

# Systeme II

## 7. Sicherheit

Christian Schindelhauer

Technische Fakultät

Rechnernetze und Telematik

Albert-Ludwigs-Universität Freiburg

(Version 17.07.2017)

- Folien und Inhalte aus
  - Computer Networking: A Top Down Approach  
5th edition.  
Jim Kurose, Keith Ross  
Addison-Wesley, April  
2009.
  - Copyright liegt bei den  
Autoren Kurose und Ross

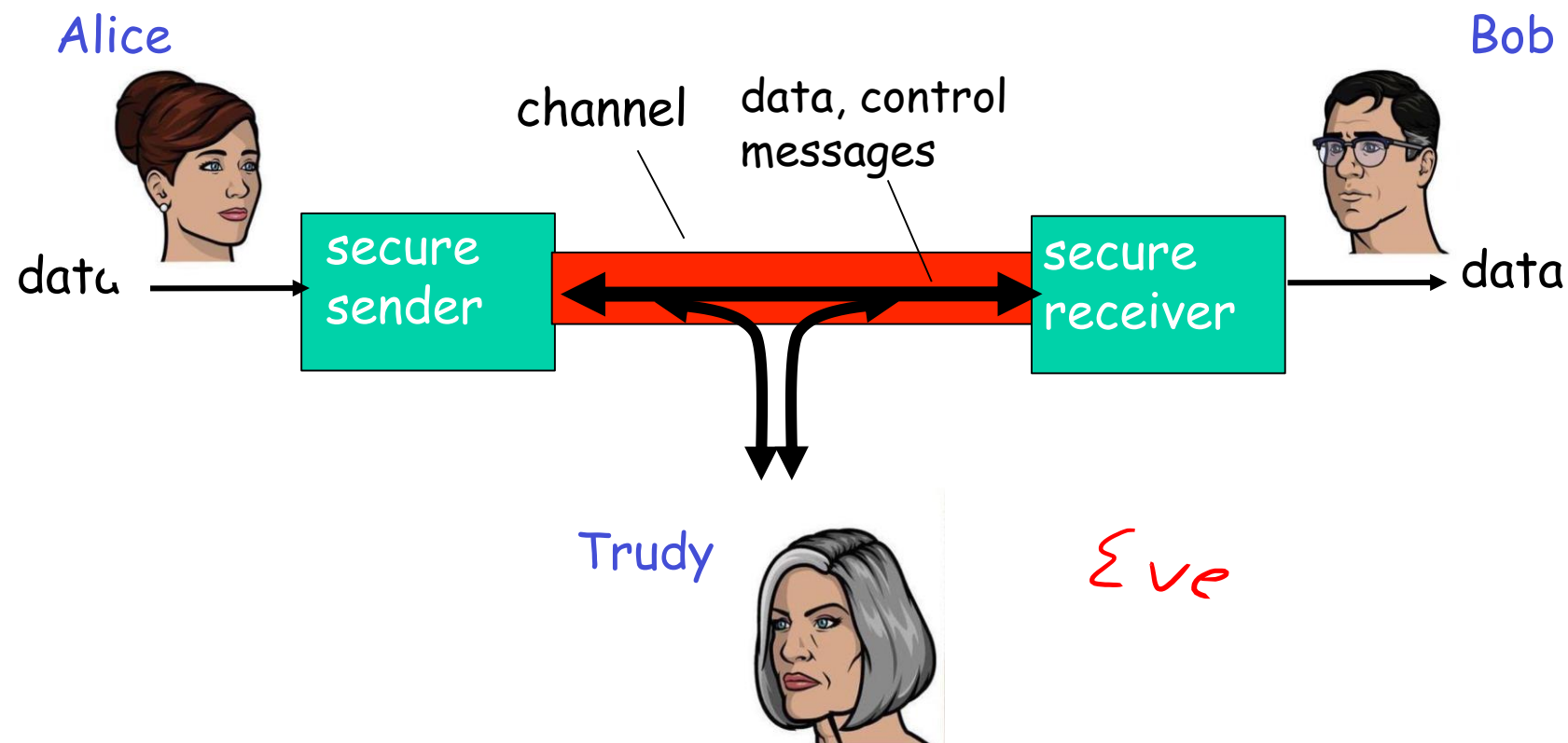
- Grundlagen von Netzwerksicherheit
  - Kryptographie und deren vielfältige Einsatzmöglichkeiten
  - Authentifizierung
  - Message Integrity
- Sicherheit in der Praxis
  - Firewalls und Intrusion Detection
  - Sicherheit in Anwendungs-, Transport-, Vermittlungs- und Sicherungsschicht

