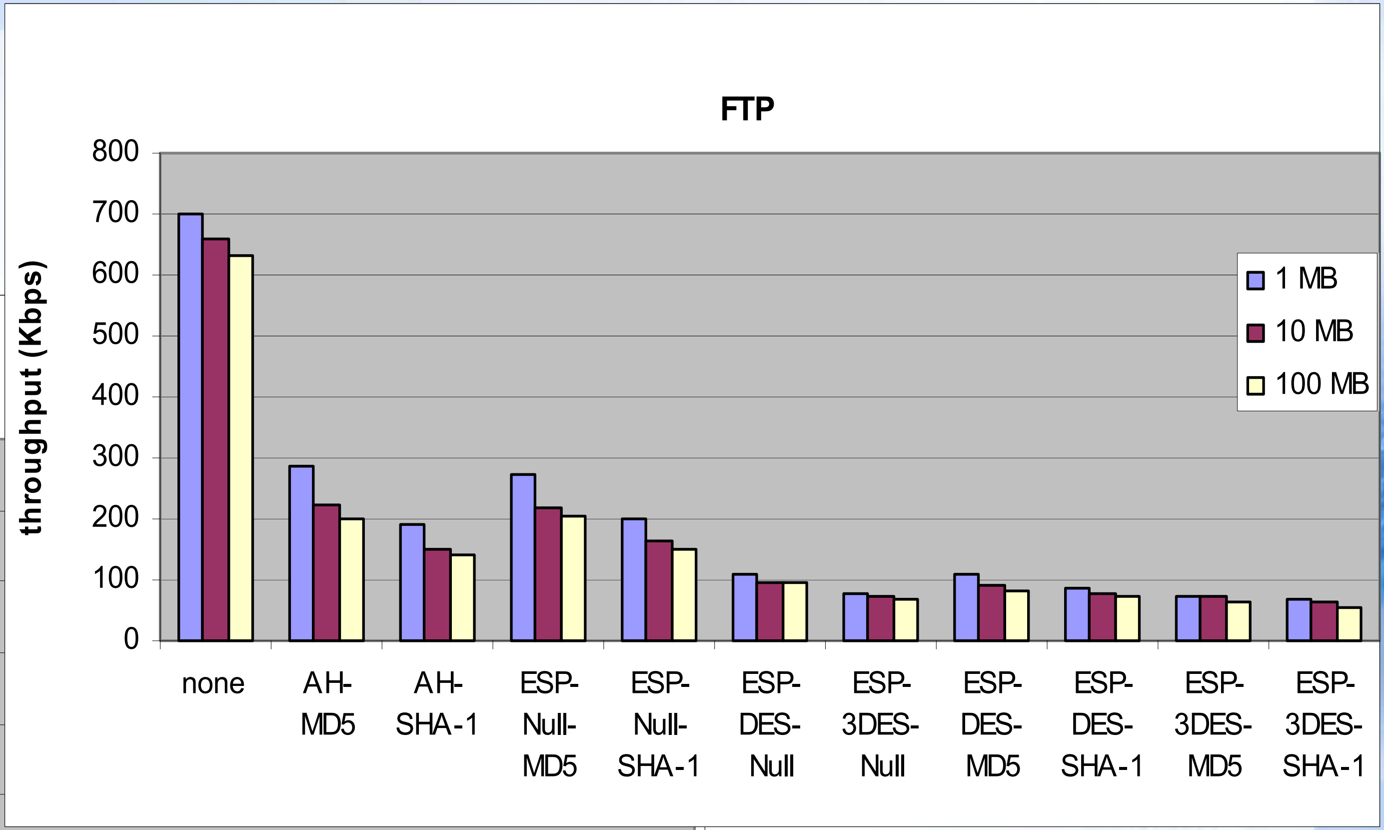
Test Bed Configuration

- Test setup
  - Security tunnel is built using two VPN Routers
    - SAMSUNG S3C4510, vLinux kernel 2.2.14
  - Client
    - Pentium III, 850 MHz, Windows 2000
  - Server
    - Pentium III, 850 MHz, Windows 2000
- Two application protocols where evaluated
  - FTP
  - HTTP



