

## Application of Cryptography: IPsec & IKE

Guevara Noubir  
<http://www.ccs.neu.edu/home/noubir/Courses/CSG252/F04>

Textbook: "Network Security",  
Kauffman – Perlman - Speciner

Reading: Chapters 17-18

---

---

---


---

---

---

---

---



## Outline

- Introduction to secure networking
- IPsec:
  - Authentication Header and Encapsulated Secure Payload
- Internet Key Exchange (IKE)

Fall'04: CSG252      Classical Cryptography      2

---

---

---


---

---

---

---

---



## Approach to Secure Networking

- Key Distribution Centers: Trusted Third Parties

or

- Public Key Systems + Public Key Infrastructure

Fall'04: CSG252      Classical Cryptography      3

---

---

---

---

---

---

---

---

## Key Distribution Center

- n Solve the scalability problem of a set of  $n$  nodes using secret key
  - o  $n*(n-1)/2$  keys
- n New nodes are configured with a key to the KDC
  - o e.g.,  $K_A$  for node  $A$
- n If node  $A$  wants to communicate with node  $B$ 
  - o  $A$  sends a request to the KDC
  - o The KDC securely sends to  $A$ :  $E_{K_A}(R_{AB})$  and  $E_{K_B}(R_{AB}, A, B)$
- n Advantage:
  - o Single location for updates, single key to be remembered
- n Drawbacks:
  - o If the KDC is compromised!
  - o Single point of failure/performance bottleneck => multiple KDC?
- n Example of systems: Kerberos

Fall'04: CSG252 Classical Cryptography 4

---

---

---

---

---

---

---

---

## Public Keys and Certification Authorities

- n How do you know the public key of a node?
- n Typical solution:
  - o Use a trusted node as a certification authority (CA)
  - o The CA generates certificates: Signed( $A$ , public-key, validity information)
  - o Every body needs to know the CA public key
  - o Certificates can be stored in a directory service or exchanged during the authentication process
- n Advantages:
  - o The CA doesn't have to be online => more physical protection
  - o Not a performance bottleneck, not a single point of failure
  - o Certificates are not security sensitive: only threat is DoS
  - o A compromised CA cannot decrypt conversation but can lead to impersonation
  - o A certification hierarchy can be used: e.g., X.509

Fall'04: CSG252 Classical Cryptography 5

---

---

---

---

---

---

---

---

## Securing Network Stacks

- n Where to put the security in a protocol stack?
- n Practical considerations:
  - o End to end security
  - o No modification to OS/network stack

Control/Management (configuration)	Applications Layer telnet/ftp, http, <b>shttp</b> , mail: PGP	Network Security Tools: Monitoring/Logging/Intrusion Detection
	(SSL/TLS, SSH)	
	Transport Layer (TCP)	
	(IPSec, IKE)	
	Network Layer (IP)	
	Link Layer (IEEE802.1x/IEEE802.10)	
Physical Layer (spread-Spectrum, quantum crypto, etc.)		

Fall'04: CSG252 Classical Cryptography 6

---

---

---

---

---

---

---

---

## SSL vs. IPsec

- n SSL:
  - o Avoids modifying "TCP stack" and requires minimum changes to the application
  - o Mostly used to authenticate servers
- n IPsec
  - o Transparent to the application and requires modification of the network stack
  - o Authenticates network nodes and establishes a secure channel between nodes
  - o Application still needs to authenticate the users

Fall'04: CS6252 Classical Cryptography 7

---

---

---

---

---

---

---

---

## Some Issues with Real-time Communication

- n Session key establishment
- n Perfect Forward Secrecy
  - o Diffie-Hellman based PFS
  - o Escrow-foilage:
    - o If keys are escrowed Diffie-Hellman protects against passive attacks
    - o Signature keys are usually not escrowed
- n Preventing Denial of Service
  - o SYN attack on TCP: use stateless cookies = hash(IP addr, secret)
  - o Puzzles: e.g., what 27-bit number has an MD = x?
  - o These techniques do not fully protect against DDOS launched through viruses
- n Hiding endpoint identity:
  - o DH + authentication allows anonymous connection or detects man-in-the-middle
- n Live partner reassurance:
  - o Modify DH to include a nonce in the computation of the session key
- n Optimization using parallel computation, session resumption, deniability

