UNIVERSITY OF AMSTERDAM

Jaap van Ginkel

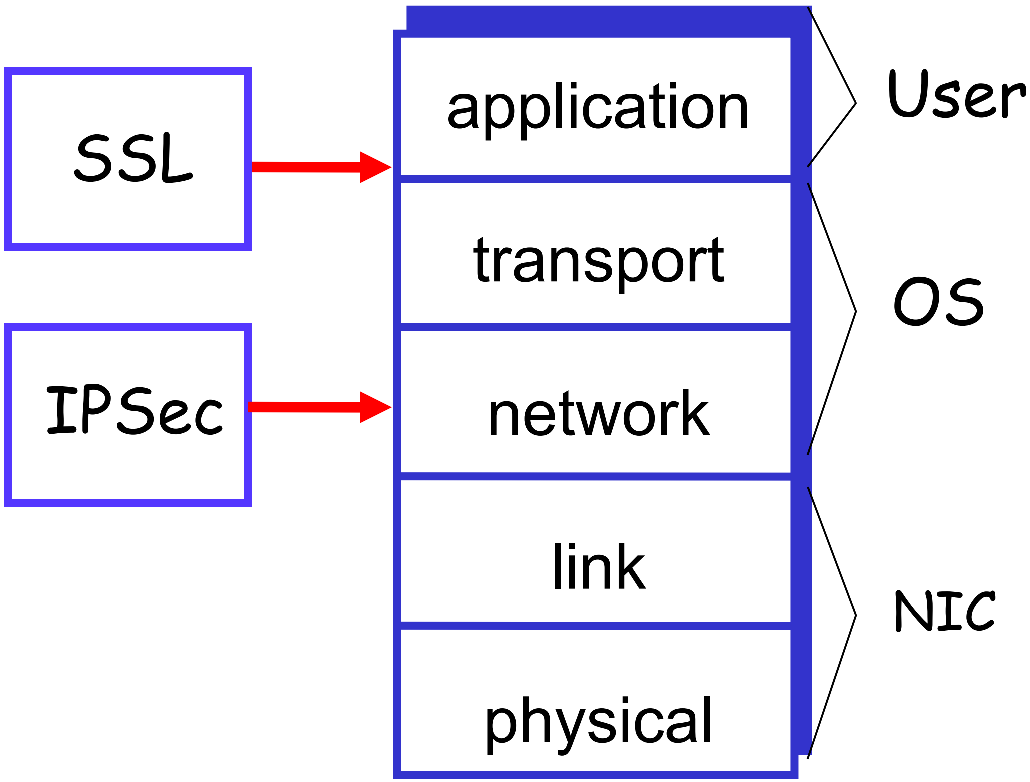
Security of Systems and Networks

October 12, 2019 IPsec

IPSec

IPSec and SSL

- IPSec lives at the network layer
- IPSec is transparent to applications



IPSec and Complexity

- IPSec is a complex protocol
- Over-engineered
  - Lots of generally useless extra features
- Flawed
  - Some serious security flaws
- Interoperability is serious challenge
  - Defeats the purpose of having a standard!
- Complex
- Did I mention, it's complex?

IKE and ESP/AH

- Two parts to IPSec
- **IKE:** Internet Key Exchange
  - Mutual authentication
  - Establish shared symmetric key
  - Two "phases" — like SSL session/connection
- **ESP/AH**
  - ESP: Encapsulating Security Payload — for encryption and/or integrity of IP packets
  - AH: Authentication Header — integrity only

IKE

IKE

- IKE has 2 phases
  - Phase 1 – IKE security association (SA)
  - Phase 2 – AH/ESP security association
- Phase 1 is comparable to SSL session
- Phase 2 is comparable to SSL connection
- Not an obvious need for two phases in IKE
- If multiple Phase 2's do not occur, then it is **more** expensive to have two phases!

IKE Phase 1

- Four different "key" options
  - Public key encryption (original version)
  - Public key encryption (improved version)
  - Public key signature
  - Symmetric key
- For each of these, two different "modes"
  - Main mode
  - Aggressive mode
- **There are 8 versions of IKE Phase 1!**
- Evidence that IPSec is over-engineered?

IKE Phase 1

- We'll discuss 6 of 8 phase 1 variants
  - Public key signatures (main and aggressive modes)
  - Symmetric key (main and aggressive modes)
  - Public key encryption (main and aggressive)
- Why public key encryption and public key signatures?
  - Always know your own private key
  - **May not** (initially) know other side's public key

IKE Phase 1

- Uses ephemeral Diffie-Hellman to establish session key
  - Achieves perfect forward secrecy (PFS)
- Let a be Alice's Diffie-Hellman exponent
- Let b be Bob's Diffie-Hellman exponent
- Let g be generator and p prime
- Recall p and g are public

IKE Phase 1: Digital Signature (Main Mode)

Alice

- Alice → Bob: IC, CP
- Bob → Alice: IC,RC, CS
- Alice → Bob: IC,RC, g^a mod p, R_A
- Bob → Alice: IC,RC, g^b mod p, R_B
- Alice → Bob: IC,RC, E("Alice", proof_A, K)
- Bob → Alice: IC,RC, E("Bob", proof_B, K)

Bob

- CP = crypto proposed, CS = crypto selected
- IC = initiator "cookie", RC = responder "cookie"
- K = h(IC,RC,g^{ab} mod p,R_A,R_B)
- SKEYID = h(R_A, R_B, g^{ab} mod p)
- proof_A = [h(SKEYID,g^a,g^b,IC,RC,CP,"Alice")]$_{\text{Alice}}$

IKE Phase 1: Public Key Signature (Aggressive Mode)