Results


HTTP
throughput (Kbps)
800
700
600
500
400
300
200
100
0
1 MB
10 MB
100 MB
none
AH-MD5
AH-SHA-1
ESP-Null-MD5
ESP-Null-SHA-1
ESP-DES-Null
ESP-3DES-Null
ESP-DES-MD5
ESP-DES-SHA-1
ESP-3DES-MD5
ESP-3DES-SHA-1


FTP
throughput (Kbps)
800
700
600
500
400
300
200
100
0
1 MB
10 MB
100 MB
none
AH-MD5
AH-SHA-1
ESP-Null-MD5
ESP-Null-SHA-1
ESP-DES-Null
ESP-3DES-Null
ESP-DES-MD5
ESP-DES-SHA-1
ESP-3DES-MD5
ESP-3DES-SHA-1

Analysis

- Largest throughput degradation comes with [ESP-3DES-SHA-1]
  - ~ 90 % compared to the [null]

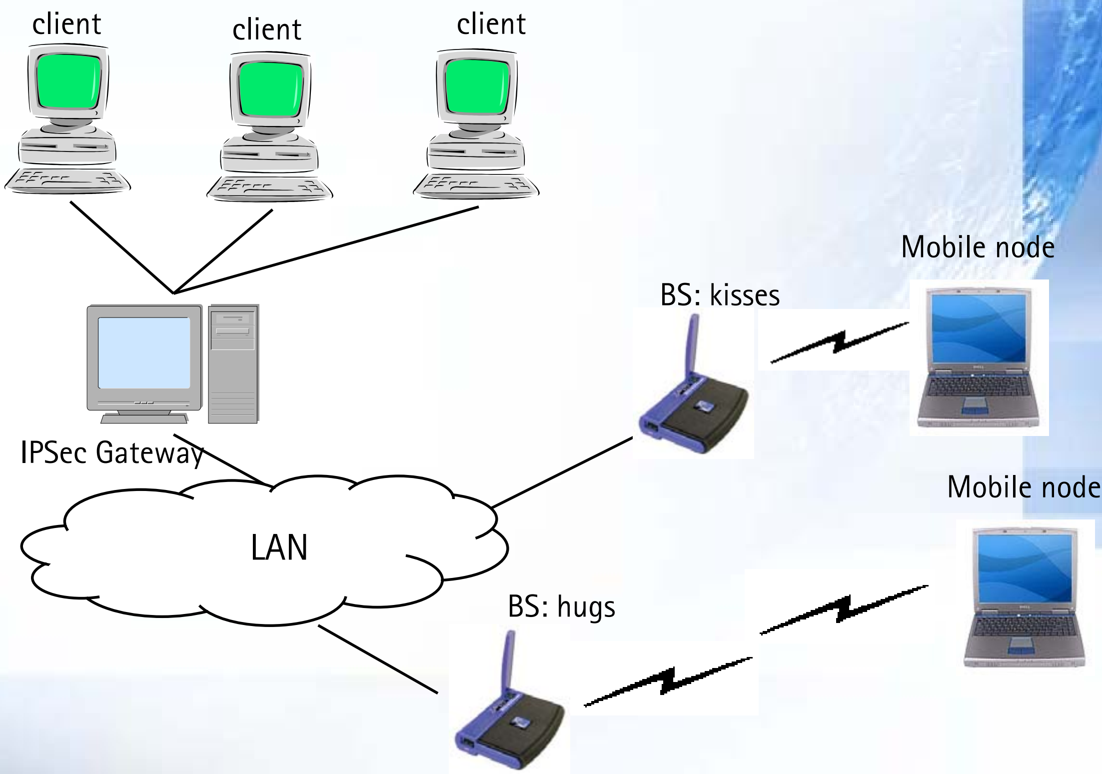
- IPSec throughput is almost the same for both FTP and HTTP
  - The end-to-end throughput will not be affected by the upper layer protocols
  - In other words . . .
  - The data type and payload will not have any influence in IPSec throughput