Fall'04: CS6252 Classical Cryptography 8

---

---

---

---

---

---

---

---

## IPsec Protocol Suite (IETF Standard)

- n Provides inter-operable cryptographically based security services:
  - o Services: confidentiality, authentication, integrity, and key management
  - o Protocols:
    - o Authentication Header (AH): RFC2402
    - o Encapsulated Security Payload (ESP): RFC2406
    - o Internet Key Exchange (IKE)
  - o Environments: IPv4 and IPv6
  - o Modes:
    - o Transport (between two hosts)
    - o Tunnel (between hosts/firewalls)

Fall'04: CS6252 Classical Cryptography 9

---

---

---

---

---

---

---

---

## IPsec

- n Assumption:
  - End nodes already established a shared session key (manually or IKE)
- n Security Association:
  - Each secure connection is called a *security association (SA)*
  - For each SA: key, end-node, sequence number, services, algorithms
  - SA is unidirectional and identified by (destination-address, SPI = Security Parameter Index)
- n Protocols:
  - Authentication Header: integrity protection
  - Encapsulated Security Payload: encryption and/or integrity

---

---

---

---

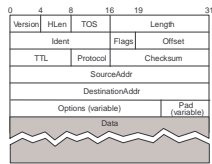
---

---

---

---

## IP Packets




---

---

---

---

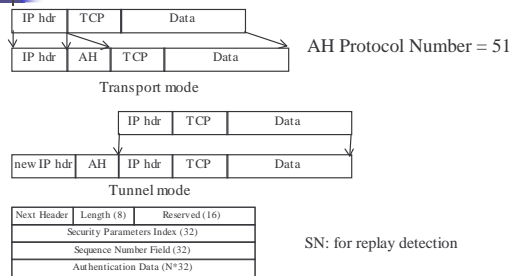
---

---

---

---

## AH Formatting




---

---

---

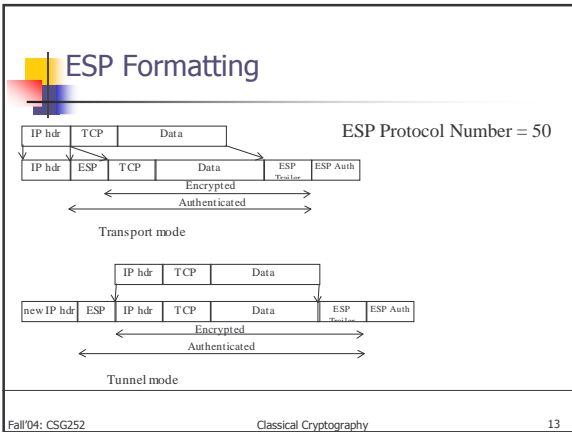
---

---

---

---

---




---

---

---

---

---

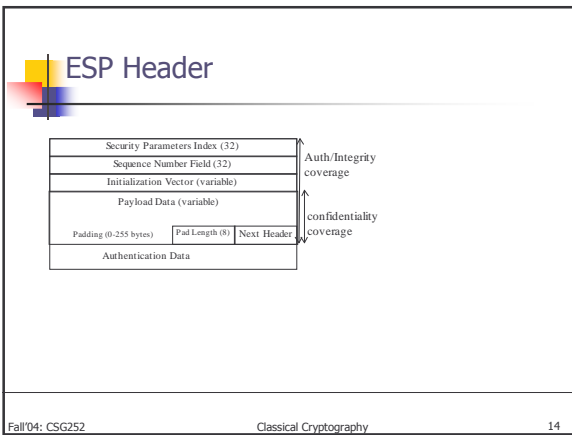
---

---

---

---

---




---

---

---

---

---

---

---

---

---

---

- ### IPsec: Internet Key Exchange
- Goal:
    - Mutual authentication and establishment of a shared secret session key using:
      - Pre-shared secret key or public signature-only key, or public encryption key
    - Negotiation of features and cryptographic algorithms
  - Specification documents:
    - ISAKMP (Internet Security Association and Key Management Protocol): RFC 2408
    - IKE: RFC 2409
    - DOI (Domain Of Interpretation): RFC 2407
- Fall'04: CSG252 Classical Cryptography 15

---

---

---

---

---

---

---

---

---

---

## Photuris

- n Photuris goal: signed Diffie-Hellman exchange
  1.  $A \rightarrow B: C_A$
  2.  $B \rightarrow A: C_A, C_B$ , crypto offered
  3.  $A \rightarrow B: C_A, C_B, g^a \text{ mod } p$ , crypto selected
  4.  $B \rightarrow A: C_A, C_B, g^b \text{ mod } p$
  5.  $A \rightarrow B: C_A, C_B, g^{ab} \text{ mod } p$ {A, signature of previous message}
  6.  $B \rightarrow A: C_A, C_B, g^{ab} \text{ mod } p$ {B, signature of previous message}
  - n Role of  $C_A, C_B$ , and messages
  - n Additional features: SPI selection
  - n Why not sign messages 3 & 4...?