- Vertraulichkeit (Confidentiality)
  - Nur der Sender, gewünschter Empfänger sollte den Nachrichteninhalt „verstehen“
- Authentifizierung
  - Sender und Empfänger möchten sich ihrer Identität versichern
- Integrität (message integrity)
  - Sender und Empfänger wollen, dass eine Nachricht nicht unbemerkt verändert werden
    - bei der Übertragung oder später
- Zugriff und Verfügbarkeit
  - von Diensten

*DoS = Denial of Service*

# Freunde und Feinde: Alice, Bob und Trudy

- Standardnamen im Sicherheitsbereich
- Alice und Bob möchten „sicher“ kommunizieren
- Trude (In-Trude-r) möchte mithören, löschen, hinzufügen, verändern



# Wer steckt hinter Alice und Bob

---

- Echte Menschen
- Web-Browser
- Online-Banking-Clients und Servers
- DNS-Servers
- Routers, die Routing-Tabellen austauschen
- etc.

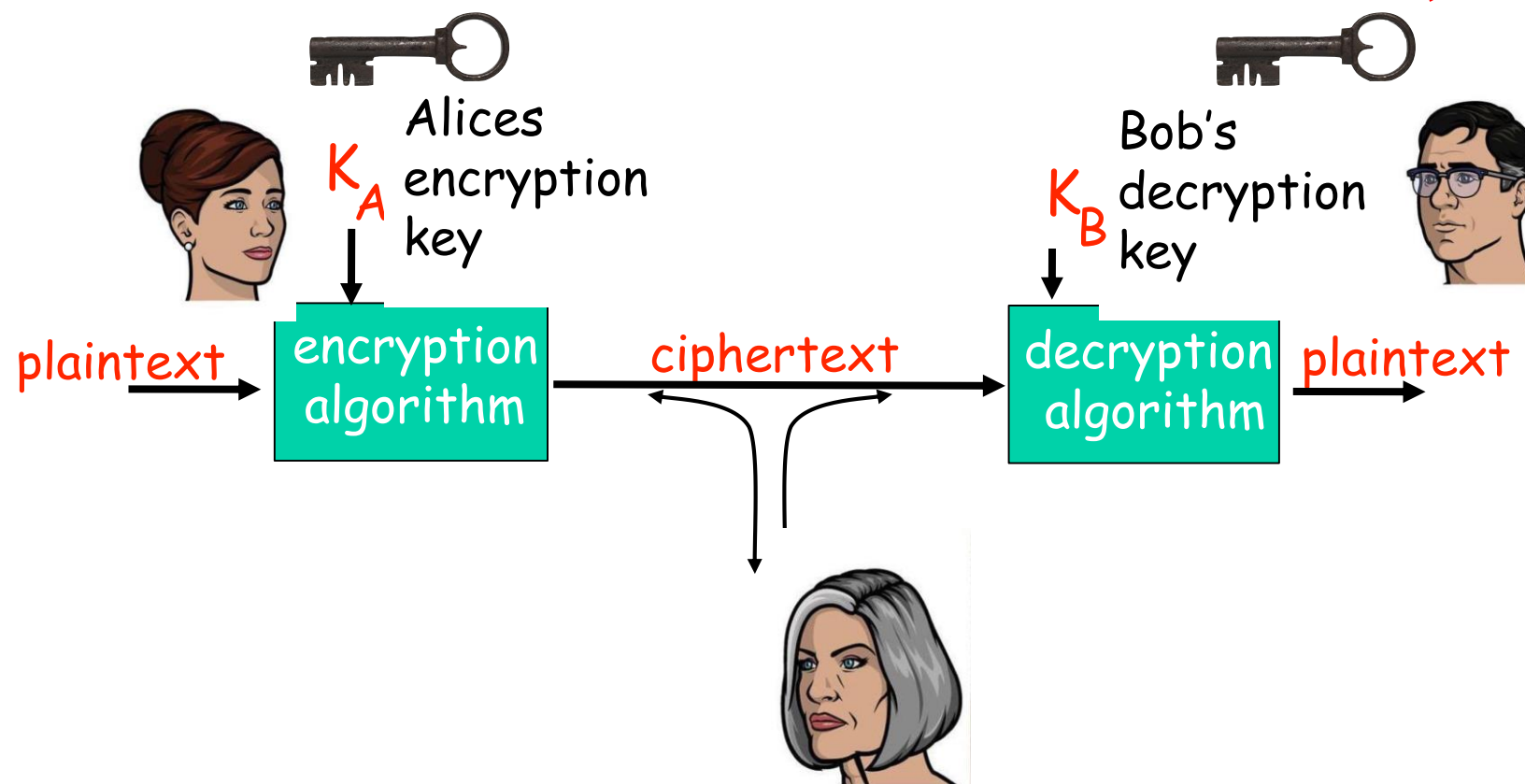
# Was kann ein böser Mensch so tun?