Alice

IC, "Alice", g^a mod p, R_A, CP →

← IC,RC, "Bob", R_B, g^b mod p, CS, proof_B

IC,RC, proof_A →

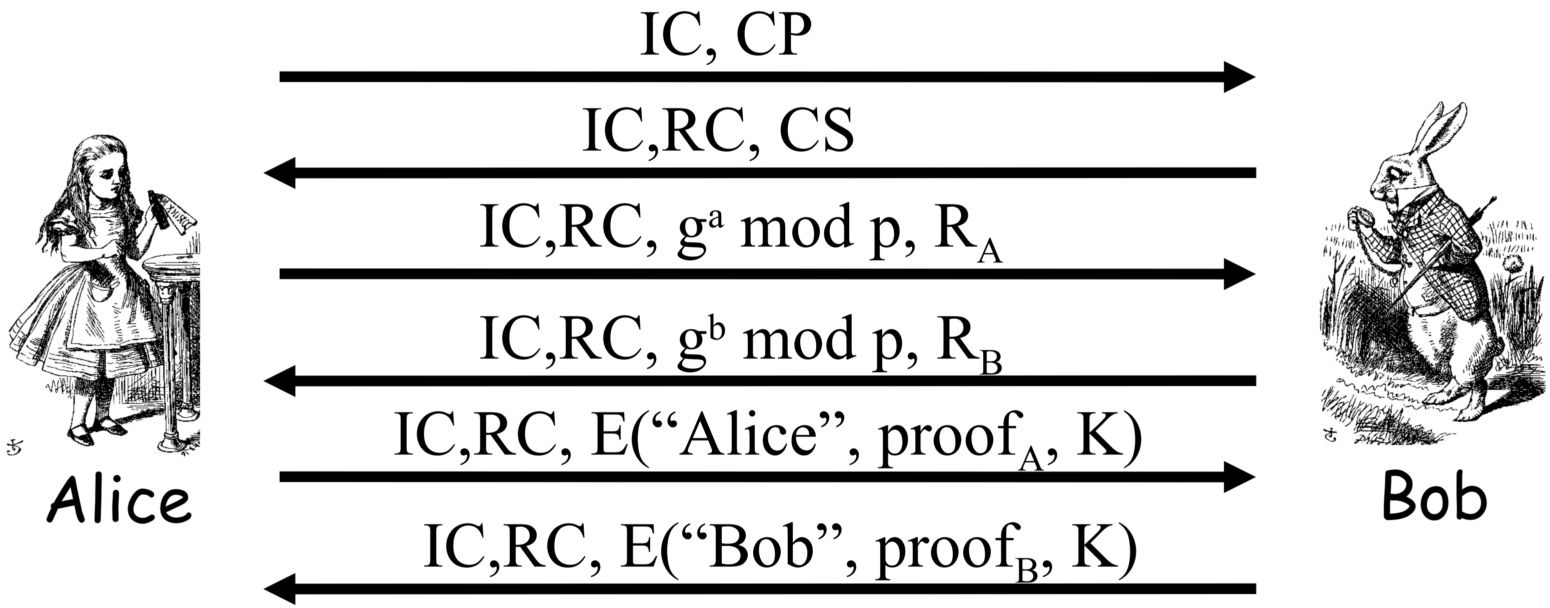
Bob

- Main difference from main mode
  - Not trying to protect identities
  - Cannot negotiate g or p

Main vs Aggressive Modes

- Main mode **MUST** be implemented
- Aggressive mode **SHOULD** be implemented
  - In other words, if aggressive mode is not implemented, "you should feel guilty about it"
- Might create interoperability issues
- For public key signature authentication
  - Passive attacker knows identities of Alice and Bob in aggressive mode
  - Active attacker can determine Alice's and Bob's identity in main mode

IKE Phase 1: Symmetric Key (Main Mode)

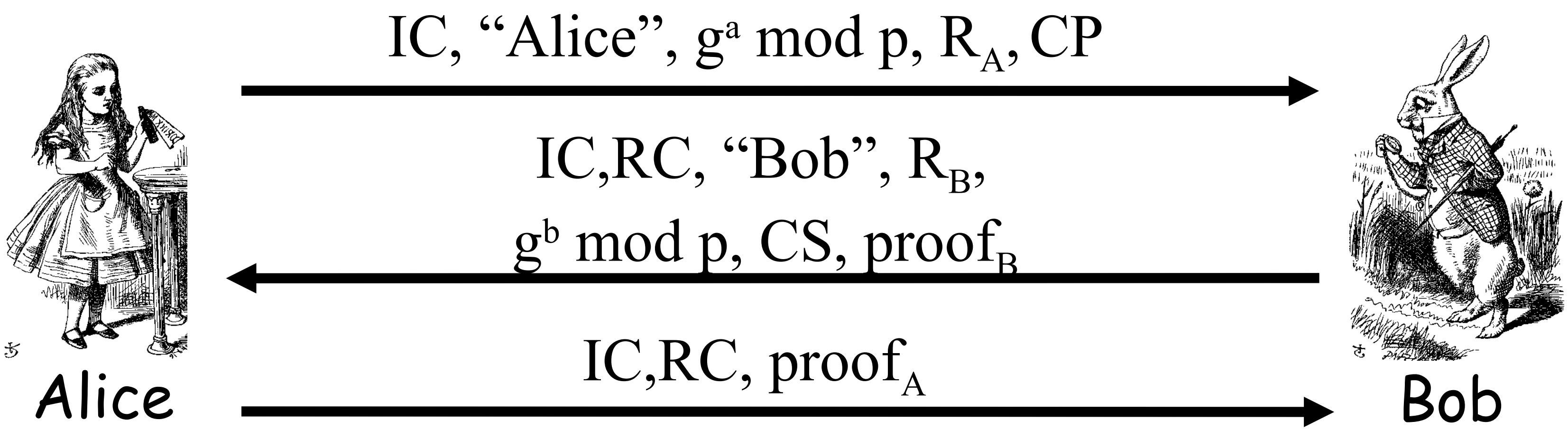


- Same as signature mode except
  - K_{AB} = symmetric key shared in advance
  - $K = h(IC,RC,g^{ab} \bmod p,R_A,R_B,K_{AB})$
  - $SKEYID = h(K, g^{ab} \bmod p)$
  - $\text{proof}_A = h(SKEYID,g^a,g^b,IC,RC,CP,\text{“Alice”})$

Problems with Symmetric Key (Main Mode)

- Catch-22
  - Alice sends her ID in message 5
  - Alice's ID encrypted with K
  - To find K Bob must know K_{AB}
  - To get K_{AB} Bob must know he's talking to Alice!
- Result: **Alice's ID must be IP address!**
- Useless mode for the "road warrior"
- Why go to all of the trouble of trying to hide identities in 6 message protocol?

