Cryptosystem Concepts

SE3A04 – Tutorial

Jason Jaskolka

Department of Computing and Software
Faculty of Engineering
McMaster University
Hamilton, Ontario, Canada
jaskolj@mcmaster.ca

October 2, 2012

Outline

What is Cryptography?

Definition (Cryptography)

Cryptography derives from two Greek words meaning *secret writing* and is the art and science of concealing meaning.

- Cryptography provides a cornerstone for secure communication
- Allows for communication which can only be read by the intended receiver
- Cryptography is deeply rooted in mathematics

Cryptosystems

Definition (Cryptosystem)

A **cryptosystem** is a 5-tuple $(\mathcal{E}, \mathcal{D}, \mathcal{M}, \mathcal{K}, \mathcal{C})$, where

- $\mathcal{M}$ is the set of *plaintexts*
- $\mathcal{K}$ is the set of *keys*
- $\mathcal{C}$ is the set of *ciphertexts*
- $\mathcal{E} : \mathcal{M} \times \mathcal{K} \to \mathcal{C}$ is the set of *enciphering functions*
- $\mathcal{D} : \mathcal{C} \times \mathcal{K} \to \mathcal{M}$ is the set of *deciphering functions*

Cryptosystems

Example (Cæsar Cipher)

The **Cæsar cipher** is a widely known cipher in which letters are shifted. For example, if the key is 3, then the letter A becomes D, the letter B becomes E, and so on. Informally, this cipher is a cryptosystem with:

- $\mathcal{M} = \{\text{all sequences of Roman letters}\}$
- $\mathcal{K} = \{i \in \mathbb{Z} \mid 0 \leq i \leq 25\}$
- $\mathcal{C} = \mathcal{M}$
- $\mathcal{E} = \{E_k \mid k \in \mathcal{K} \wedge \forall(m \mid m \in \mathcal{M} : E_k(m) = (m + k) \mod 26)\}$
- $\mathcal{D} = \{D_k \mid k \in \mathcal{K} \wedge \forall(c \mid c \in \mathcal{C} : D_k(c) = (26 + c - k) \mod 26)\}$

Symmetric-Key Cryptosystems

Definition (Symmetric-Key Cryptosystems)

A **symmetric-key cryptosystem** is a class of cryptosystems that use the same cryptographic keys for both encryption of plaintext and decryption of ciphertext.

- The keys represent a shared secret between two or more parties that can be used to maintain a private communication link

Symmetric-Key Cryptosystems

- Symmetric-key cryptosystems can use either **stream ciphers** or **block ciphers**

Definition (Stream Cipher)

Stream ciphers encrypt the bits of a message one at a time.

Definition (Block Cipher)

Block ciphers encrypt a number of bits as a single unit, padding the plaintext so that it is a multiple of the block size (commonly 64 bits).

Symmetric-Key Cryptosystems

- There exists a wide variety of popular and well-respected symmetric-key cryptosystems
- **Examples**:
  - Data Encryption Standard (DES)
  - Triple DES (3DES)
  - Advanced Encryption Standard (AES)
  - International Data Encryption Algorithm (IDEA)
  - Blowfish
  - etc.
- Each have their own strengths and weaknesses
- Many implementations and libraries exist

Taking Care of Our Keys

- The goal of cryptography is to keep enciphered information secret
- Standard cryptographic practice assumes that an **adversary** knows the algorithms used to encipher the plaintext, but not the specific cryptographic key which is used
  - In other words, the adversary knows $\mathcal{E}$ and $\mathcal{D}$
- This means that we need to take care of our cryptographic keys!

Why Do We Need KDCs?

- Assume that network security is based on secret keys
- Assume that we have a network of n nodes and that each node would need to authenticate each other node, i.e., the network topology is K_n (complete graph of n vertices)
- Each node would need to know $n-1$ keys; one for each other node on the network
- If a new node is added to the network, then n new keys would need to be generated and somehow securely distributed to all other nodes on the network
- This is clearly untenable for all but very small networks

Why Do We Need KDCs?