Fall'04: CS6252 Classical Cryptography 16

---

---

---

---

---

---

---

---

## Simple Key-Management for Internet Protocol (SKIP)

- n Uses long term Diffie-Hellman keys
- n Parties assumed to know each other public keys (i.e.,  $g^p \text{ mod } p$ ) or exchange certificates
- n Session key  $X = g^{ab} \text{ mod } p$  is established in 0 messages
- n Each packet is encrypted using data key  $S$  and each packet contains:  $X\{S\}$ 
  - n Same  $S$  can be used for several packets
- n Later on PFS was added by periodically forgetting the keys and doing a new DH

Fall'04: CS6252 Classical Cryptography 17

---

---

---

---

---

---

---

---

## ISAKMP (RFC2408)

- n Proposed by NSA as a framework and accepted by IETF
  - n Runs over UDP and allows to exchange fields to create a protocol
- n IKE (RFC2409) based on OAKLEY & SKEME using ISAKMP syntax
- n IKE phases:
  1. Mutual authentication and session key establishment (also called ISAKMP SA or IKE SA)
  2. AH/ESP SAs establishment
  - Each source/destination/port has its own SA/keys otherwise ESP traffic not using integrity could be decrypted...
  - IKE uses default port 500

Fall'04: CS6252 Classical Cryptography 18

---

---

---

---

---

---

---

---

## Phase 1 IKE

- n Two modes:
  - n Aggressive mode: mutual authentication and session key establishment in three messages
    - n  $A \rightarrow B: g^a \text{ mod } p, A$ , crypto proposal
    - n  $B \rightarrow A: g^b \text{ mod } p$ , crypto choice, proof I'm  $B$
    - n  $A \rightarrow B$ : proof I'm  $A$
  - n Main: additional features such as hiding end-points identities and negotiating crypto DH algorithm
    - n  $A \rightarrow B$ : crypto suite I support
    - n  $B \rightarrow A$ : crypto suite I choose
    - n  $A \rightarrow B: g^a \text{ mod } p$
    - n  $B \rightarrow A: g^b \text{ mod } p$
    - n  $A \rightarrow B: g^{ab} \text{ mod } p \{A, \text{proof I'm } A\}$
    - n  $B \rightarrow A: g^{ab} \text{ mod } p \{B, \text{proof I'm } B\}$

Fall'04: CSG252 Classical Cryptography 19

---

---

---

---

---

---

---

---

## Phase 1 IKE

- n Key types:
  - n Pre-shared secret key
  - n Public encryption key: fields are separately encrypted using the public key
  - n Optimized public encryption key: used to encrypt a random symmetric key, and then data is encrypted using the symmetric key
  - n Public signature key: used only for signature purpose
- = 8 variants of IKE phase 1: 2 modes x 4 key types
- n Proof of Identity:
  - n Required in messages 2-3 aggressive mode and 5-6 main mode
  - n Proves the sender knows the key associated with the identity
  - n Depends on the key type
  - n Hash of identity key, DH values, nonces, crypto choices, cookies
  - n Alternative: MAC of previous messages