IKE Phase 1: SymmetricKey (Aggressive Mode)



- Same format as digital signature aggressive mode
- Not trying to hide identities…
- As a result, does **not** have problems of main mode
- But does not (pretend to) hide identities

IKE Phase 1: Public Key Encryption (Main Mode)

Alice

IC, CP →

← IC,RC, CS

IC,RC, g^a mod p, $\{R_A\}_{Bob}$, $\{\text{“Alice”}\}_{Bob}$ →

← IC,RC, g^b mod p, $\{R_B\}_{Alice}$, $\{\text{“Bob”}\}_{Alice}$

IC,RC, E($proof_A$, K) →

← IC,RC, E($proof_B$, K)

Bob

- CP = crypto proposed, CS = crypto selected
- IC = initiator “cookie”, RC = responder “cookie”
- $K = h(IC,RC,g^{ab} \bmod p,R_A,R_B)$
- $SKEYID = h(R_A, R_B, g^{ab} \bmod p)$
- $proof_A = h(SKEYID,g^a,g^b,IC,RC,CP,\text{“Alice”})$

IKE Phase 1: Public Key Encryption (Aggressive Mode)

$IC, CP, g^a \bmod p, \{\text{“Alice”}\}_{Bob}, \{R_A\}_{Bob}$ →

Alice

$IC, RC, CS, g^b \bmod p, \{\text{“Bob”}\}_{Alice}, \{R_B\}_{Alice}, proof_B$ ←

$IC, RC, proof_A$ →

Bob

- K, $proof_A$, $proof_B$ computed as in main mode
- Note that identities are hidden
  - The only aggressive mode to hide identities
  - Then why have main mode?

Public Key Encryption Issue?

- Public key encryption, aggressive mode
- Suppose **Trudy** generates
  - Exponents **a** and **b**
  - Nonces $\mathbf{R_A}$ and $\mathbf{R_B}$
- Trudy can compute "valid" keys and proofs: $\mathbf{g^{ab} \bmod p}$, **K**, **SKEYID**, $\mathbf{proof_A}$ and $\mathbf{proof_B}$
- Also true of main mode

Public Key Encryption Issue?

Trudy as Alice

IC, CP, $g^a \bmod p$, $\{\text{"Alice"}\}_{Bob}$, $\{R_A\}_{Bob}$ →

← IC,RC, CS, $g^b \bmod p$, $\{\text{"Bob"}\}_{Alice}$, $\{R_B\}_{Alice}$, $proof_B$

IC,RC, $proof_A$ →

Trudy as Bob

- Trudy can create exchange that appears to be between Alice and Bob
- Appears valid to any observer, **including Alice and Bob!**

Plausible Deniability

- Trudy can create "conversation" that appears to be between Alice and Bob
- Appears valid, even to Alice and Bob!
- A security failure?
- In this mode of IPSec, it is a feature
  - **Plausible deniability:** Alice and Bob can deny that any conversation took place!
- In some cases it might be a security failure
  - If Alice makes a purchase from Bob, she could later repudiate it (unless she had signed)

IKE Phase 1 Cookies

- Cookies (or "anti-clogging tokens") supposed to make denial of service more difficult
- No relation to Web cookies
- To reduce DoS, Bob wants to remain stateless as long as possible
- But Bob must remember CP from message 1 (required for proof of identity in message 6)
- Bob must keep state from 1st message on!
- These cookies offer little DoS protection!

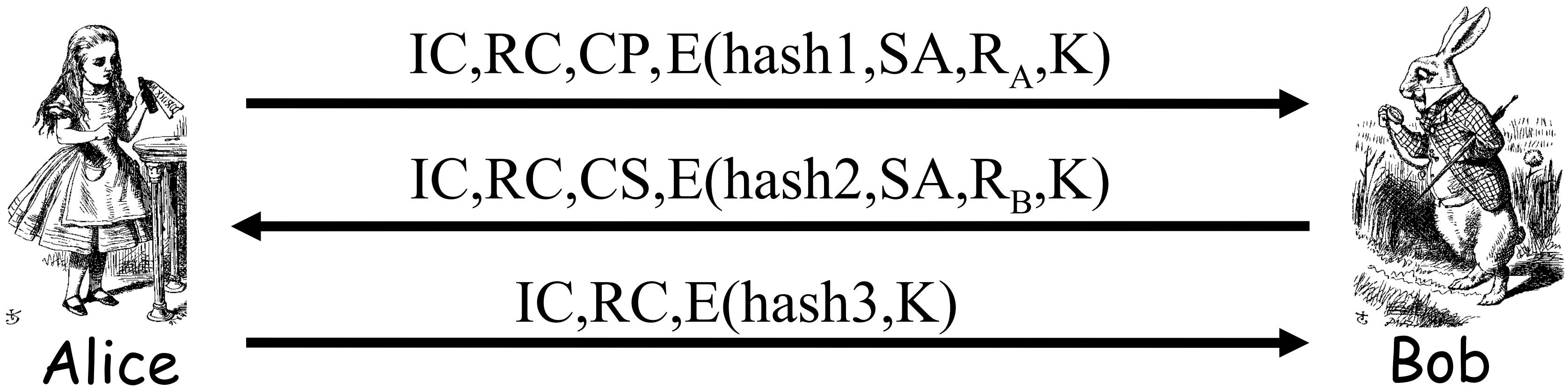
IKE Phase 1 Summary

- Result of IKE phase 1 is
  - Mutual authentication
  - Shared symmetric key
  - IKE **Security Association (SA)**
- But phase 1 is expensive (in public key and/or main mode cases)
- Developers of IKE thought it would be used for lots of things — not just IPSec
- Partly explains over-engineering…