Evaluation of IPSec-based Wireless VPN

Ref[5]

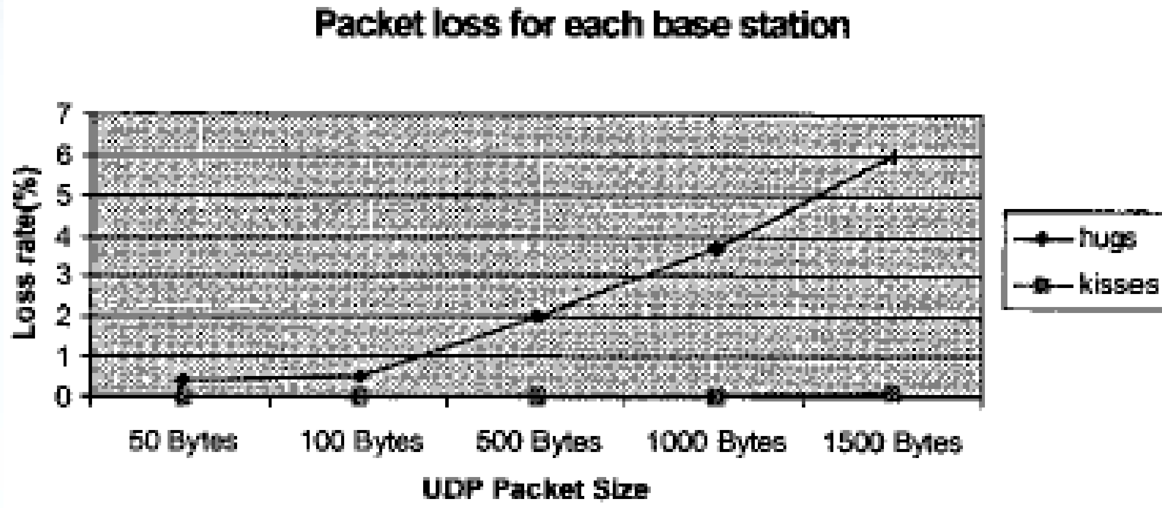
Test Bed Configuration

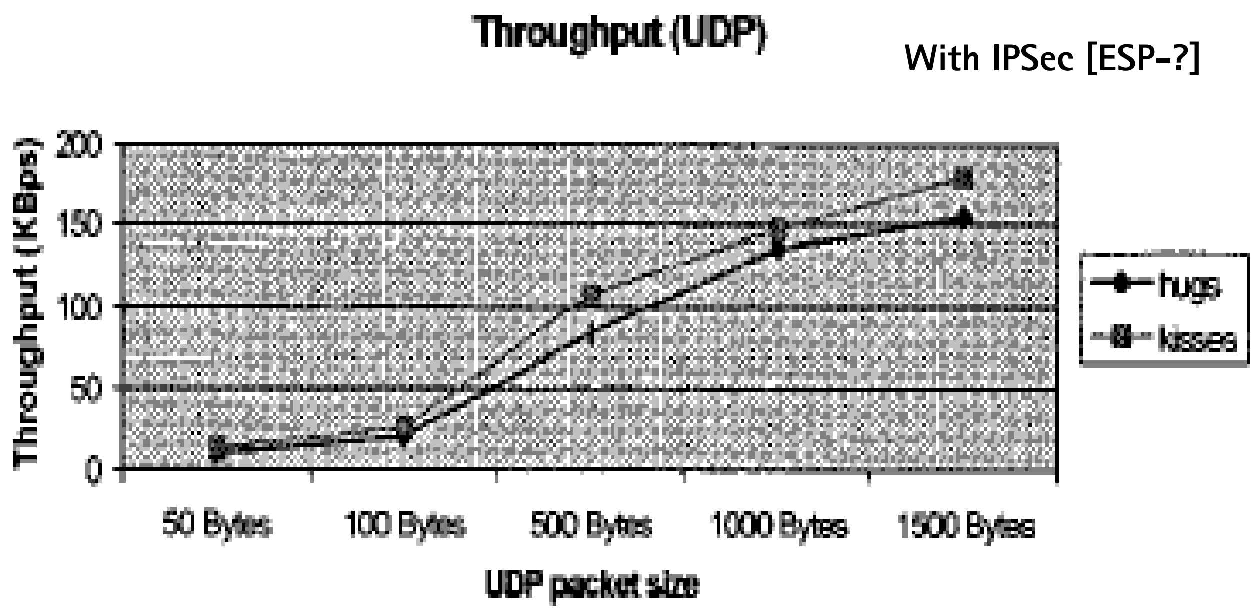
- Test setup
  - Clients
    - Intel PC, RedHat Linux 6.2, kernel 2.2.16
  - Mobile Nodes
    - Apple 4G computers, OS9
  - IPSec Gateway
    - Pentium 1, 32 MB, FreeS/WAN 1.5
  - Base Stations : IEEE 802.11 Wireless LAN
    - Base station (hugs) : ~5 meters away from the mobile nodes
    - Base station (kisses): ~25 meters away from the mobile nodes
  - => The signal strength of kisses is 150% higher than that of hugs