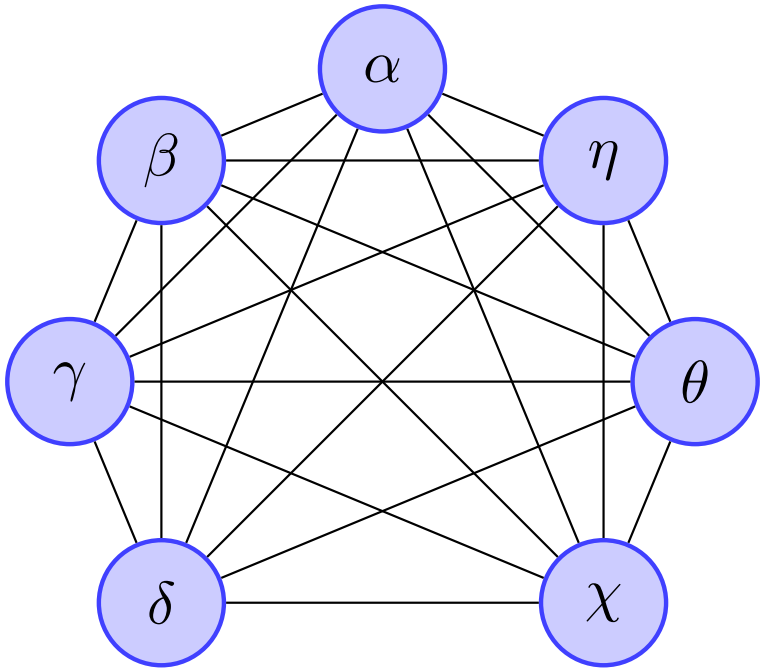


Figure : A network of 7 nodes (K_7) without the use of a KDC

Why Do We Need KDCs?

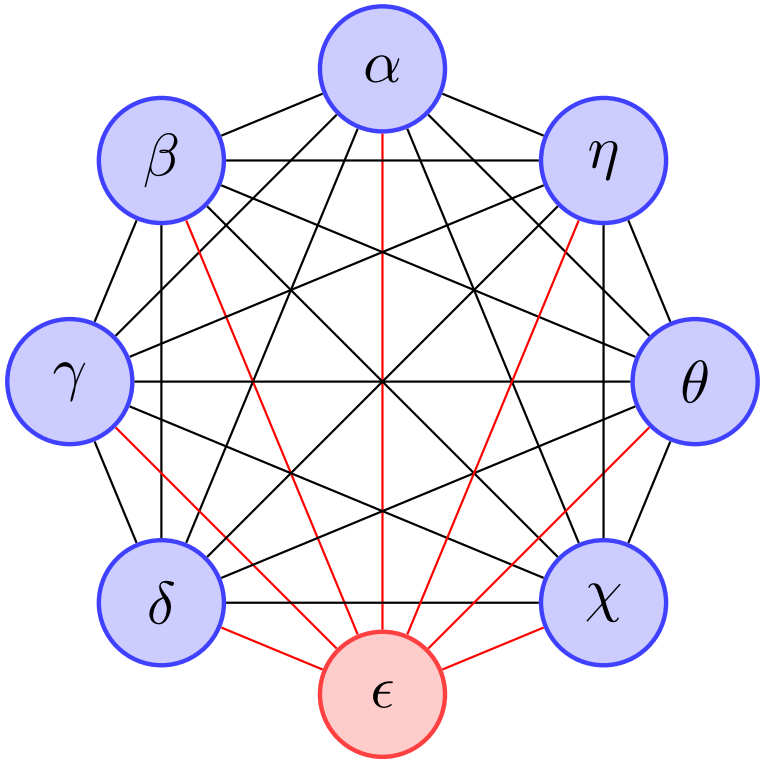


Figure : The addition of a node when a KDC is not used; K_8

Key Distribution Centres

Definition (Key Distribution Centre (KDC))

A **Key Distribution Centre (KDC)** is a trusted node which knows the keys of all other nodes in the network.