IKE Phase 2

- Phase 1 establishes IKE SA
- Phase 2 establishes IPSec SA
- Comparison to SSL
  - SSL session is comparable to IKE Phase 1
  - SSL connections are like IKE Phase 2
- IKE **could** be used for lots of things
- But in practice, it's **not!**

IKE Phase 2



- Key K, IC, RC and SA known from Phase 1
- Proposal CP includes ESP and/or AH
- Hashes 1,2,3 depend on SKEYID, SA, R_A and R_B
- Keys derived from KEYMAT = h(SKEYID,R_A,R_B,junk)
- Recall SKEYID depends on phase 1 key method
- Optional PFS (ephemeral Diffie-Hellman exchange)

IPSec

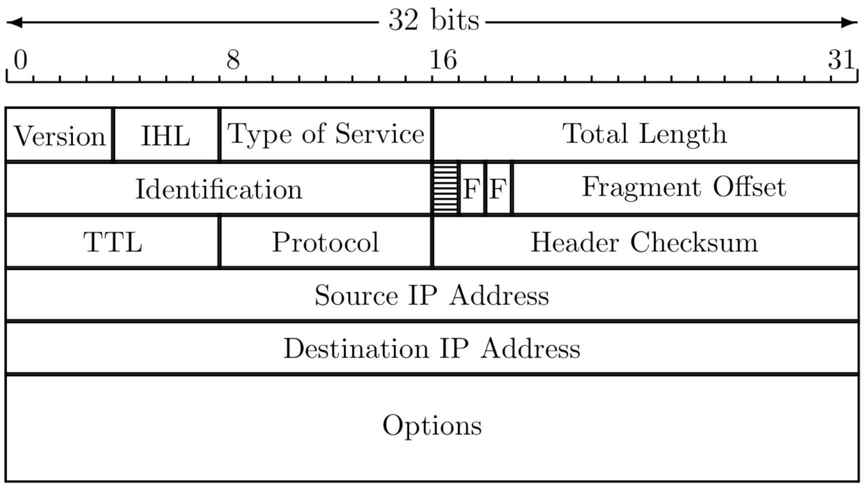
- After IKE Phase 1, we have an IKE SA
- After IKE Phase 2, we have an IPSec SA
- Both sides have a shared symmetric key
- Now what?
  - We want to protect IP datagrams
- But what is an IP datagram?
  - From the perspective of IPSec…

IP Review

- IP datagram is of the form

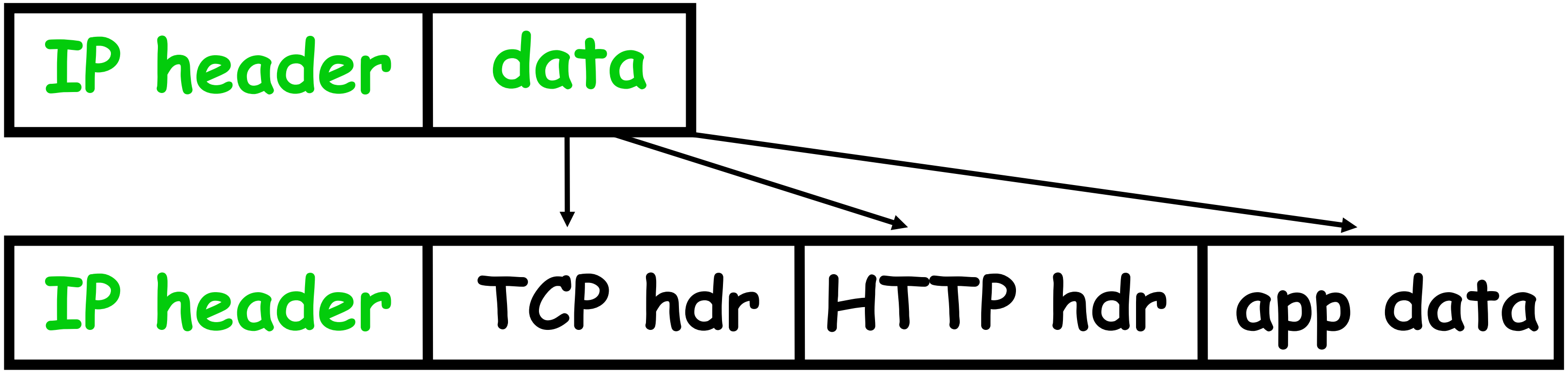
IP header	data

- Where IP header is



IP and TCP

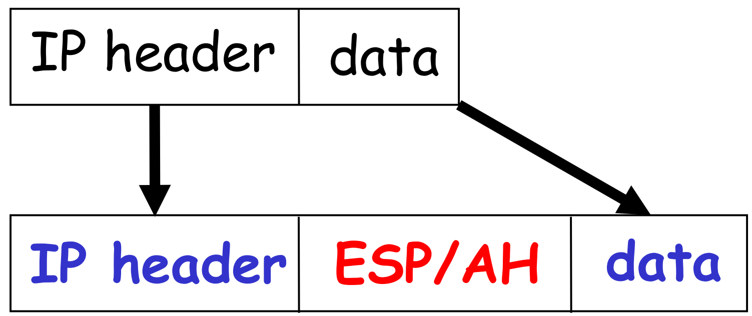
- Consider HTTP traffic (over TCP)
- IP encapsulates TCP
- TCP encapsulates HTTP



- IP data includes TCP header, etc.