Fall'04: CSG252 Classical Cryptography 20

---

---

---

---

---

---

---

---

## Phase 1 IKE

- n Negotiating cryptographic parameters
  - n  $A$  specifies suites of acceptable algorithms:
    - {(3DES, MD5, RSA public key encryption, DH), (AES, SHA, pre-shared key, elliptic curve), ...}
  - n The standard specifies a MUST be implemented set of algorithms:
    - Encryption=DES, hash=MD5/SHA, authentication=pre-shared key/DH
  - n The lifetime of the SA can also be negotiated
- n Session keys:
  - n Key seed: SKEYID
    - Signature public keys:  $\text{SKEYID}_d = \text{prf}(\text{nonces}, g^a \text{ mod } p)$
    - Encryption public keys:  $\text{prf}(\text{hash}(\text{nonces}), \text{cookies})$
    - Pre-shared secret key:  $\text{prf}(\text{pre-shared secret key}, \text{nonces})$
  - n Secret to generate other keys:  $\text{SKEYID}_d = \text{prf}(\text{SKEYID}, (g^a, \text{cookies}, 0))$
  - n Integrity key:  $\text{SKEYID}_a = \text{prf}(\text{SKEYID}, (\text{SKEYID}_d, (g^a, \text{cookies}, 1)))$
  - n Encryption key:  $\text{SKEYID}_e = \text{prf}(\text{SKEYID}, (\text{SKEYID}_a, (g^a, \text{cookies}, 2)))$
- n Message IDs:
  - n Random 32-bits serves the purpose of a SN but in an inefficient manner because they have to be remembered

Fall'04: CSG252 Classical Cryptography 21

---

---

---

---

---

---

---

---

## IKE Phase 1: Public Signature Keys, Main Mode

- n Description:
  - n Both parties have public keys for signatures
  - n Hidden endpoint identity (except for ...?)
- n Protocol:
  - n  $A \rightarrow B: CP$
  - n  $B \rightarrow A: CPA$
  - n  $A \rightarrow B: g^a \text{ mod } p, \text{ nonce}_A$
  - n  $B \rightarrow A: g^b \text{ mod } p, \text{ nonce}_B$
  - n  $K = f(g^{ab} \text{ mod } p, \text{ nonce}_A, \text{ nonce}_B)$
  - n  $A \rightarrow B: K\{A, \text{proof I'm } A, [\text{certificate}] \}$
  - n  $B \rightarrow A: K\{B, \text{proof I'm } B, [\text{certificate}] \}$
- n Questions:
  - n What is the purpose of the nonces?
  - n Can we make to protocol shorter (5 messages)? At what expense?

Fall'04: CSG252

Classical Cryptography

22

---

---

---

---

---

---

---

---

---

---

## IKE Phase 1: Public Signature Keys, Aggressive Mode

- n Protocol:
  - n  $A \rightarrow B: CP, g^a \text{ mod } p, \text{ nonce}_A, A$
  - n  $B \rightarrow A: CPA, g^b \text{ mod } p, \text{ nonce}_B, B, \text{proof I'm } B, [\text{certificate}]$
  - n  $A \rightarrow B: \text{proof I'm } A, [\text{certificate}]$

Fall'04: CSG252

Classical Cryptography

23

---

---

---

---

---

---

---

---

---

---

## IKE Phase 1: Public Encryption Keys, Main Mode, Original

- n Protocol:
  - n  $A \rightarrow B: CP$
  - n  $B \rightarrow A: CPA$
  - n  $A \rightarrow B: g^a \text{ mod } p, \{\text{nonce}_A\}_B, \{A\}_B$
  - n  $B \rightarrow A: g^b \text{ mod } p, \{\text{nonce}_B\}_A, \{B\}_A$
  - n  $K = f(g^{ab} \text{ mod } p, \text{ nonce}_A, \text{ nonce}_B)$
  - n  $A \rightarrow B: K\{\text{proof I'm } A\}$
  - n  $B \rightarrow A: K\{\text{proof I'm } B\}$

Fall'04: CSG252

Classical Cryptography

24

---

---

---

---

---

---

---

---

---

---

IKE Phase 1:  
Public Encryption Keys, Aggressive Mode, Original

Protocol:

- n  $A \rightarrow B: CP, g^a \text{ mod } p, \{\text{nonce}_{A \rightarrow B}\} \{A\}_B$
- n  $B \rightarrow A: CPA, g^b \text{ mod } p, \{\text{nonce}_{B \rightarrow A}\} \{B\}_{A, \text{proof I'm } B}$
- n  $A \rightarrow B: \text{proof I'm } A$

Fall'04: CSG252      Classical Cryptography      25

---

---

---

---

---

---

---

---

IKE Phase 1:  
Public Encryption Keys, Main Mode, Revised

Protocol:

- n  $A \rightarrow B: CP$
- n  $B \rightarrow A: CPA$
- $K_A = \text{hash}(\text{nonce}_{A, \text{cookie}_A})$
- n  $A \rightarrow B: \{\text{nonce}_{A \rightarrow B}\} K_A\{g^a \text{ mod } p\}, K_A\{A\}, [K_A\{A \& \text{cert}\}]$
- $K_B = \text{hash}(\text{nonce}_{B, \text{cookie}_B})$
- n  $B \rightarrow A: \{\text{nonce}_{B \rightarrow A}\} K_B\{g^b \text{ mod } p\}, K_B\{B\}$
- $K = f(g^{ab} \text{ mod } p, \text{nonce}_{A, \text{nonce}_{B, \text{cookie}_A, \text{cookie}_B})$
- n  $A \rightarrow B: K\{\text{proof I'm } A\}$
- n  $B \rightarrow A: K\{\text{proof I'm } B\}$

Fall'04: CSG252      Classical Cryptography      26

---

---

---

---

---

---

---

---

IKE Phase 1:  
Public Encryption Keys, Aggressive Mode, Revised

Protocol:

- $K_A = \text{hash}(\text{nonce}_{A, \text{cookie}_A})$
- n  $A \rightarrow B: CP, \{\text{nonce}_{A \rightarrow B}\} K_A\{g^a \text{ mod } p\}, K_A\{A\}, [K_A\{A \& \text{cert}\}]$
- $K_B = \text{hash}(\text{nonce}_{B, \text{cookie}_B})$
- n  $B \rightarrow A: CPA, \{\text{nonce}_{B \rightarrow A}\} K_B\{g^b \text{ mod } p\}, K_B\{B\}, \text{proof I'm } B$
- $K = f(g^{ab} \text{ mod } p, \text{nonce}_{A, \text{nonce}_{B, \text{cookie}_A, \text{cookie}_B})$
- n  $A \rightarrow B: K\{\text{proof I'm } A\}$

Fall'04: CSG252      Classical Cryptography      27

---

---

---

---

---

---

---

---

### IKE Phase 1: Shared Secret Keys, Main Mode

- Assumption  $A$  and  $B$  share a secret  $J$
- Protocol:
  - $A \rightarrow B$ :  $CP$
  - $B \rightarrow A$ :  $CPA$
  - $A \rightarrow B$ :  $g^a \text{ mod } p$ ,  $\text{nonce}_A$
  - $B \rightarrow A$ :  $g^b \text{ mod } p$ ,  $\text{nonce}_B$
  - $K = f(J, g^{ab} \text{ mod } p, \text{nonce}_A, \text{nonce}_B, \text{cookie}_A, \text{cookie}_B)$
  - $A \rightarrow B$ :  $K\{\text{proof I'm } A\}$
  - $B \rightarrow A$ :  $K\{\text{proof I'm } B\}$

Fall'04: CS6252      Classical Cryptography      28

---

---

---

---

---

---

---

---

---

---

### IKE Phase 1: Shared Secret Keys, Aggressive Mode

- Protocol:
  - $A \rightarrow B$ :  $CP, g^a \text{ mod } p$ ,  $\text{nonce}_A, A$
  - $B \rightarrow A$ :  $CPA, g^b \text{ mod } p$ ,  $\text{nonce}_B, B$ ,  $\text{proof I'm } B$
  - $A \rightarrow B$ :  $\text{proof I'm } A$

Fall'04: CS6252      Classical Cryptography      29

---

---

---

---

---

---

---

---

---

---

### IKE: Phase 2

- Also known as "Quick Mode": 3- messages protocol
  - $A \rightarrow B$ :  $X, Y, CP, \text{traffic}, SPI_A, \text{nonce}_X [g^x \text{ mod } p]_{\text{optional}}$
  - $B \rightarrow A$ :  $X, Y, CPA, \text{traffic}, SPI_B, \text{nonce}_Y [g^y \text{ mod } p]_{\text{optional}}$
  - $A \rightarrow B$ :  $X, Y, \text{ack}$
- All messages are encrypted using SKEYID\_e, and integrity protected using SKEYID\_a (except  $X, Y$ )
- Parameters:
  - $X$ : pair of cookies generated during phase 1
  - $Y$ : 32-bit number unique to this phase 2 session chosen by the initiator
  - CP: Crypto Proposal, CPA: Crypto Proposal Accepted
  - DH is optional and could be used to provide PFS
  - Nonces and cookies get shuffled into SKEYID to produce the SA encryption and integrity keys

Fall'04: CS6252      Classical Cryptography      30

---

---

---

---

---

---

---

---

---

---



---

---

---

---

---

---

---