- KDCs help to make things more manageable
- If a new node is added to the network, only, that node and the KDC need to be configured with a key for that node
- The use of a KDC creates a network topology of S_n (star graph of n vertices)

Key Distribution Centres

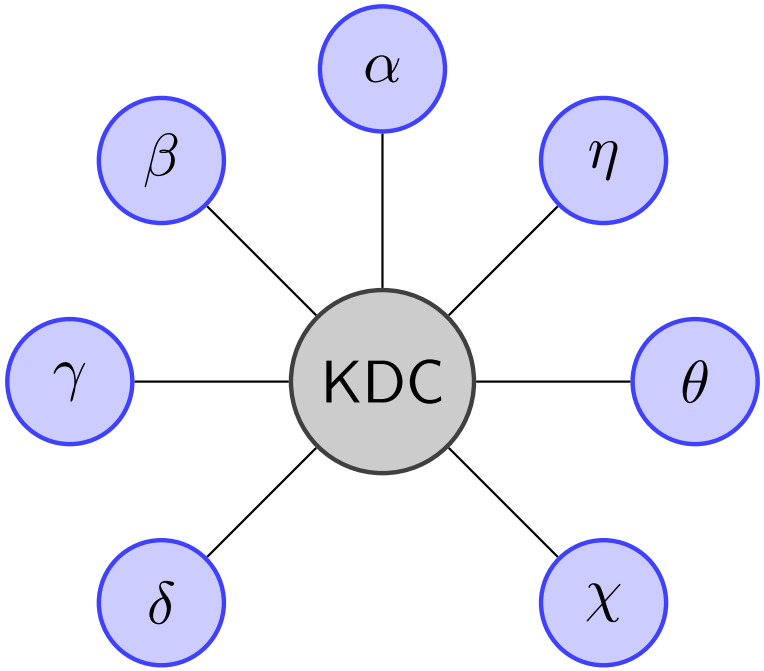


Figure : A network of 7 nodes (S_7) with the use of a KDC

Key Distribution Centres

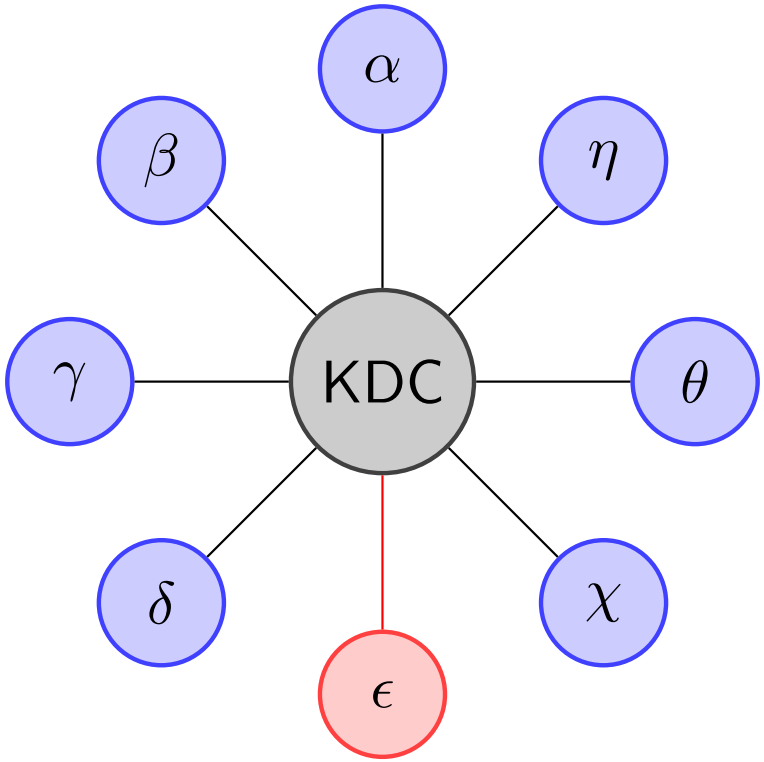


Figure : The addition of a node when using a KDC; S_8

Some Disadvantages of KDCs

- The KDC has enough information to impersonate anyone to anyone
  - If the KDC is compromised, then all of the network resources are vulnerable
- The KDC is a single point of failure
  - If the KDC goes down, then the network becomes unusable past the time of failure, i.e., nobody can start using something on the network, but previously distributed keys can continue to be used
- The KDC can sometimes become a performance bottleneck since each node needs to communicate with it frequently
  - This problem can be alleviated by utilizing multiple KDCs

