Übung zur Vorlesung „Sicherheit“
Übung 1

Thomas Agrikola
Thomas.Agrikola@kit.edu

04.05.2017

Literatur zur Vorlesung

Jonathan Katz, Yehuda Lindell. *Introduction to Modern Cryptography*. ISBN 1-584-88551-3.
`https://www.cs.umd.edu/~jkatz/imc.html`

Ross Andersson. *Security Engineering*. ISBN 0-470-06852-3.
`https://www.cl.cam.ac.uk/~rja14/book.html`

Skript auf der Vorlesungs-Webseite.

Ein paar Worte zu Übung und Übungsblättern

- Übungsblätter freiwillig
  - Kein Klausurbonus o.ä.
  - Aber: Klausur wird sich an den Übungen orientieren!
- Diskussionen/Fragen willkommen
- Übung ca. alle 2 Wochen, Donnerstags

- Feedback-Kasten auf Website
  - für anonymes Feedback
- Mailingliste
  - Kommunikation, Fragen, Diskussionen, ...

Übung: Struktur

- Übungsteil
  - Besprechung des aktuellen Übungsblatts
  - Nicht unbedingt immer alle Aufgaben
- „Tutoriumteil“
  - Vorbereitung aufs nächste Übungsblatt
  - Besprechen von ähnlichen Aufgaben
- Manchmal auch VL-Wiederholung

Experiment: Socrative

`https://b.socrative.com/login/student/`

Room: `SICHERHEIT`

- App um Quiz durchzuführen
- Zugang durch Browser oder App
- Als Quizteilnehmer kein Account notwendig.

Fragen?

Restliche Agenda für heute

- Aufgaben besprechen
  - Vigenère
  - Blockchiffren
  - Zufallsgenerator (LFSRs)
- Beweisbare Sicherheit
- One-Time-Pad

Übungsblatt 1

Aufgabe 1. Gegeben ist der folgende Chiffretext. Ermitteln Sie den dazugehörigen Klartext. (Hinweis: Als Verschlüsselungsmechanismus wurde das Vigenère-Verfahren benutzt.)

```
MZMTELFMPLWUFEWXTZPCJDQPBVUKSIEEQDIMIIDRLQNTPWFBZNYNTTQLMAFAGQADDYXOPLIDIWYXFIN
ISZBSCZUOPLIDMNGTTMCCEDTTCVMBEYGWACCPUMOMRWVZUPQLRCSRBSCTXITLXQFEGSDCDCSRICCGAO
YGDMJWCAAZOYWMSPWOMATQOUAXCXTWOFEPVZQYOPOCTQVOCROQPQOMATQOUELQXTMQGVEBEMTGJWGWT
IYYGOWFLXANEFIMBEYGWJFRMFQDAPQICRLMBEFIDMHCVQWEFIDAHFSIMCCEIICCSRQEGRQQRFXQMYDM
RBJDSGZNFEDTPQFMJMYKQELQKAIOCHUVEMFDMLIMZOEFIHQRCRQZPAMBPPPATMYHSTVSYPXJCMGWBSU
EUBPQWGJXGXFMOYRQENGTTMCRSFPPHSGZYYPANEFIEWNGIFGZDXTMLPXEESCRNIMZESMDFSIMORLMBE
FAMQECWOQAFIDELQIEAPLXUIWJCVCDREZWEFIDZPAVQIEGSZWQRLQDTEIZMCCGUXSCVFPHYMFMDALMT
WCRSMOZENJLEIFWMPIMSSGWOQAFIDMYASPMORAUKPUMFPVCCEWQBMRNPPIZBWCRSBSZENJLEIECNAIQ
LPBMZLPAVKXEGRSIDYQBTPULUKSRYDVPBSGBEMFQBSCTAMXRLQDTQMAVZDWUVMWEXNCCHFMYLCEWYCR
OZJNXQLLAGAZOGRSBZRLQSPWAAZOCQUTJRLQNTPWFVLKIANECRZGDMREETDINIMZESMYCZQZPVTXITL
IPBSCQQBSMHTMFQIPAESHUMDMJNIMZESMDLSFMDPIHMLJXTIEFITIOSWQLEFIYMEFSPTLRIDXFZPUAS
CHNGVYWUAVGEZLDSKSMDRXTIEFITIOZIQVFQMZOEFIYMEFSPIDCEDTJYWQQRFXQMYDSGZEWWUF
```

Socrative: Aufgabe 1/Vigenère

`https://b.socrative.com/login/student/`

Room: `SICHERHEIT`

- App um Quiz durchzuführen
- Zugang durch Browser oder App
- Als Quizteilnehmer kein Account notwendig.