Results (1/2)

- Throughput
  - The UDP throughput of kisses and hugs is about the same
    - The gap increases for larger packet sizes
- Packet Loss
  - The signal from hugs is weaker than kisses
  - When the packet size increase, the loss rate increases as well

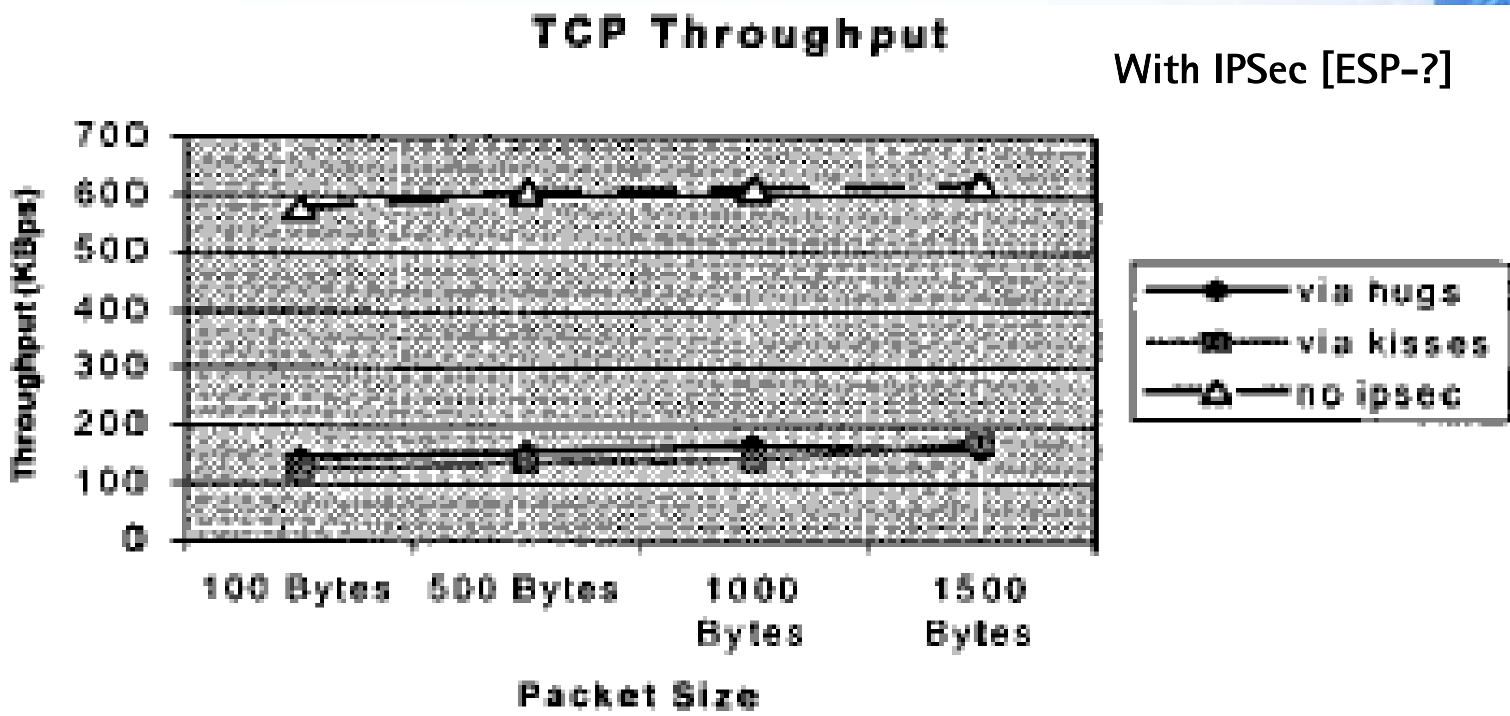




With IPSec [ESP-?]

Results (2/2)

- TCP throughput without IPSec is about 3 times that with IPSec
- When packet size increased from 1 KB to 1.5 KB
  - TCP TP increased for the connection via kisses
  - TCP TP decreased for the connection via hugs
    - => For different signal strength, TCP has different optimal
    - => strong signal strength supports larger optimal packet size



Analysis

- UDP
  - UDP Packet loss favors small packet sizes
  - UDP throughout favors strong signals and large packet size

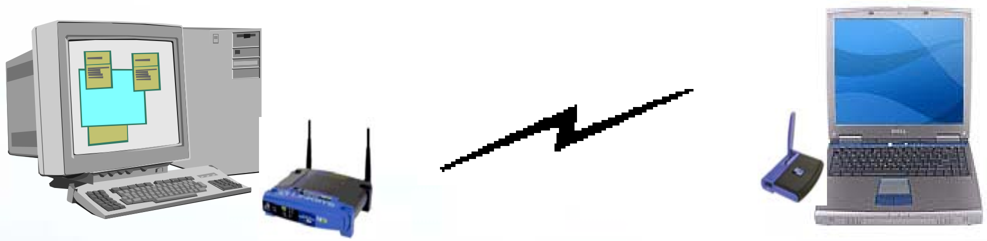
- TCP
  - There exist an optimal packet size for highest throughput

Wireline vs. Wireless VPN IPSec performance

Ref[6]