Node to Node Communication with a KDC

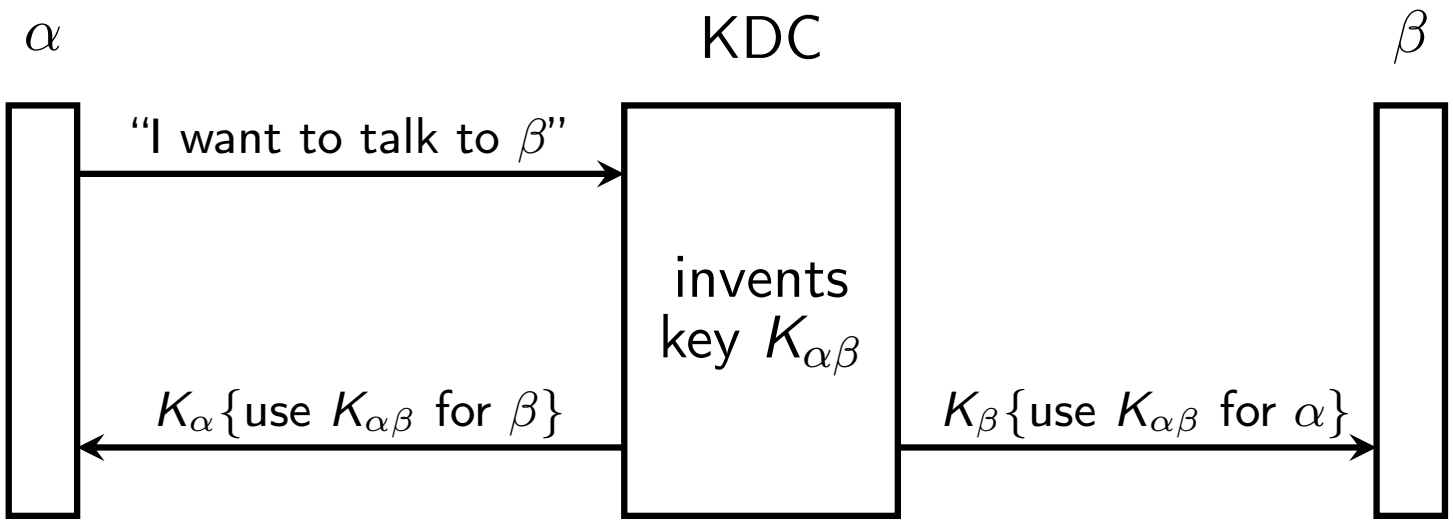


Figure : KDC operation (in principle)