Übungsblatt 1 – Aufgabe 1

Lösungsvorschlag zu Aufgabe 1. Beim Brechen von Vigènere-Chiffraten gehen wir wie folgt vor:

- Wir raten die Länge ℓ des Schlüssels $K = K_1 K_2 \dots K_\ell$ (oder wir nutzen die Kasiski-, Friedmann- oder die Autokorrelations-Methode).
- Wir teilen das Chiffrat in ℓ Teile auf, z.B. für $\ell = 5$:

  MZMTELFMPLWUFEWXTZPCJDQPBVUKSIEE...

  MZMTELFMPLWUFEWXTZPCJDQPBVUKSIEE...

  ...

  MZMTELFMPLWUFEWXTZPCJDQPBVUKSIEE...
- Wir führen eine Frequenzanalyse auf jedem Teiltext durch.

```
A: **********************************
B: ******
C: ***********
D: ******************
E: **************************************************
F: *********
G: ********
H: ************************
I: ****************************
J: *
K: ***
L: ****************
M: **********
N: ***************************
O: ********************************
P: ********
Q:
R: ************************
S: ***************************
T: ************************************
U: ***********
V: ****
W: *********
X: *
Y: ********
Z:
```

Erwartete Buchstabenhäufigkeit im Englischen (Quelle: Wikipedia).

Übungsblatt 1 – Aufgabe 1

Vorgehen:

- Wir vergleichen die natürliche Alphabetverteilung mit der Verteilung, die aus der Frequenzanalyse hervorging.

Die Buchstabenverteilung für den obigen Chiffretext und ersten Teiltext, der mit K_1 verschlüsselt wurde, sieht wie folgt aus:

```
A: **********
B:
C: ******
D: **
E: ********************************
F: ******
G: ******************
H: **********
I: ************************************************************
I: ************************************************************
J: ****
K: ******
L: ************************
M: ***************************
N:
O: ********
P: **************
Q: ******
R: ******************************
S: ************************************
T: ****************
U:
V: ****************
W: ****************************
X: ********************************
Y: ********
Z: ****
```

Übungsblatt 1 – Aufgabe 1

- Vermutlich wurde `E` (der häufigste Buchstabe im natürlichen Alphabet) auf `I` abgebildet. Damit ist $\texttt{I} - \texttt{E} = \texttt{E}$ der wahrscheinlichste Kandidat für K_1.
- Wir folgen dieser Strategie für K_2 bis K_5 und erhalten einen Schlüsselkandidaten: `EMIAY`
- Beim Entschlüsseln des Chiffretexts mit diesem Schlüssel erhalten wir folgendes Ergebnis:
  `INETGHTEPNSIXEYTHRPEFRIPDRICSKASIDKIWADTHEF`
  `TRSTTZPUBLTSHASFCCESDFULGPNERAWATTANKONTSEV`
  `IGPNERENIPHECEARL ...`
- Der Schlüssel ist also noch nicht korrekt, also betrachten wir z.B. die Stelle 4 genauer:

Die Buchstabenverteilung für den obigen Chiffretext und Teiltext, der mit K_4 verschlüsselt wurde, sieht wie folgt aus:

```
A: ****
B:
C: **********************
D: ********************************
E: ************************************************************
E: ************************************************************
F: **************
G:
H: ******
I:
J: ********
K:
L: ************************
M: **************
N: **************
O: ******************************
P: ********************************************
P: ********************************************
Q: ******
R: ********
S: ****************************
T: *************************
U:
V: ******
W: **************
X: ******
Y: **********************
Z: **************
```

Übungsblatt 1 – Aufgabe 1

- Der Buchstabe E könnte beim Verschlüsseln auch auf den zweithäufigsten Buchstaben P abgebildet worden sein. Damit wäre $\mathtt{P} - \mathtt{E} = \mathtt{L}$ ein Kandidat für K_4.
- Entschlüsseln mit dem Schlüssel EMILY ergibt (unter Hinzufügen von Groß-/Kleinschreibung, Leer- und Satzzeichen und Ersetzen von Zahlwörtern) den Klartext:

In 1863 Friedrich Kasiski was the first to publish a successful general attack on the Vigenère cipher. Earlier attacks relied on knowledge of the plaintext, or use of a recognizable word as a key. Kasiski's method had no such dependencies. Kasiski was the first to publish an account of the attack, but it is clear that there were others who were aware of it. In 1854, Charles Babbage was goaded into breaking the Vigenère cipher when John Hall Brock Thwaites submitted a 'new' cipher to the Journal of the Society of the Arts....

http://en.wikipedia.org/wiki/Vigenère_cipher

Socrative: Blockchiffren

`https://b.socrative.com/login/student/`

Room: `SICHERHEIT`

- App um Quiz durchzuführen
- Zugang durch Browser oder App
- Als Quizteilnehmer kein Account notwendig.