- Abhören (eavesdrop)
  - Nachrichten abfangen und lesen
- Einfügen von Nachrichten
  - Nachrichten werden in die bestehende Verbindung eingefügt
- Sich als jemand anders ausgeben (impersonation)
  - Quell-Adresse kann in einem Paket gefälscht werden
- Hijacking
  - Übernahme einer bestehenden Verbindung durch Ersetzen des Empfängers oder Senders
- Denial of Service
  - Dienst abschalten
    - durch Überlast oder direkten Angriff

$K_A$  injektiv

- $m$ : Originalnachricht (message)
- $K_A(m)$ : mit Schlüssel  $K_A$  verschlüsselte Nachricht
- $m = K_B(K_A(m))$

$K_A = K_B$   
 $K_A \neq K_B$   
 AES  
 symm.  
 unsymm.  
 RSA





*a b c d e f*  
*↓ ↓ ↓ ↓ ↓ ↓*  
*x y z a b c d e f*

## ■ Monoalphabetischer Schlüssel

- ersetze jeden Buchstaben durch einen anderen

## ■ Beispiel: Edgar Allen Poe „The Gold Bug“

- 53\$05))6\*;4826)4)4;806\*;488¶60))85;1-(;:\*8-83(88)5\*
- ;46(;88\*96\*?;8)\*(;485);5\*2:\*(;4956\*2(5\*-4)8¶8\*;40692
- 85);)68)4;1(9;48081;8:81;4885;4)485528806\*81(9;48;
- (88;4(?34;48)4;161;:188;?;

*a*  
*b*  
*c*  
*d*  
*e*  
*:*  
*:*

*8*

## ■ Jedes Symbol steht für einen Buchstaben:

- 8 = e
- ; = h
- ...

- ersetze jeden Buchstaben durch einen anderen

Permutation  
 $26!$

E.g.: Plaintext: bob. i love you. alice  
ciphertext: nkn. s gktc wky. mgsbc

- n monoalphabetische Schlüssel,  $M_1, M_2, \dots, M_n$
- Zyklus-Muster
  - e.g.,  $n=4$ ,  $M_1, M_3, M_4, M_3, M_2$ ;  $M_1, M_3, M_4, M_3, M_2$ ;
- Für jeden neuen Buchstaben aus den monoalphabetischen Schlüsseln einer ausgewählt
  - „aus“: a from  $M_1$ , u from  $M_3$ , s from  $M_4$
  - Schlüssel: n Schlüsselverfahren und der Zyklus

- Cipher-text only Attack
  - nur mit verschlüsselten Text
  - Zwei Ansätze:
    - Durchsuche alle Schlüssel und teste ob sie einen vernünftigen Text produzieren
    - Statistische Analyse des Schlüssels
- Known-Plaintext-Attack
  - mit der Originalnachricht und dem verschlüsselten Text
- Chosen Plaintext Attack
  - Trudy wählt den Text und lässt Alice ihn verschlüsseln
  - Trudy erhält den verschlüsselten Text