Node to Node Communication with a KDC and Tickets

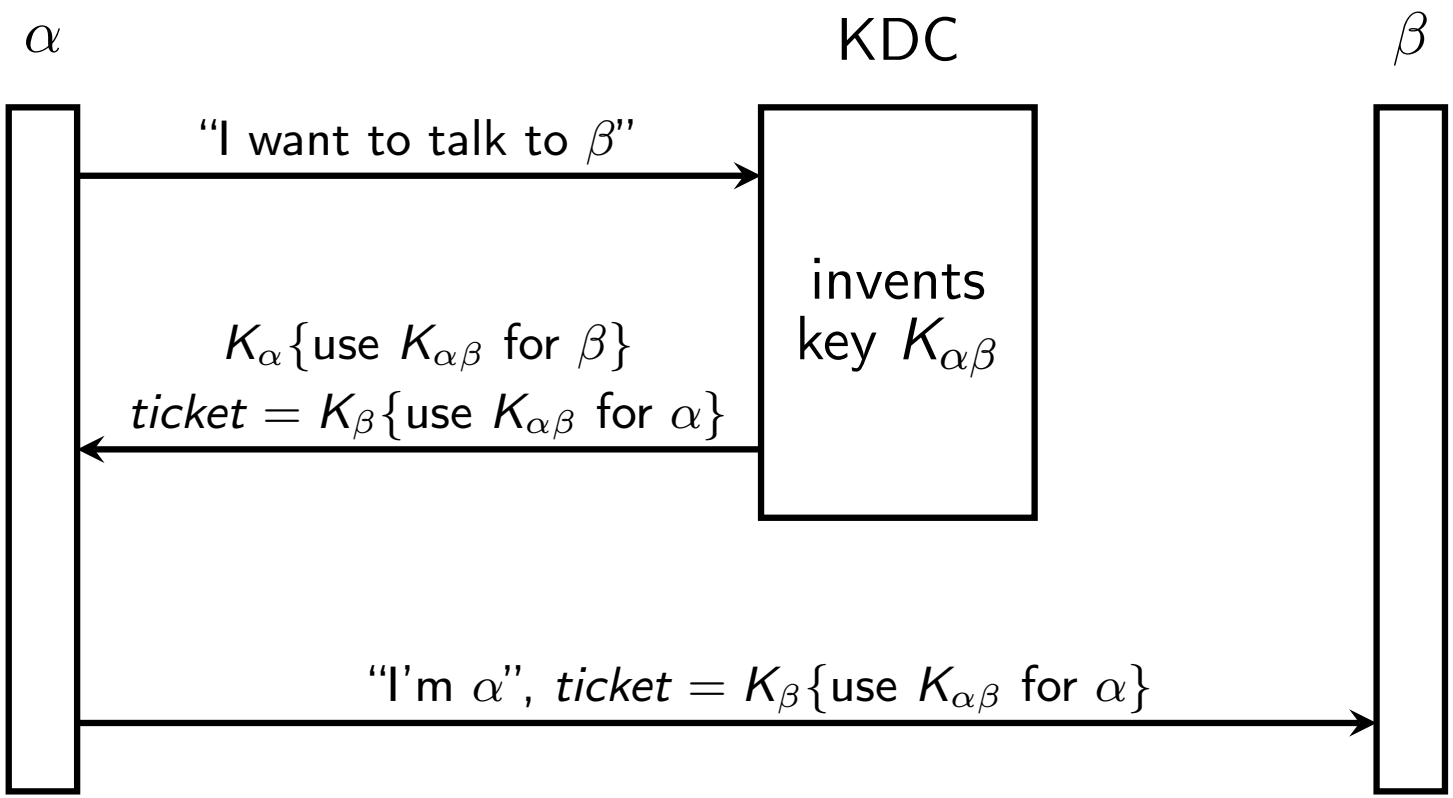


Figure : Alternative KDC operation with tickets (in principle)

A Word About Tickets

- The message that the KDC sends to α to forward to β is often called a **ticket**
- Besides containing the key to be used for communication between α and β, the ticket generally contains other information such as an expiration time and the identity of α
- Tickets are constructed by trusted intermediaries to enable two parties to authenticate each other

Why Use Mediated Authentication?

- In an exchange with a KDC, the KDC does not know really who it is communicating with
- It is possible that an impostor can send a message to the KDC claiming to be α
- However, this would require the impostor to know the key of α in order to decrypt the encrypted $K_{\alpha\beta}$

Why Use Mediated Authentication?

- It is only after α and β perform some kind of **mutual authentication** after the key exchange that they know for certain who they are communicating with and that each knows the key $K_{\alpha\beta}$
- Mutual authentication is known to have issues in terms of scaling
- Instead, we want to use **mediated authentication**

Mediated Authentication

Definition (Mediated Authentication)