Übungsblatt 1 – Aufgabe 2

Wir wissen, dass die Blockchiffre $(\mathsf{E}, \mathsf{D})\colon \{0,1\}^8 \times \{0,1\}^4 \to \{0,1\}^4$ bei Eingabe eines festen Schlüssels K_0 eine Eingabe M wie folgt auf eine Ausgabe $C := \mathsf{E}(K_0, M)$ abbildet:

M	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001
C	0000	0001	1001	1110	1111	1011	0111	0110	1101	0010
M	1010	1011	1100	1101	1110	1111				
C	1100	0101	1010	0100	0011	1000				

Verschlüsseln Sie die Klartexte $M_1 = 0100\,0111\,0100\,0001$ und $M_2 = 0100\,1101\,0100\,0001$ unter K_0 in den Betriebsmodi ECB, CBC, und CTR. Wählen Sie, falls nötig, einen geeigneten Initialisierungsvektor.

Übungsblatt 1 – Aufgabe 2

M	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001
C	0000	0001	1001	1110	1111	1011	0111	0110	1101	0010
M	1010	1011	1100	1101	1110	1111				
C	1100	0101	1010	0100	0011	1000				

CBC-Mode:

- $C_0 := IV$
- $C_i := \mathrm{E}(K_0, M_{1,i} \oplus C_{i-1})$
- $M_1 = 0100\,0111\,0100\,0001$

$C_0 = IV = 0000$
$C_1 = \mathrm{E}(K_0, 0100 \oplus 0000) = \mathrm{E}(K_0, 0100) = 1111$
$C_2 = \mathrm{E}(K_0, 0111 \oplus 1111) = \mathrm{E}(K_0, 1000) = 1101$
$C_3 = \mathrm{E}(K_0, 0100 \oplus 1101) = \mathrm{E}(K_0, 1001) = 0010$
$C_4 = \mathrm{E}(K_0, 0001 \oplus 0010) = \mathrm{E}(K_0, 0011) = 1110$
$\Rightarrow C = 1111\,1101\,0010\,1110$
(Rest analog)

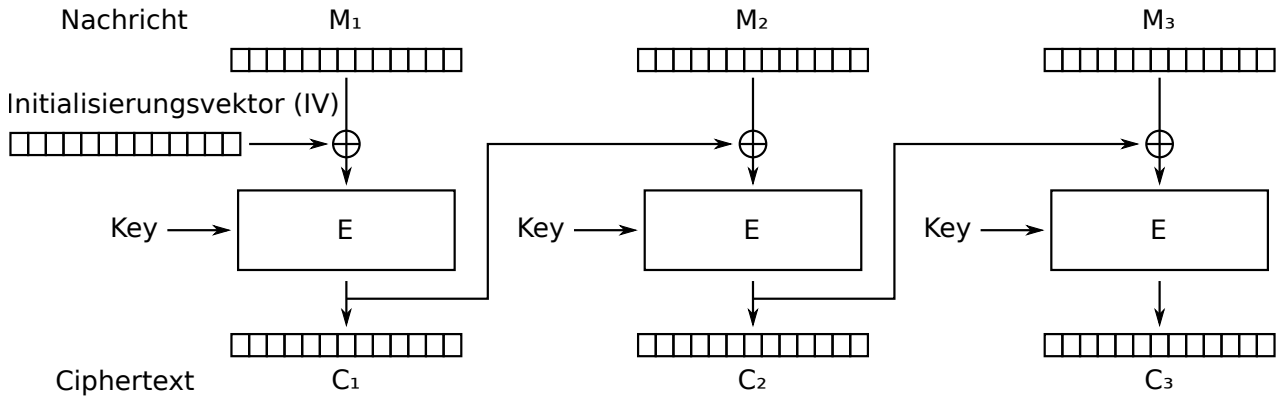
Übungsblatt 1 – Aufgabe 2

Worauf sollte bei der Wahl eines Initialisierungsvektors IV in den einzelnen Modi geachtet werden?

