IPSec/VPN Security Policy: Correctness, Conflict Detection and Resolution

Zhi Fu Motorola

S. Felix Wu UC Davis

He Huang *James Loh* NortelNetworks

Fengmin Gong IntruVert

Ilia Baldine MCNC

Chong Xu Cosine

http://www.cs.ucdavis.edu/~wu

wu@cs.ucdavis.edu

* Many of us were with NCSU while the work was done.

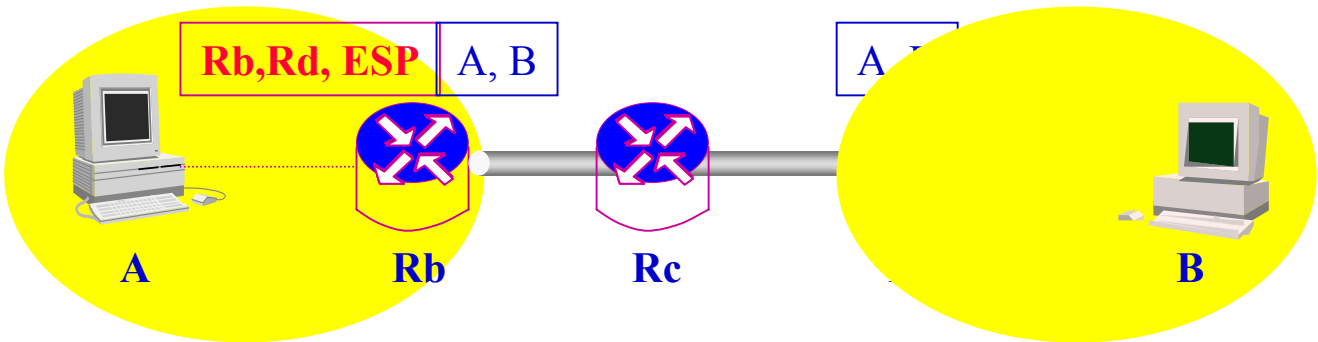
Content

IPSec: Policy and Requirement

- Motivation/Conflict Examples
- Security Requirements
- Conflict Detection and Analysis
- Example
- Remarks

IPSec Policy:
Implementation Policy

- Policy:
  if <condition> then <action>
- IPSec policy:
  Condition: src,dst,src-port,dst-port, protocol, …
  Action: Deny | Allow | ipsec (entry, exit, mode, sec-prot, alg)
- Example:
  Condition: src=A, dst=B, port=*, prot=TCP
  Action: ipsec (Rb, Rd, tun, ESP, 3DES)



Conflict #1:
Privacy and Content Examination


```
(1.[srcIP=A dstIP=B prot=TCP srcPort=ANY dstPort=ANY] →
   IPSec Prot=ESP Mode=Transport
         Algorithm=3DES
         from=A to=B)
(2.[srcIP=* dstIP=* prot=ESP srcPort=ANY dstPort=ANY] → deny )
```

Firewall-- drop any incoming IPSec/ESP encrypted packet

if <src=*, dst=*, prot = esp>

then

<drop>

IPSec Gateway--all packets from A to B should be encrypted with 3DES

if <src = 169.237.6.236, dst = 152.1.75.170, prot=*>

then

<ipsec(169.237.6.236, 152.1.75.170, transport, esp, 3des) >

All Packets Dropped !!

Conflict #2:
Selector Confusion


```
(1.[srcIP=A dstIP=B prot=ANY srcPort=ANY dstPort=ANY] →
   IPSec Prot=AH Mode=Tunnel
         Algorithm=HMAC-SHA
         from=A to=SG-2)
(2.[srcIP=A dstIP=B prot=ANY srcPort=ANY dstPort=ANY] → allow)
(3.[srcIP=* dstIP=* prot=ANY srcPort=ANY dstPort=ANY] → deny )
```

Security Policies:

A: All packets from A to B must authenticate to SG-2

SG-1: All packets from A to B will be allowed, but otherwise denied.

The src and dst changed to be A and SG-2 thus the policy at SG-1 will drop the traffic from A to B.

One way or the other, we loss…..