IPSec Transport Mode

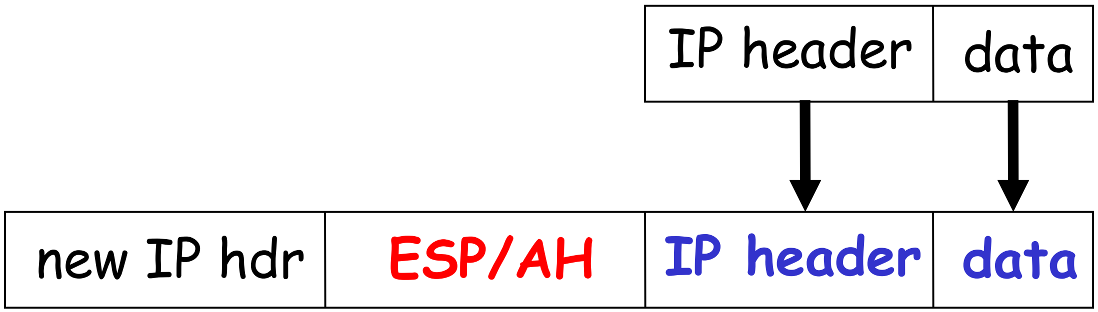
- IPSec **Transport Mode**



- Transport mode designed for host-to-host
- Transport mode is efficient
  - Adds minimal amount of extra header
- The original header remains
  - Passive attacker can see who is talking

IPSec Tunnel Mode

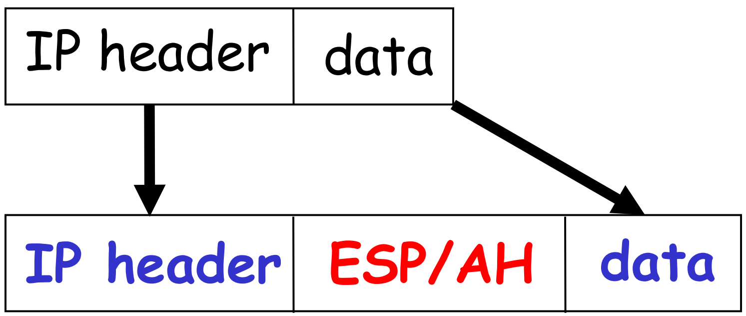
- IPSec **Tunnel Mode**



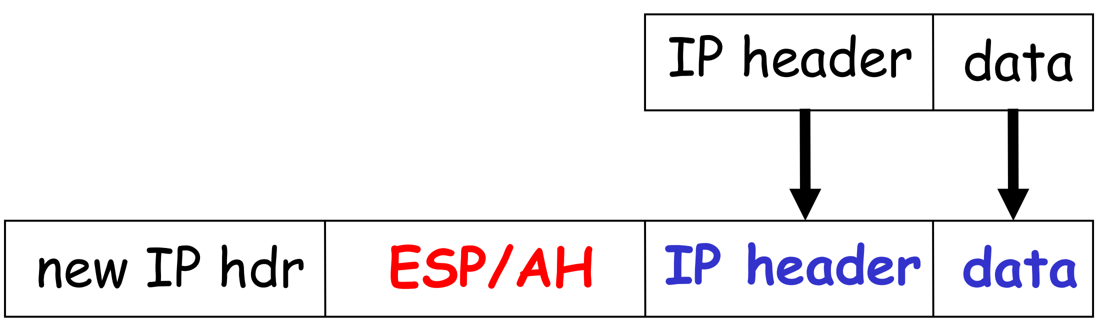
- Tunnel mode for firewall to firewall traffic
- Original IP packet encapsulated in IPSec
- Original IP header not visible to attacker
  - New header from firewall to firewall
  - Attacker does not know which hosts are talking

Comparison of IPSec Modes

- Transport Mode



- Tunnel Mode



- Transport Mode
  - Host-to-host
- Tunnel Mode
  - Firewall-to-firewall
- Transport mode not necessary
- Transport mode is more efficient

IPSec Security

- What kind of protection?
  - Confidentiality?
  - Integrity?
  - Both?
- What to protect?
  - Data?
  - Header?
  - Both?
- ESP/AH do some combinations of these

AH vs ESP

- AH
  - Authentication Header
  - **Integrity only** (no confidentiality)
  - Integrity-protect everything beyond IP header and some fields of header (why not all fields?)
- ESP
  - Encapsulating Security Payload
  - **Integrity and confidentiality**
  - Protects everything beyond IP header
  - Integrity only by using NULL encryption

ESP's NULL Encryption

- According to RFC 2410
  - NULL encryption "is a block cipher the origins of which appear to be lost in antiquity"
  - "Despite rumors", there is no evidence that NSA "suppressed publication of this algorithm"
  - Evidence suggests it was developed in Roman times as exportable version of Caesar's cipher
  - Can make use of keys of varying length
  - No IV is required
  - Null(P,K) = P for any P and any key K
- Security people have a strange sense of humor!

Why Does AH Exist? (1)

- Cannot encrypt IP header
  - Routers must look at the IP header
  - IP addresses, TTL, etc.
  - IP header exists to route packets!
- AH protects **immutable fields** in IP header
  - Cannot integrity protect all header fields
  - TTL, for example, must change
- ESP does not protect IP header at all

Why Does AH Exist? (2)

- ESP encrypts everything beyond the IP header (if non-null encryption)
- If ESP encrypted, firewall cannot look at TCP header (e.g., port numbers)
- Why not use ESP with null encryption?
  - Firewall sees ESP header, but does not know whether null encryption is used
  - End systems know, but **not** firewalls
- Aside 1: Do firewalls reduce security?
- Aside 2: Is IPSec compatible with NAT?

Why Does AH Exist? (3)

- The real reason why AH exists
  - At one IETF meeting "someone from Microsoft gave an impassioned speech about how AH was useless..."
  - "...everyone in the room looked around and said `Hmm. He's right, and we hate AH also, but if it annoys Microsoft let's leave it in since we hate Microsoft more than we hate AH."