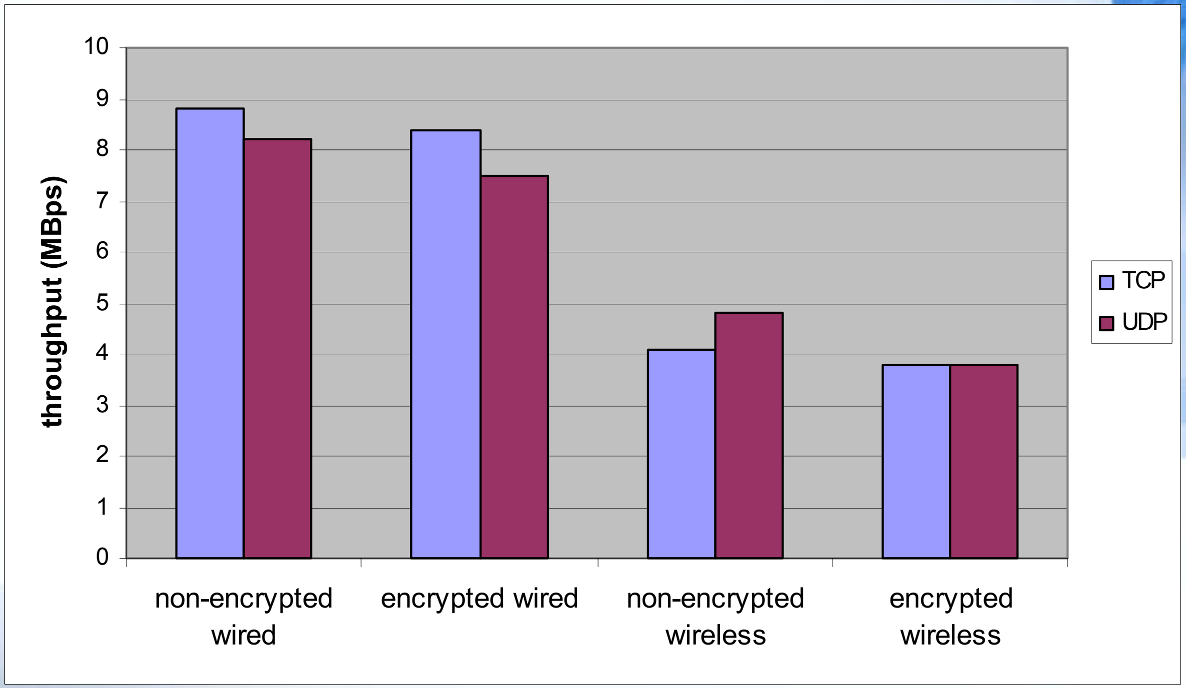
Test Bed Configuration

- Client
  - Windows 2000, 11 Mbps wireless connection
- Server
  - Windows 2000, 100 Mbps wireless connection



Test Results

- For wired connection
  - ~ 4.5% throughput degradation for TCP
  - ~ 8.5% throughput degradation for UDP
- For wireless connection
  - ~ 7% throughput degradation for TCP
  - ~ 20% throughput degradation for UDP



Overall Conclusions

Conclusions (1/2)

- Without IPSec, TCP throughput is less than that of UDP (with no IPSec)
- With IPSec, the gap between TCP and UDP throughputs vanishes
- IPSec processing overhead is dominant to TCP and UDP overheads
- IPSec/IPv4 exhibits the same throughput as IPSec/IPv6
- For streaming data, the AH throughput is almost twice that of ESP