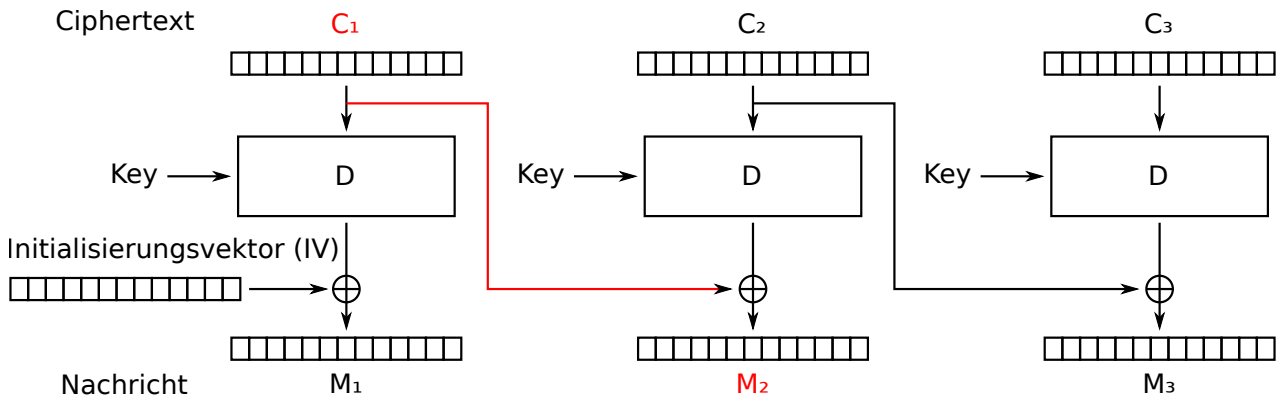
Der IV muss für jede Verschlüsselung neu zufällig gleichverteilt gewählt werden!
(Warum: Siehe nächstes Übungsblatt)

Demo: Verwundbarkeit von CBC

Verschlüsselung:



Entschlüsselung:

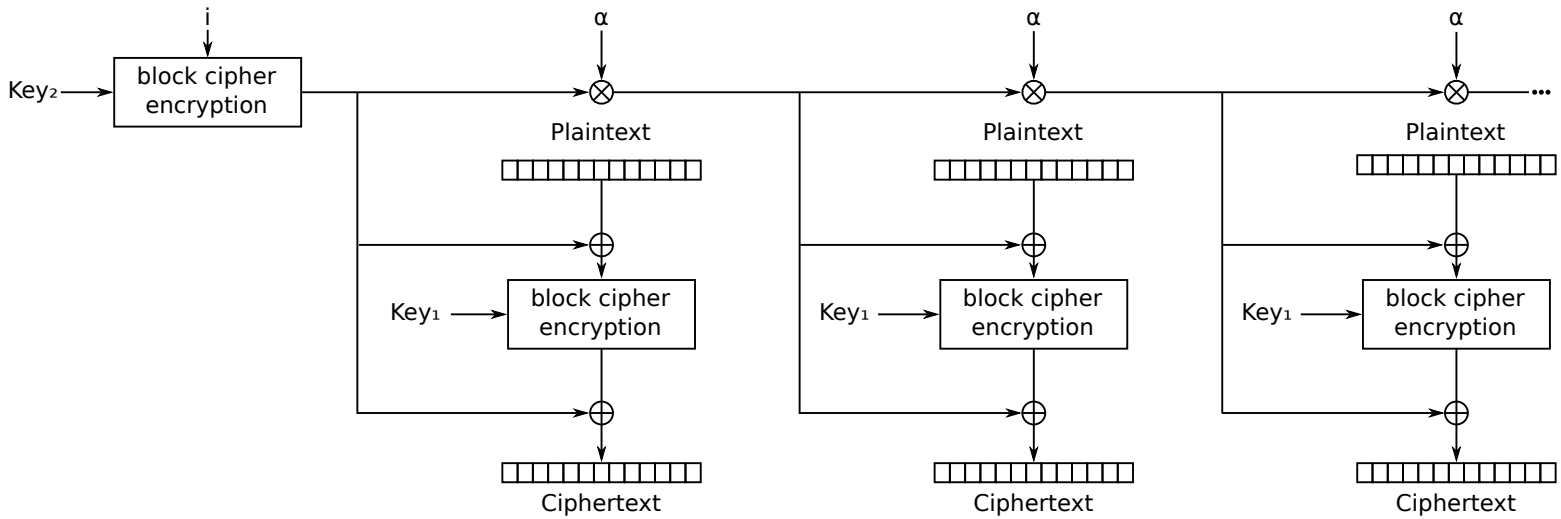


Hier: Setze $C_1 := C_{1,\text{alt}} \oplus M_2 \oplus M'$, dann entschlüsselt C_2 zu M'.