University of Amsterdam

Jaap van Ginkel

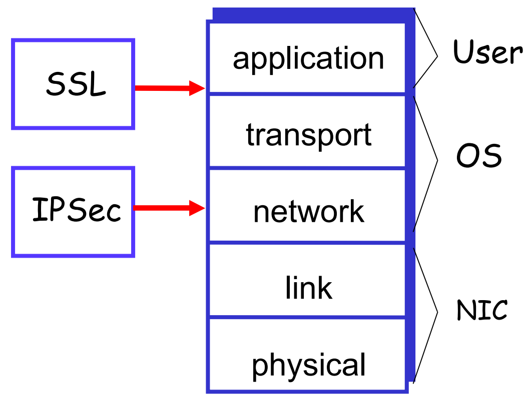
Security of Systems and Networks

October 12, 2019 IPsec

IPSec

IPSec and SSL

- IPSec lives at the network layer
- IPSec is transparent to applications



IPSec and Complexity

- IPSec is a complex protocol
- Over-engineered
  - Lots of generally useless extra features
- Flawed
  - Some serious security flaws
- Interoperability is serious challenge
  - Defeats the purpose of having a standard!
- Complex
- Did I mention, it's complex?

IKE and ESP/AH

- Two parts to IPSec
- **IKE:** Internet Key Exchange
  - Mutual authentication
  - Establish shared symmetric key
  - Two “phases” — like SSL session/connection
- **ESP/AH**
  - ESP: Encapsulating Security Payload — for encryption and/or integrity of IP packets
  - AH: Authentication Header — integrity only

IKE

IKE

- IKE has 2 phases
  - Phase 1 — IKE security association (SA)
  - Phase 2 — AH/ESP security association
- Phase 1 is comparable to SSL session
- Phase 2 is comparable to SSL connection
- Not an obvious need for two phases in IKE
- If multiple Phase 2's do not occur, then it is **more** expensive to have two phases!

IKE Phase 1

- Four different "key" options
  - Public key encryption (original version)
  - Public key encryption (improved version)
  - Public key signature
  - Symmetric key
- For each of these, two different "modes"
  - Main mode
  - Aggressive mode
- **There are 8 versions of IKE Phase 1!**
- Evidence that IPSec is over-engineered?

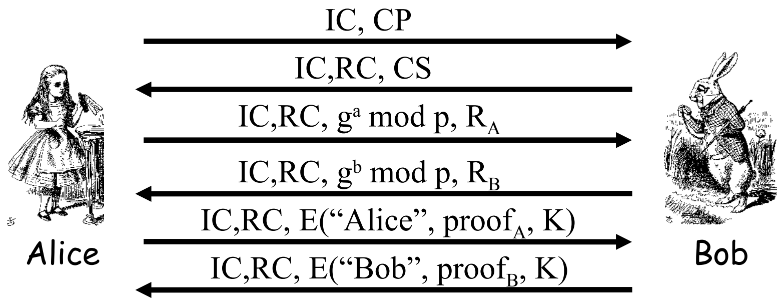
IKE Phase 1

- We'll discuss 6 of 8 phase 1 variants
  - Public key signatures (main and aggressive modes)
  - Symmetric key (main and aggressive modes)
  - Public key encryption (main and aggressive)
- Why public key encryption and public key signatures?
  - Always know your own private key
  - **May not** (initially) know other side's public key

IKE Phase 1

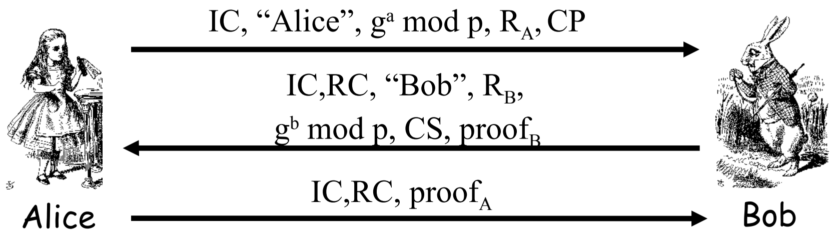
- Uses ephemeral Diffie-Hellman to establish session key
  - Achieves perfect forward secrecy (PFS)
- Let a be Alice's Diffie-Hellman exponent
- Let b be Bob's Diffie-Hellman exponent
- Let g be generator and p prime
- Recall p and g are public

IKE Phase 1: Digital Signature (Main Mode)



- CP = crypto proposed, CS = crypto selected
- IC = initiator "cookie", RC = responder "cookie"
- $K = h(IC,RC,g^{ab} \bmod p,R_A,R_B)$
- $SKEYID = h(R_A, R_B, g^{ab} \bmod p)$
- $proof_A = [h(SKEYID,g^a,g^b,IC,RC,CP,\text{"Alice"})]_{Alice}$

IKE Phase 1: Public Key Signature (Aggressive Mode)



- Main difference from main mode
  - Not trying to protect identities
  - Cannot negotiate g or p

Main vs Aggressive Modes

- Main mode **MUST** be implemented
- Aggressive mode **SHOULD** be implemented
  - In other words, if aggressive mode is not implemented, "you should feel guilty about it"
- Might create interoperability issues
- For public key signature authentication
  - Passive attacker knows identities of Alice and Bob in aggressive mode
  - Active attacker can determine Alice's and Bob's identity in main mode

IKE Phase 1: Symmetric Key (Main Mode)

Alice

IC, CP →

← IC,RC, CS

IC,RC, g^a mod p, R_A →

← IC,RC, g^b mod p, R_B

IC,RC, E("Alice", $proof_A$, K) →

← IC,RC, E("Bob", $proof_B$, K)