- The overhead of ESP is larger than that of AH
- For the streaming data; larger MTU results in less throughput
- By applying IPSec, the throughput degradation with Request/Response data is less than streaming data
- The processing overhead of the Request/Response messages is far less than that of streaming data

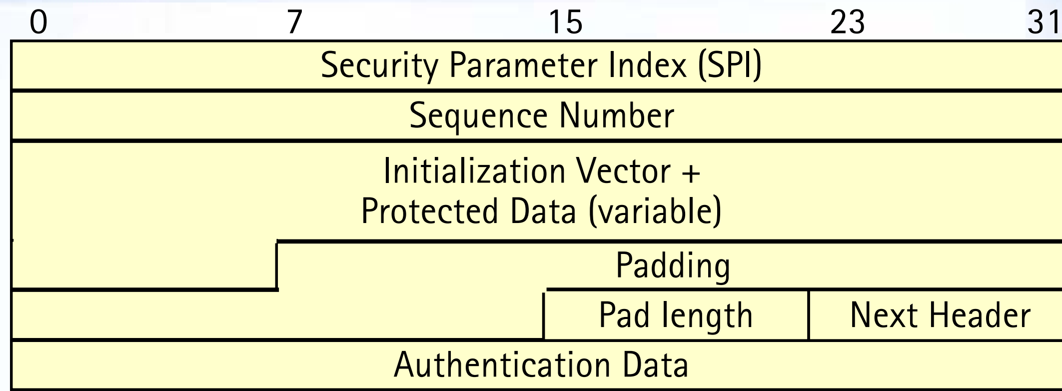
Conclusions (2/2)

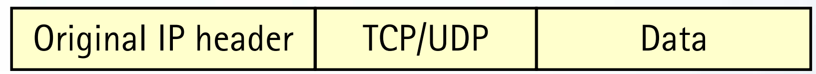
- The [ESP-3DES-SHA-] causes the largest throughput degradation amongst other IPSec scenarios
- The IPSec throughout is independent than upper layer protocols
- In wireless environment, there exist an optimal TCP packet size for the highest throughput as a function of wireless signal strength
- In wireless environment, the UDP throughput degradation is higher than that of TCP