A

SG-1

```
(1.[srcIP=A dstIP=B prot=ANY srcPort=ANY dstPort=ANY] →
   IPSec Prot=AH  Mode=Tunnel
         Algorithm=HMAC-SHA
         from=A    to=SG-2)
(2.[srcIP=A dstIP=B prot=ANY srcPort=ANY dstPort=ANY] →
   IPSec Prot=ESP Mode=Tunnel
         Algorithm=3DES
         from=SG-1 to=SG-2)
```

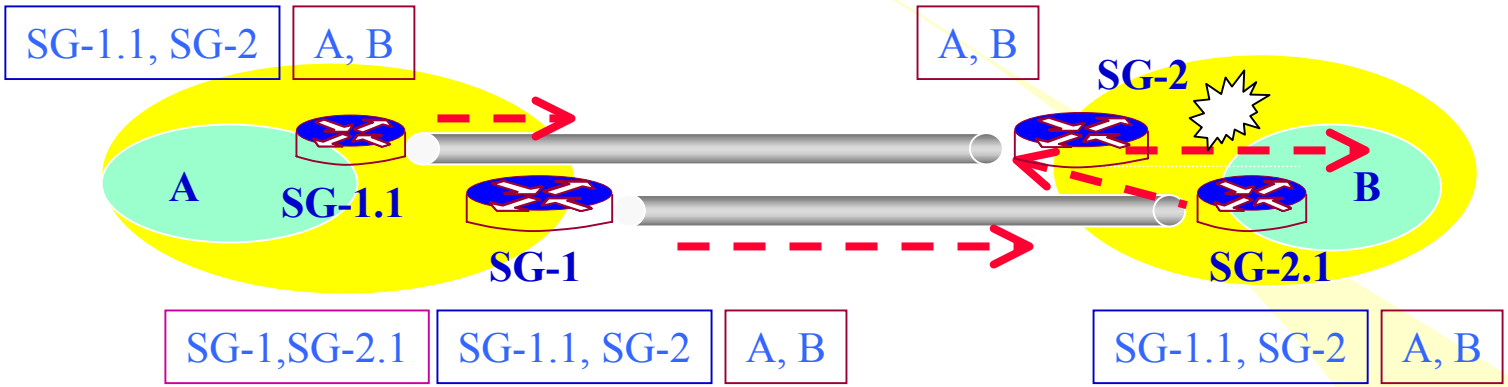
A

A

SG-1

```
(1.[srcIP=A dstIP=B prot=ANY srcPort=ANY dstPort=ANY] →
   IPSec Prot=ESP Mode=Tunnel
         Algorithm=3DES
         from=A    to=B)
(2.[srcIP=A dstIP=B prot=ANY srcPort=ANY dstPort=ANY] →
   IPSec Prot=AH  Mode=Tunnel
         Algorithm=HMAC-SHA
         from=A    to=B)
(3.[srcIP=* dstIP=* prot=ESP srcPort=ANY dstPort=ANY] → deny )
```

Conflict #3:
Tunnel Overlapping




```
(1.[srcIP=A dstIP=B prot=ANY srcPort=ANY dstPort=ANY] →
   IPSec Prot=ESP Mode=Tunnel
         Algorithm=3DES
         from=SG-1.1 to=SG-2  )
(2.[srcIP=* dstIP=* prot=ANY srcPort=ANY dstPort=ANY] →
   IPSec Prot=ESP Mode=Tunnel
         Algorithm=Blowfish
         from=SG-1   to=SG-2.1)
```

Policy Conflict
IPSec/VPN Policy

- A set of (implementation) policies does not quite work well together such that the packets (information bits) are either dropped or revealed/sent unsafely.
- **Requirement(s):** Intention(s) behind the implementation-level policies:
  - e.g., I want to maintain the privacy of certain flows: IPSec ESP Tunnels.
- **Conflicts:**
  - ***a set of policies together does not support the requirements***
  - ***requirements conflict among themselves.***

Policy versus Requirement

- Policy: (implementation, low-level)
  - How should a network entity or a policy domain handle a particular flow of packets?
  - Currently, the processing is based on the selector (i.e., the packet header information).
- Requirement: (intention, high-level)
  - End-to-end, **flow-based** policy + access control + content examination + pairing.
  - If I want to send a sequence of information bits from A to B, how should they be handled, regardless of any packet header transformation (e.g., tunnel/NAT)

IPSec Security Requirements (1)