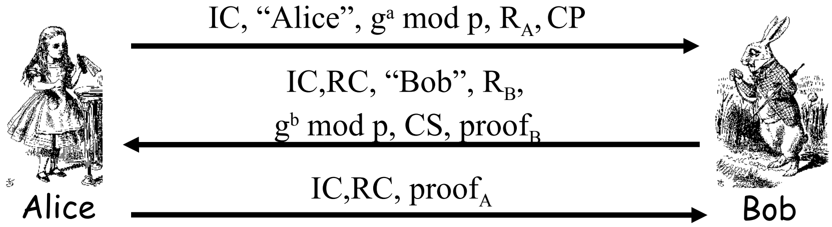
Bob

- Same as signature mode except
  - K_{AB} = symmetric key shared in advance
  - K = h(IC,RC,g^{ab} mod p,R_A,R_B,K_{AB})
  - SKEYID = h(K, g^{ab} mod p)
  - $proof_A$ = h(SKEYID,g^a,g^b,IC,RC,CP,"Alice")

Problems with Symmetric Key (Main Mode)

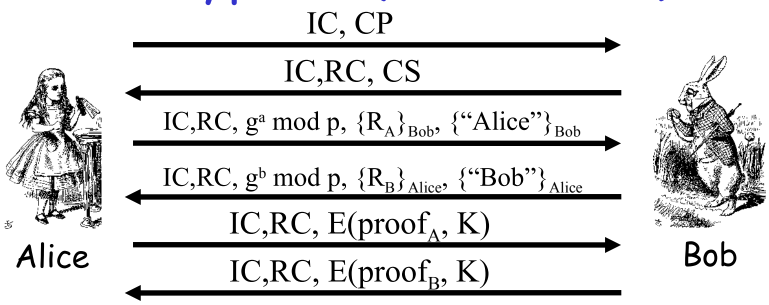
- Catch-22
  - Alice sends her ID in message 5
  - Alice's ID encrypted with K
  - To find K Bob must know K_{AB}
  - To get K_{AB} Bob must know he's talking to Alice!
- Result: **Alice's ID must be IP address!**
- Useless mode for the "road warrior"
- Why go to all of the trouble of trying to hide identities in 6 message protocol?

IKE Phase 1: SymmetricKey (Aggressive Mode)



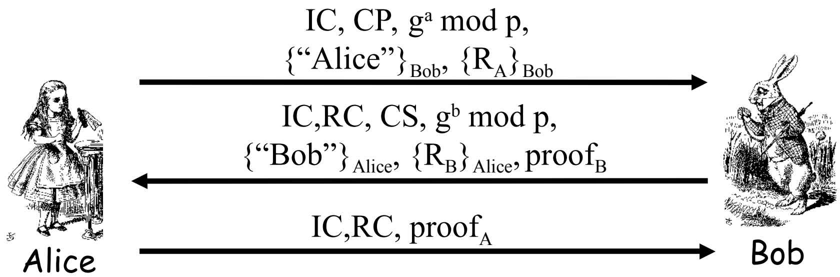
- Same format as digital signature aggressive mode
- Not trying to hide identities…
- As a result, does **not** have problems of main mode
- But does not (pretend to) hide identities

IKE Phase 1: Public Key Encryption (Main Mode)



- CP = crypto proposed, CS = crypto selected
- IC = initiator “cookie”, RC = responder “cookie”
- K = h(IC,RC,g^{ab} mod p,R_A,R_B)
- SKEYID = h(R_A, R_B, g^{ab} mod p)
- $proof_A$ = h(SKEYID,g^a,g^b,IC,RC,CP,“Alice”)

IKE Phase 1: Public Key Encryption (Aggressive Mode)



- K, proof$_A$, proof$_B$ computed as in main mode
- Note that identities are hidden
  - The only aggressive mode to hide identities
  - Then why have main mode?

Public Key Encryption Issue?

- Public key encryption, aggressive mode
- Suppose **Trudy** generates
  - Exponents **a** and **b**
  - Nonces R_A and R_B
- Trudy can compute "valid" keys and proofs: g^{ab} **mod p**, **K**, **SKEYID**, $proof_A$ and $proof_B$
- Also true of main mode

Public Key Encryption Issue?

Trudy as Alice

IC, CP, g^a mod p, {"Alice"}$_{Bob}$, {R_A}$_{Bob}$ →

← IC,RC, CS, g^b mod p, {"Bob"}$_{Alice}$, {R_B}$_{Alice}$, proof$_B$

IC,RC, proof$_A$ →

Trudy as Bob

- Trudy can create exchange that appears to be between Alice and Bob
- Appears valid to any observer, ***including Alice and Bob!***

Plausible Deniability

- Trudy can create "conversation" that appears to be between Alice and Bob
- Appears valid, even to Alice and Bob!
- A security failure?
- In this mode of IPSec, it is a feature
  - **Plausible deniability:** Alice and Bob can deny that any conversation took place!
- In some cases it might be a security failure
  - If Alice makes a purchase from Bob, she could later repudiate it (unless she had signed)

IKE Phase 1 Cookies

- Cookies (or "anti-clogging tokens") supposed to make denial of service more difficult
- No relation to Web cookies
- To reduce DoS, Bob wants to remain stateless as long as possible
- But Bob must remember CP from message 1 (required for proof of identity in message 6)
- Bob must keep state from 1st message on!
- These cookies offer little DoS protection!

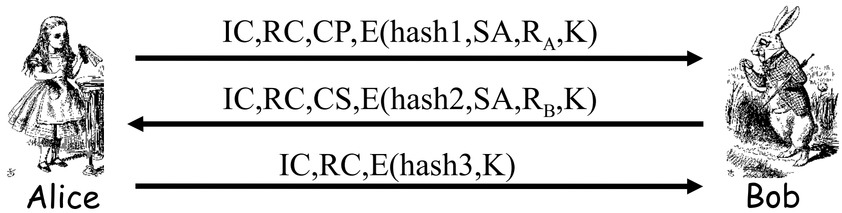
IKE Phase 1 Summary

- Result of IKE phase 1 is
  - Mutual authentication
  - Shared symmetric key
  - IKE ***Security Association* (SA)**
- But phase 1 is expensive (in public key and/or main mode cases)
- Developers of IKE thought it would be used for lots of things — not just IPSec
- Partly explains over-engineering…