Appendices

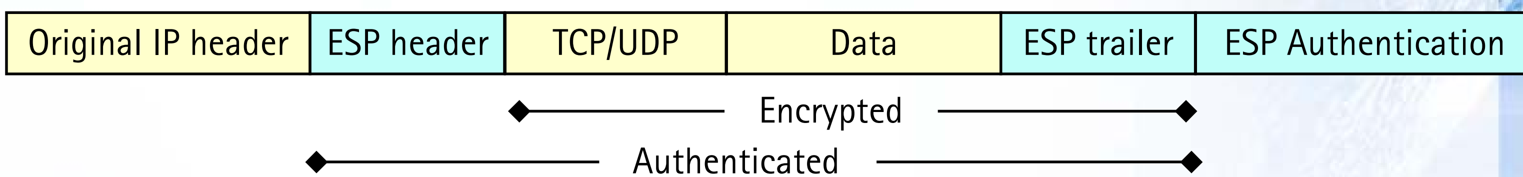
IPSec Extension Headers: ESP

- Encapsulation Security Protocol
  - Provides confidentiality

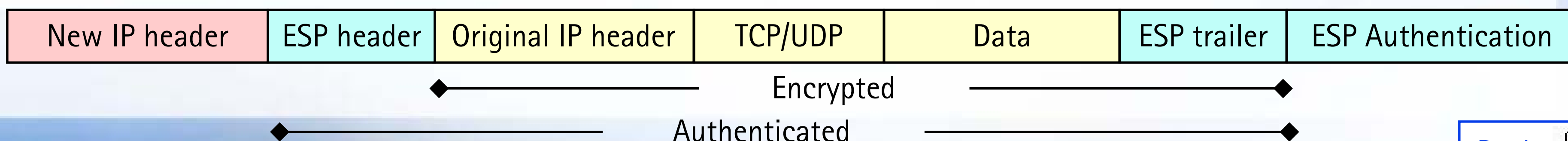




- Transport Mode ESP



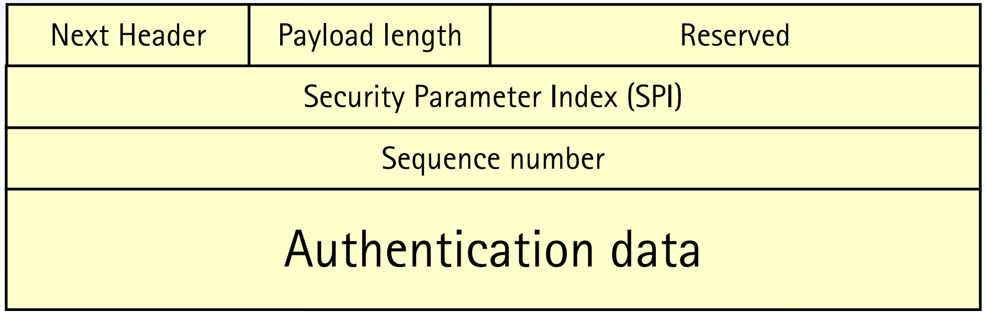
- Tunnel Mode ESP



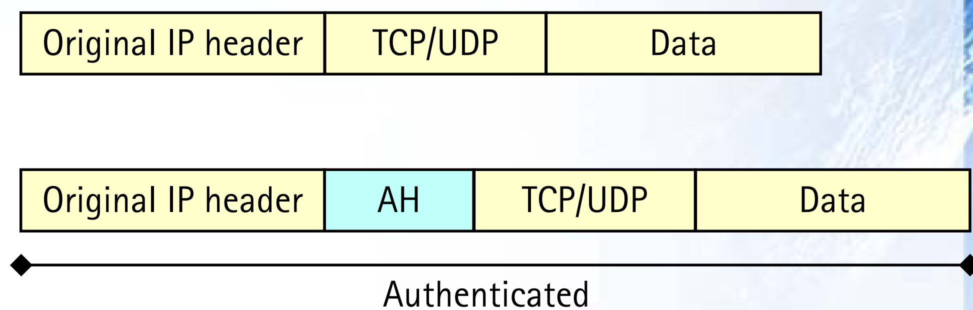
Back

IPSec Extension Headers: AH

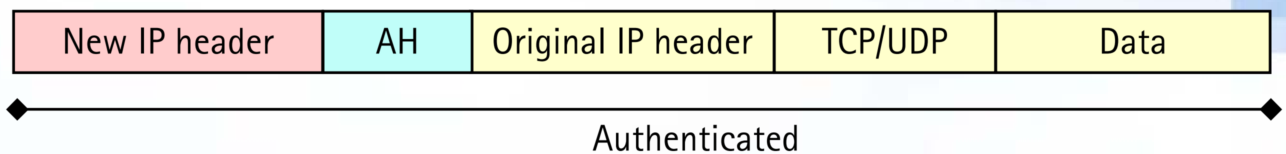
- Authentication Header
  - Provides the services of authentication and data integrity



- Transport Mode AH



- Tunnel Mode AH



Back

Authentication & Encryption Algorithms

- Authentication Algorithms
  - MAC
    - Used by the two parties to validate the exchanged messages
  - MD5
    - An algorithm that takes as input a message of any length, and produces as an output a 128-bits "message digest" of the input
  - SHA-1
    - An algorithm that takes as input a message of any length, and produces as an output a 160-bits authenticator value
- Encryption Algorithms
  - DES
    - A symmetric key encryption method. It uses 56-bit key and them breaks the message into 64-bits blocks and encrypts them
  - 3DES
    - Triple-DES. Three keys are involved in the operation

Back

IPSec Management

- Key Management
  - IKE (Internet Key Exchange)
    - Protocol used to negotiate the cryptographic keys used by IPSec
- Security Policy
  - AS (Security Association)
    - A contract between two IPSec peers. It defines the security services that will be used
  - SAD (Security Association Database)
    - A database containing all SA's that are currently established
  - SPD (Security Policy Database)
    - Contains the defined security policies to be enforced for any SA
- Domain of Interoperations
    - Parameters used and negotiated by different peers

Back