Quelle: Wikipedia

Betriebsmodus: Xor-Encrypt-Xor (XEX)

Benutze verschiedene Schlüssel für Erzeugung des Initialisierungsvektors und für eigentliche Block-Verschlüsselung.



XEX mode encryption

- $IV := \mathrm{E}(k_2, i)$ (i ist die Nummer des Sektors)
- $X_j := IV \otimes \alpha^{j-1}$
- $C_j := \mathrm{E}\left(k_1, (M_j \oplus X_j)\right) \oplus X_j$

Quelle: Wikipedia

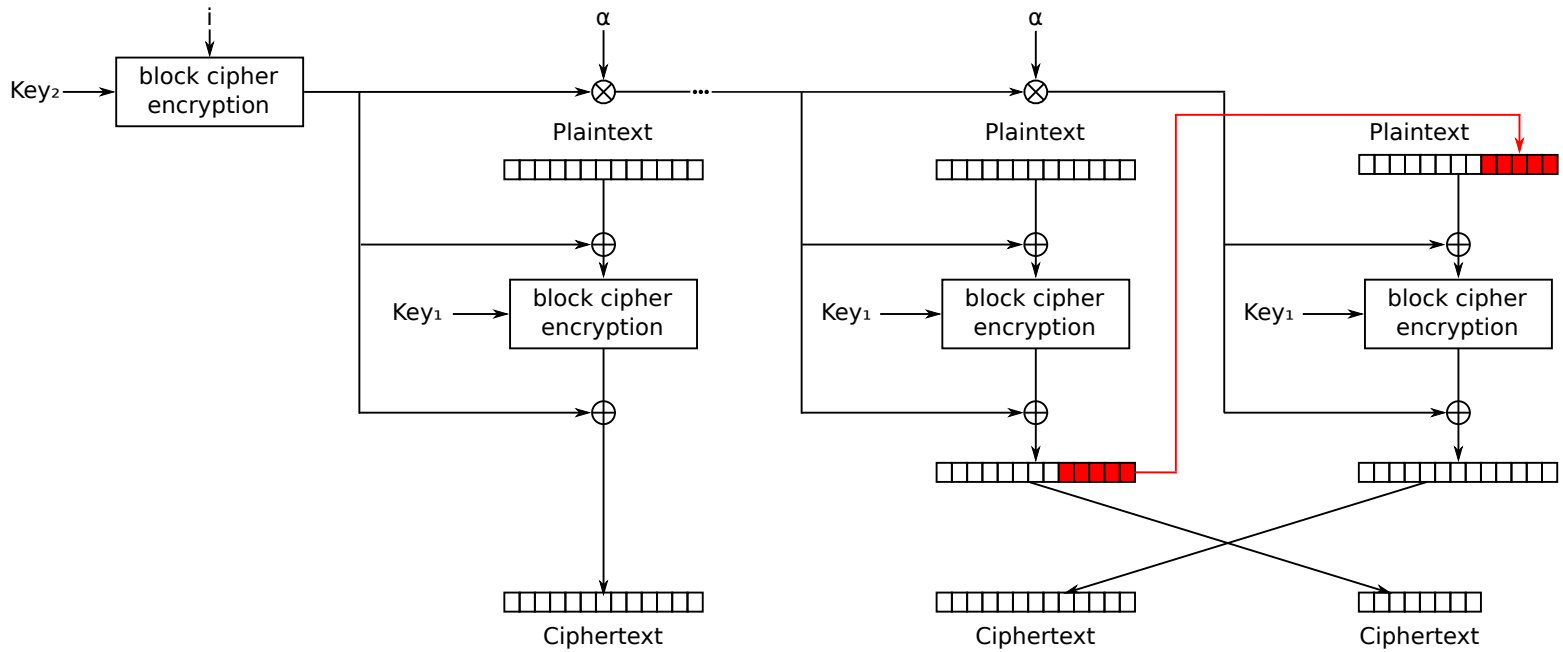
Betriebsmodus: Xor-Encrypt-Xor (XEX)

Vor-/Nachteile:

- Chiffrate abhängig von Sektor-Adresse i und Position des Blocks j innerhalb des Sektors
- Ver- und Entschlüsselung parallelisierbar
- keine XOR-Homomorphie, **aber:** keine explizite Prüfsumme wie GCM