- **Access Control Requirement (ACR)**

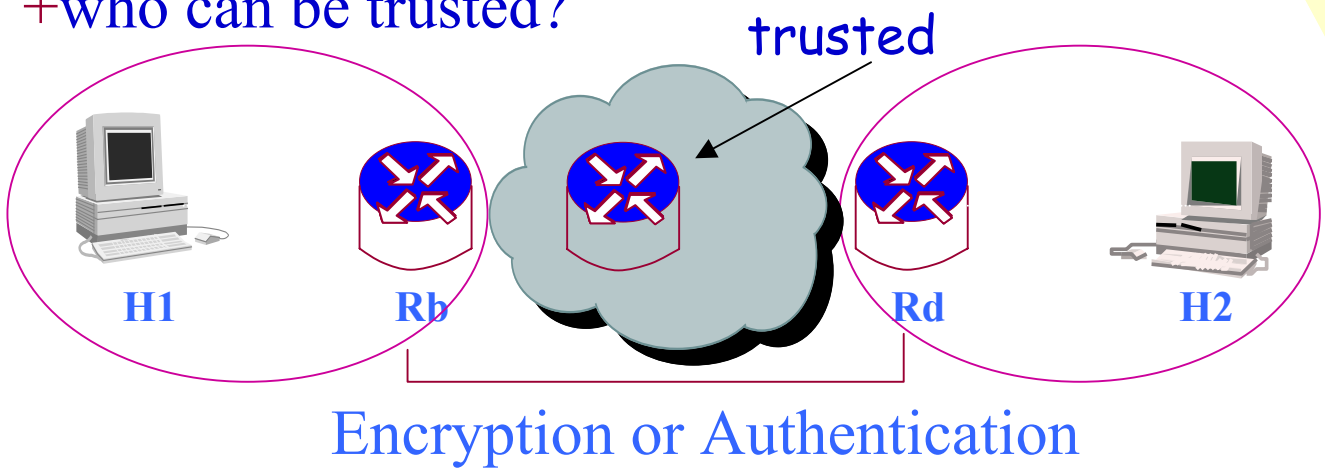
  Restrict access only to trusted traffic

  - E.g. Deny all telnet traffic

- **Security Coverage Requirement (SCR)**

  Apply security functions to prevent traffic from being compromised during transmission across certain area.

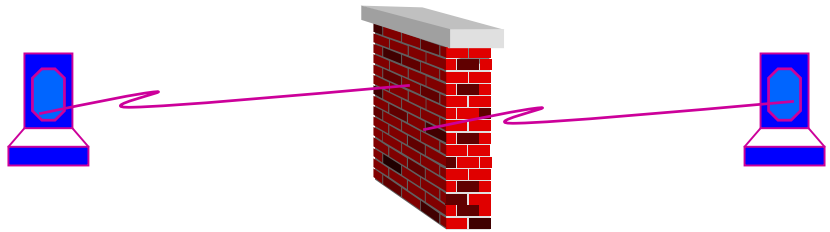
  +who can be trusted?



IPSec Security Requirement (2)