IKE Phase 2

- Phase 1 establishes IKE SA
- Phase 2 establishes IPSec SA
- Comparison to SSL
  - SSL session is comparable to IKE Phase 1
  - SSL connections are like IKE Phase 2
- IKE **could** be used for lots of things
- But in practice, it's **not!**

IKE Phase 2



- Key K, IC, RC and SA known from Phase 1
- Proposal CP includes ESP and/or AH
- Hashes 1,2,3 depend on SKEYID, SA, R_A and R_B
- Keys derived from KEYMAT = h(SKEYID,R_A,R_B,junk)
- Recall SKEYID depends on phase 1 key method
- Optional PFS (ephemeral Diffie-Hellman exchange)

IPSec

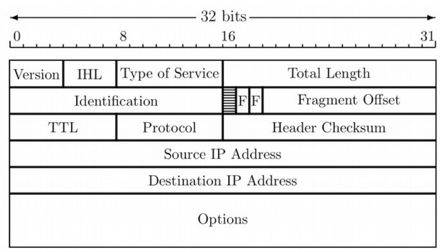
- After IKE Phase 1, we have an IKE SA
- After IKE Phase 2, we have an IPSec SA
- Both sides have a shared symmetric key
- Now what?
  - We want to protect IP datagrams
- But what is an IP datagram?
  - From the perspective of IPSec…

IP Review

- IP datagram is of the form

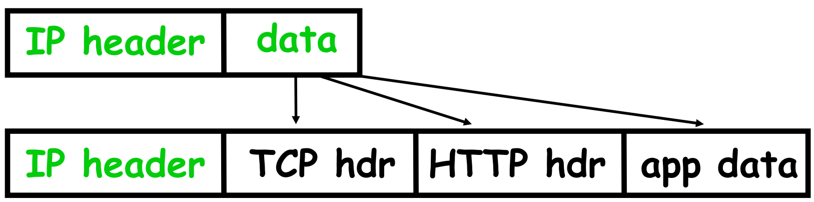
IP header	data

- Where IP header is



IP and TCP

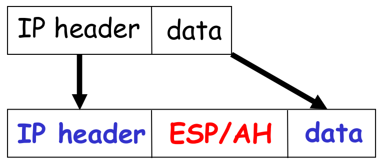
- Consider HTTP traffic (over TCP)
- IP encapsulates TCP
- TCP encapsulates HTTP



- IP data includes TCP header, etc.

IPSec Transport Mode

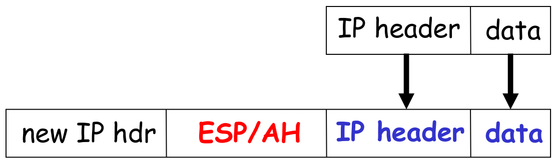
- IPSec **Transport Mode**



- Transport mode designed for host-to-host
- Transport mode is efficient
  - Adds minimal amount of extra header
- The original header remains
  - Passive attacker can see who is talking

IPSec Tunnel Mode

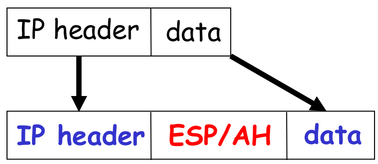
- IPSec **Tunnel Mode**



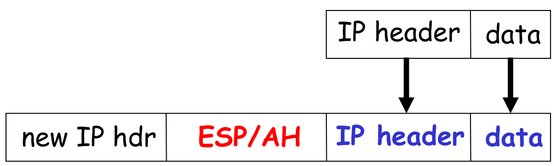
- Tunnel mode for firewall to firewall traffic
- Original IP packet encapsulated in IPSec
- Original IP header not visible to attacker
  - New header from firewall to firewall
  - Attacker does not know which hosts are talking

Comparison of IPSec Modes

- Transport Mode



- Tunnel Mode



- Transport Mode
  - Host-to-host
- Tunnel Mode
  - Firewall-to-firewall
- Transport mode not necessary
- Transport mode is more efficient

IPSec Security

- What kind of protection?
  - Confidentiality?
  - Integrity?
  - Both?
- What to protect?
  - Data?
  - Header?
  - Both?
- ESP/AH do some combinations of these

AH vs ESP

- AH
  - Authentication Header
  - **Integrity only** (no confidentiality)
  - Integrity-protect everything beyond IP header and some fields of header (why not all fields?)
- ESP
  - Encapsulating Security Payload
  - **Integrity and confidentiality**
  - Protects everything beyond IP header
  - Integrity only by using NULL encryption

ESP's NULL Encryption

- According to RFC 2410
  - NULL encryption "is a block cipher the origins of which appear to be lost in antiquity"
  - "Despite rumors", there is no evidence that NSA "suppressed publication of this algorithm"
  - Evidence suggests it was developed in Roman times as exportable version of Caesar's cipher
  - Can make use of keys of varying length
  - No IV is required
  - Null(P,K) = P for any P and any key K
- Security people have a strange sense of humor!

Why Does AH Exist? (1)

- Cannot encrypt IP header
  - Routers must look at the IP header
  - IP addresses, TTL, etc.
  - IP header exists to route packets!
- AH protects **immutable fields** in IP header
  - Cannot integrity protect all header fields
  - TTL, for example, must change
- ESP does not protect IP header at all

Why Does AH Exist? (2)

- ESP encrypts everything beyond the IP header (if non-null encryption)
- If ESP encrypted, firewall cannot look at TCP header (e.g., port numbers)
- Why not use ESP with null encryption?
  - Firewall sees ESP header, but does not know whether null encryption is used
  - End systems know, but **not** firewalls
- Aside 1: Do firewalls reduce security?
- Aside 2: Is IPSec compatible with NAT?

Why Does AH Exist? (3)

- The real reason why AH exists
  - At one IETF meeting "someone from Microsoft gave an impassioned speech about how AH was useless…"
  - "…everyone in the room looked around and said `Hmm. He's right, and we hate AH also, but if it annoys Microsoft let's leave it in since we hate Microsoft more than we hate AH."