Abbreviations

3DES	Triple-DES
AH	Authenticated Header
DES	Data Encryption Standard
ESP	Encapsulated Security Protocol
HMAC	keyed-Hashing Message Authentication Code
HTTP	HyperText Transfer Protocol
ICV	Integrity Check Value
IKE	Internet Key Exchange
ISAKMP	Internet Security Association and Key Management Protocol
MAC	Message Authentication Code
MD5	Message Digest 5
MTU	Maximum Transfer Unit
SHA	Secured Hash Algorithm
SMTP	Simpel Message Transfer Protocol
VPN	Virtual Private Network

References

[1] George C. Hadjichristofi, MSc. thesis, "*IPSec Overhead in Wireline and Wireless Networks for Web and Email Applications*"

[2] Hadjichristofi, G.C.; Davis, N.J., IV; Midkiff, S.F.;"*IPSec Overhead in Wireline and Wireless Networks for Web and Email Applications*", Performance, Computing, and Communications Conference, 2003. Conference Proceedings of the 2003 IEEE International , 9-11 April 2003
Pages:543 - 547

[3] S. Ariga, et. al, "*Performance Evaluation of Data Transmission Using IPSec over IPv6 Networks*", URL: http://www.isoc.org/inet2000/cdproceedings/1i/1i_1.htm, 10.2.2004

[4] Jin-Cherng Lin; Ching-Tien Chang; Wei-Tao Chung; *"Design, implementation and performance evaluation of IP-VPN*" , Advanced Information Networking and Applications, 2003. AINA 2003. 17th International Conference on , 27-29 March 2003
Pages:206 - 209

[5] Wei Qu; Srinivas, S.; "*IPSec-based secure wireless virtual private network*"; MILCOM 2002. Proceedings , Volume: 2 , 7-10 Oct. 2002 ,Pages:1107 - 1112 vol.2

[6] Jeff Custard; "*Wireless VPN Performance Tests*"; URL: http://www.scd.ucar.edu/nets/projects/wireless/performance.tests.vpn.html, 12.2.2004

[7] Alain Jourez; "*An IPsec Primer*", URL: http://www.iihe.ac.be/scimitar/J1000/ipsec/, 12.2.2004

Thank You !