- **Content Access Requirement (CAR)**

  Specify the needs to access content of certain traffic

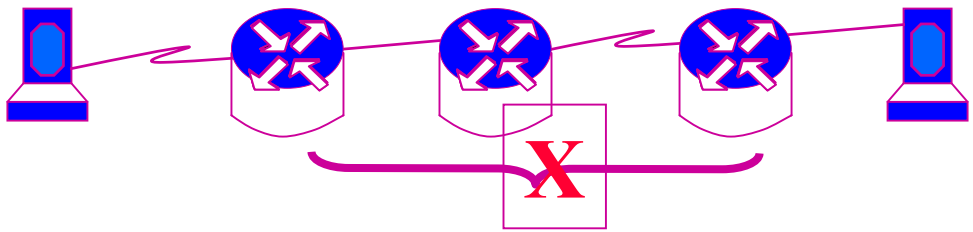


CMR: modify
CER : examine

I will examine the content for intrusion detection

- **Security Association Requirement (SAR)**

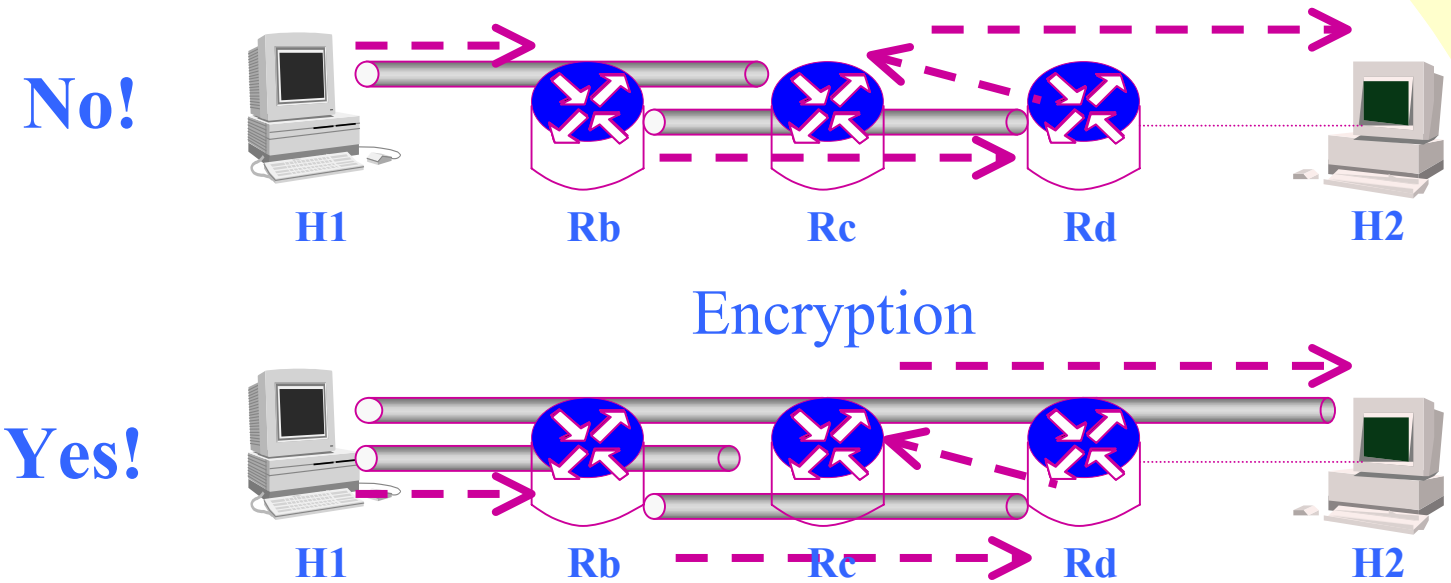
  Specify trust/distrust relationship in SA setup



Security Requirement Satisfaction (1)

- **Access Control Requirement - deny or allow**
- **Security Coverage Requirement**

  All the links and nodes in the area will need to be covered by specified security

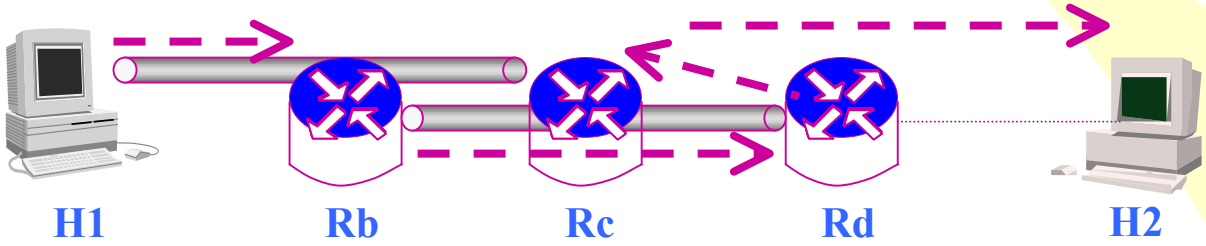


Security Requirement Satisfaction (2)

- **Content Access Requirement**

  Certain node needs to access the content, Rb? Rc?

Rb: No!
Rc: Yes!