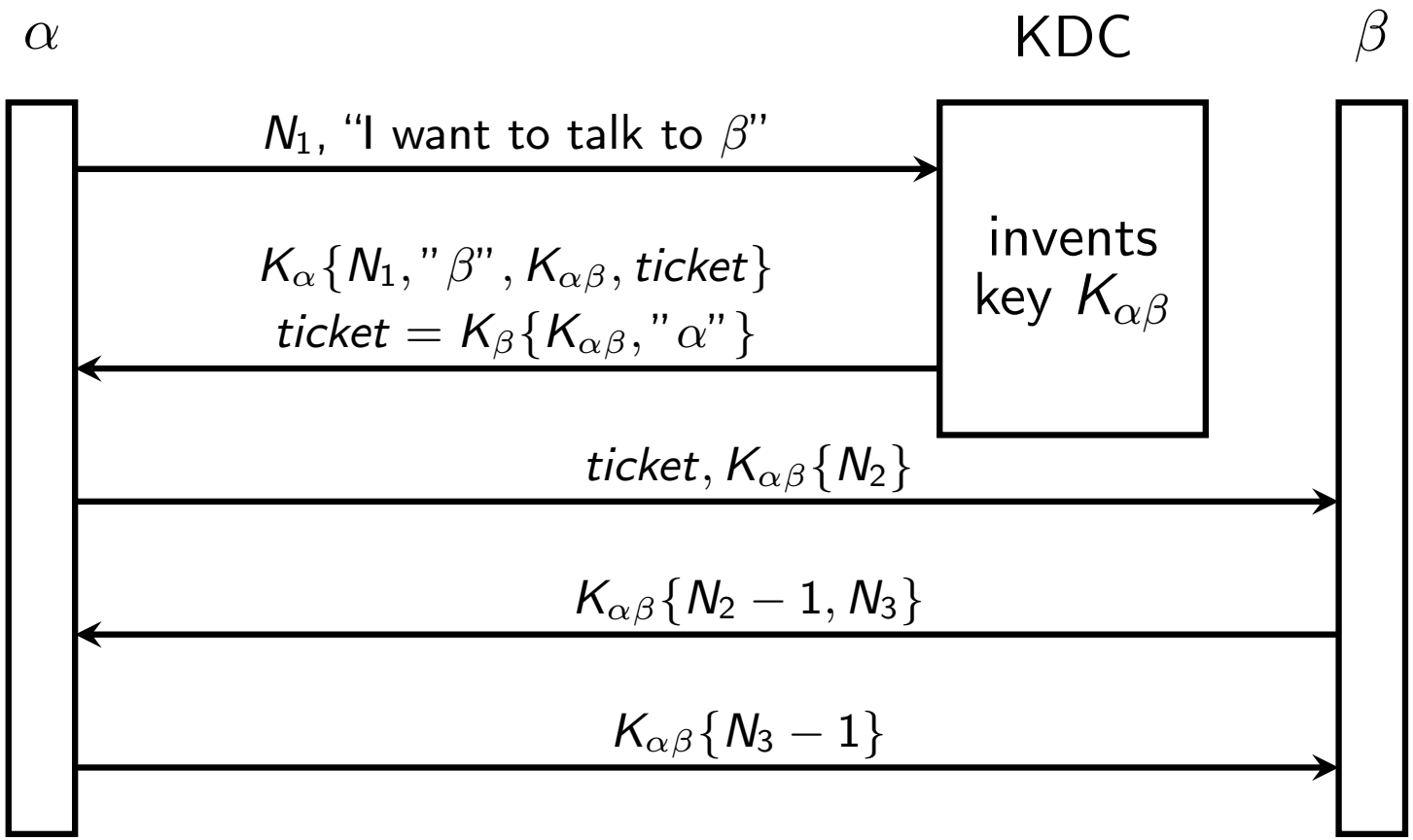
Mediated authentication involves the use of a third party (usually a KDC) to arbitrate the authentication process after a key exchange.

- With mediated authentication, the KDC generates a **session key** ($K_{\alpha\beta}$) and distributes it to the communicating parties
- The communicating parties then prove to each other that they know the session key

Needham-Schroeder Protocol

- The **Needham-Schroeder Protocol** is important because it is a classic KDC-arbitrated authentication protocol which has served as the basis for many other authentication protocols
- A **nonce** (dented by N_i) refers to a number that is used only once and may be a:
  - Sequence number (provided that state is not lost during crashes)
  - Large random number
  - Timestamps (provided clocks never go backwards)
  - etc.

Needham-Schroeder Protocol



Needham-Schroeder Protocol

- N_1 assures α that it is really communicating with the KDC
- N_2 is α's challenge to β, ensuring that β knows the session key $K_{\alpha\beta}$
- N_3 is β's challenge to α, ensuring that α knows the session key $K_{\alpha\beta}$
- After all exchanges, α and β are certain that they are communicating with one another that they each know the session key

Some Other Mediated Authentication Protocols

- There are a number of other mediated authentication protocols
- **Examples**:
  - Expanded Needham-Schroeder Protocol
  - Otway-Rees
  - Kerberos
- Most differ in the use of nonces to perform authentication
- Each has their own strengths and weaknesses

Some Comments About Implementation and Libraries

- Use of existing libraries is permitted and encouraged
- **Examples**:
  - Java Cryptography Extension
  - Bouncy Castle
  - Apache Shiro
  - etc.
- **Note**: Please document your use of existing libraries!

References

Matt Bishop.
Computer Security: Art and Science
Addison-Wesley, 2002.

Charlie Kaufman, Radia Perlman and Mike Speciner
Network Security: Private Communication in a Public World, 2nd edition
Prentice Hall, 2002.

Questions

- Questions?