PROJEKT WISE 20/21

Prof. Dr. Dirk Westhoff
Medien und Informationswesen

WS 2020/21

Hochschule Offenburg
offenburg.university

Vorbemerkungen

- Projekt:

  Kunde wünscht ‚Verschlüsselung‘
  - ‚Was genau?

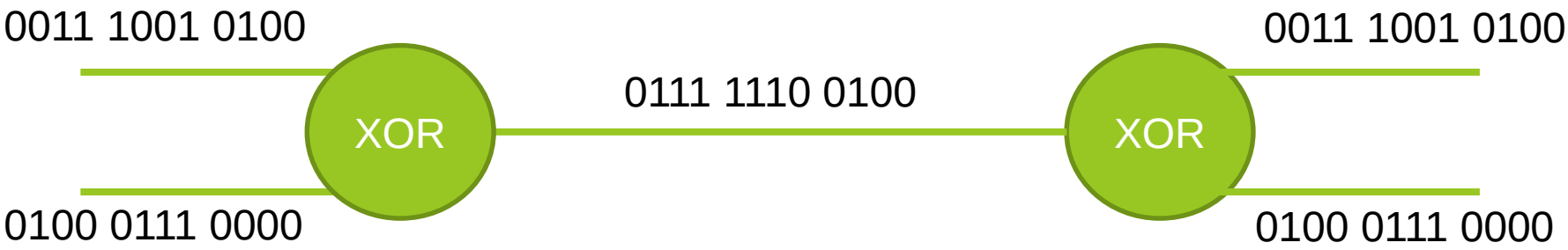
    Wie ein OTP, oder eine Stromchiffre mit fester Blockgröße‘

- Vier Komponenten
  - Client-SW: Einlesen des Textes, ‚Setzen‘ des Schlüssels, Codieren, Verschlüsseln, (Versenden..)…
  - Server-SW: (Empfangen…), Decodieren1, ‚Setzen‘ des Schlüssels, Entschlüsseln, Decodieren2
  - Testumgebung: Zum Testen von Client-SW & Server-SW „als würde“ via Internet/Socket kommuniziert werden
  - Brute-Force-Komponente: Komponente die eigene Entschlüsselungs-Methode verwendet um ein Chiffrat *ohne* Schlüssel zu entschlüsseln

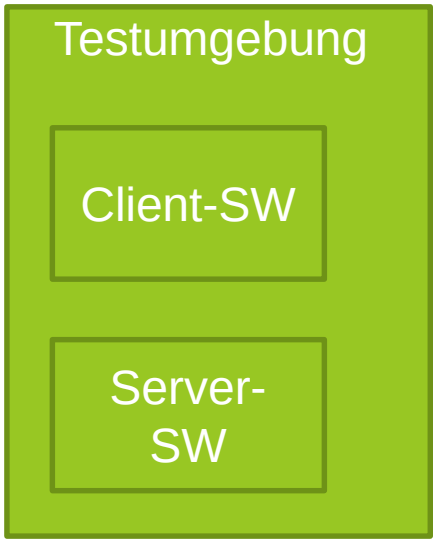
Vorbemerkungen

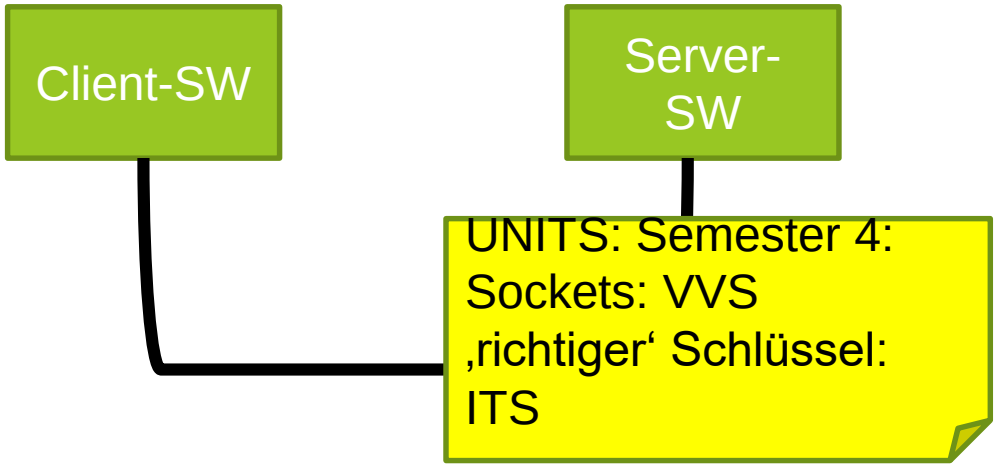
Vorgabe:

- blockweise Bearbeitung (64, 128, …., 1024)
- Ver/Entschlüsselung mittels XOR
- 27 Zeichen (a,b,…,z) und Leerzeichen
- Schlüssel: symmetrisch & gleich der Blocklänge
    - hier: deterministisch !!!



Architektur





- Client-SW:
  1. Nimmt String (oder Menge von chars) entgegen
  2. sendet verschlüsselte Nachricht (String) über Methode **schreibeNachricht(…)** an ServerSW
- Server-SW:
  1. Horcht auf verschlüsselte Nachricht über Methode **leseNachricht()**
  2. Gibt entschlüsselte Nachricht aus

‚Objectives‘

Womit ist Kunde zufrieden?

- Ver- und Entschlüsselung sollen auch funktionieren wenn die Ver- Entschlüsselungs-SW der Gegenstelle von anderen Hersteller implementiert wurde.

- daher:
  Zwei Teams -> **eine gemeinsame** Spezifikation ->
  zwei getrennte Implementierungen !!! (pro Team eine)

Implementierung **unbedingt** aufheben, da Sie im Verlauf des Studiums noch benötigt wird!!!

Teams & Sprecher

- Team1:
  Gruppen G1 – G8

- Team2:
  Gruppen G9 – G16

Jedes Team benennt einen Sprecher!

‚Objectives‘

Womit ist Kunde zufrieden?

- Brute-Force-Komponente: Entschlüsseln *ohne* Schlüsselkenntnis
  1. Schlüssel: unabhängig von der Blockgröße sollen *immer* nur die ersten 16-Bit ‚zufällig‘ vergeben werden
  2. die restlichen Bits werden mit ‚0‘ gefüllt
  3. Messe die Zeit bis das Chiffrat der anderen Gruppe entschlüsselt ist

     Option1: Klartextfolge endet immer mit ‚01111110‘
     Bsp.: ‚1100010111111001……01111110‘

     Option2: Klartextfolge endet immer mit ‚_ende_‘
     Bsp.: ‚Dies ist der Klartext……_ende_‘

     Option3: Klartextfolge endet immer mit ‚_ende_‘
     Bsp.: ‚Dies ist der Klartext……‘\0‘‘

Meilensteine…

1. Vorstellung der gemeinsamen verbindlichen Spezifikation (IETF RFC ☺)
   - beinhaltet Dokument zur Spezifikation & Foliensatz
   - gemeinsam Mo 23.11 (in der Vorlesung: zwei Sprecher)

2. Vorstellung der Lösungen (Gruppe 1 und Gruppe 2)
   - Team 1: Mo 07.12 [ca. 45 min]
   - Team 2: Mo 14.12 [ca. 45 min]
     - Entspricht Lösung der gemeinsamen Spezifikation?
     - Kompatibilitäts-Test der Lösungen
     - Performanz-Test der Brute-Force Komponente

Beispiel 1

Wie findet man einen Algorithmus?

1. Konkrete Beispiele durchspielen
2. Grundidee festhalten
3. Formalen Algorithmus aufschreiben
4. Algorithmus mit weiteren Beispielen testen

Algo finden ist hier nicht so schwierig, aber sinnvolle Datentypen, Methoden-Schnittstellen, etc.