- **Security Association Requirement**

  Some nodes are not allowed to set up SA

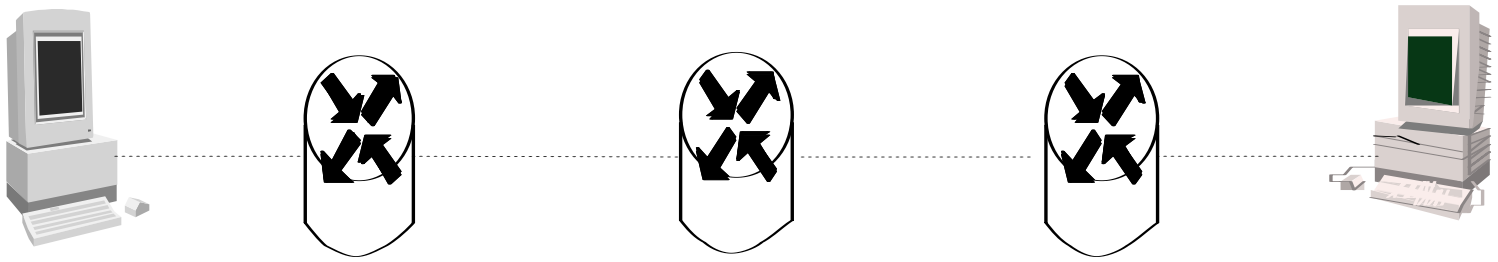
IPSec Requirement Spec.

- Formal specification:
  - ACR-SCR-CAR-SAR
- Conflict Detection in Requirements:
  - Requirement Satisfiability Problem (RSP): given a set of requirements, an algorithm to check whether at all possible to find a set of policies to satisfy all the requirements.
  - Completeness Proof
- Policy Determination:
  - Transformation: if possible, an algorithm to find the "optimal" set of policies.
  - Correctness and Efficiency

IPSec Policy Analysis

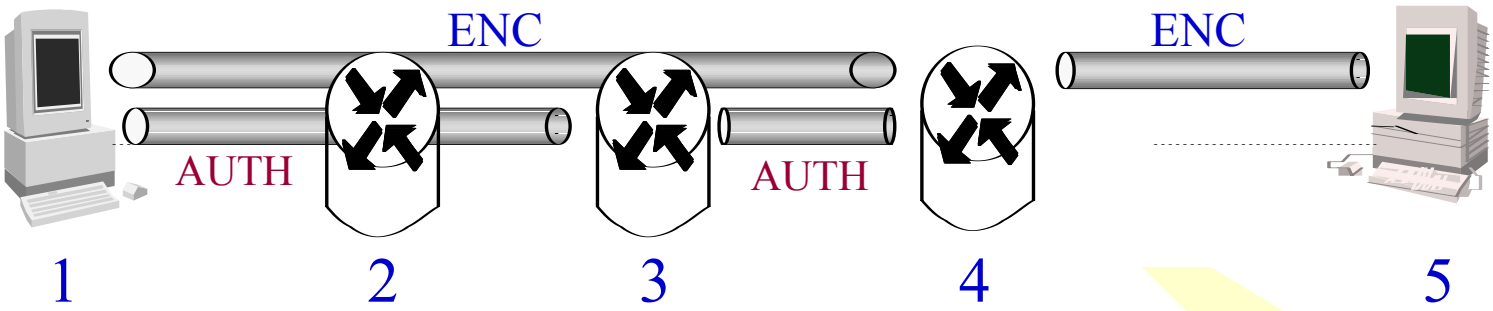
- Policy Conflict Analysis:
  - whether a set of policies satisfy all the specified requirements.
- Policy Conflict Resolution:
  - if conflicts detected, how can we resolve them, if possible?
  - E.g., Tunnel breaking on trusted nodes.

Example (per flow):



	SCR#1:	ENC 2-4 trusted 3
Coverage:	SCR#2:	AUTH 1-4 trusted 3
	SCR#3:	ENC 3-5 trusted 4
Content:	CAR#1:	(ENC, AUTH) by 4
	SAR#1:	not-ENC 2-5
SA relation:	SAR#2:	not-ENC 1-5
	SAR#3:	not-AUTH 1-4

Solution:



	SCR#1: ENC 2-4 trusted 3
Coverage:	**SCR#2: AUTH 1-4 trusted 3**
	SCR#3: ENC 3-5 trusted 4
Content:	**CAR#1: (ENC, AUTH) by 4**
	SAR#1: not-ENC 2-5
SA relation:	**SAR#2: not-ENC 1-5**
	SAR#3: not-AUTH 1-4

Remarks

- IPSec Security Requirement Specification.
- Developed algorithms to detect and analyze conflicts among policies and requirements in polynomial time
  - some of the results are in our new paper.
  - wu@cs.ucdavis.edu
- Future Works:
  - Implementation and Empirical Evaluation
  - Transformation from Policies to Requirements
  - General Topology (integrated with Routing)
  - Inter-domain & Distributed Policy Analysis
  - Other Policies (QoS and Routing)