Betriebsmodus: XEX-based tweaked-codebook mode with ciphertext stealing (XTS)

Ciphertext-stealing ermöglicht es den Betriebsmodus auf Nachrichten anzuwenden, die nicht durch die Blockgröße teilbar sind



XEX with tweak and ciphertext stealing (XTS) mode encryption

Quelle: Wikipedia

Betriebsmodus: XEX-based tweaked-codebook mode with ciphertext stealing (XTS)

Vor-/Nachteile:

- wie XEX
- Nachrichten müssen nicht durch Blocklänge teilbar sein

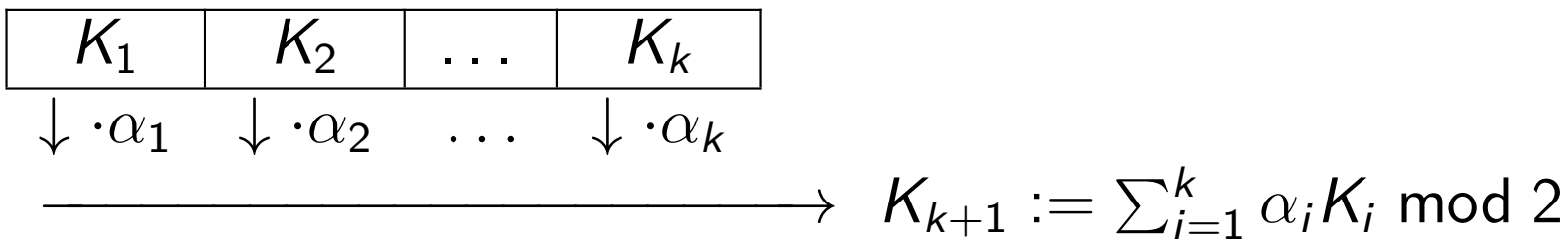
Übungsblatt 1 – Aufgabe 4

Linear feedback shift register (LFSR):

- Schlüssel K bitweise in Speicherzellen angeordnet:

K_1	K_2	$\ldots$	K_k

- Bei Zustandsupdate wird neues Bit erzeugt ($\alpha_i \in \{0, 1\}$):



- Ausgabe ist $c_1 := K_1$, neuer Zustand ist:

K_2	$\ldots$	K_k	K_{k+1}

Socrative: Aufgabe 4/LFSR

`https://b.socrative.com/login/student/`

Room: `SICHERHEIT`

- App um Quiz durchzuführen
- Zugang durch Browser oder App
- Als Quizteilnehmer kein Account notwendig.

Übungsblatt 1 – Aufgabe 4

- Wir müssen davon ausgehen, dass die Bits des initialen Zustands $K_1, \ldots, K_k$ unabhängig gewählt sind.
  - Die ersten k Ausgabe-Bits sind genau $K_1, \ldots, K_k$.
  - Wir können also nicht hoffen eines dieser Bits vorherzusagen.
- Wie sieht es mit einem der nächsten k Bits aus?

Übungsblatt 1 – Aufgabe 4

- Jedes weitere Ausgabe-Bit c_{k+j} für $j \geq 1$ kann als Produkt geschrieben werden:

$$c_{k+j} = K_{k+j} = (K_j, \ldots, K_{k+j-1}) \cdot \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{pmatrix}$$

- Also ergeben sich die Ausgabe-Bits $c_{k+1}, \ldots, c_{2k}$ als Matrix-Vektor-Produkt:

$$\underbrace{\begin{pmatrix} K_1 & \ldots & K_k \\ K_2 & \ldots & K_{k+1} \\ \vdots & & \vdots \end{pmatrix}}_{=:A} \cdot \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{pmatrix} = \begin{pmatrix} c_{k+1} \\ \vdots \\ c_{2k} \end{pmatrix}.$$

Übungsblatt 1 – Aufgabe 4

- Fall 1: Die so entstehende Matrix A ist invertierbar (in $\left(\mathbb{Z}/2\mathbb{Z}\right)^{k\times k}$).
  - Wir können die geheimen Gewichte $\alpha_1, \ldots, \alpha_k$ direkt berechnen:

$$\begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{pmatrix} = A^{-1} \cdot \begin{pmatrix} c_{k+1} \\ \vdots \\ c_{2k} \end{pmatrix}.$$

  - Jedes weitere Ausgabe-Bit kann vorhergesagt werden.

Übungsblatt 1 – Aufgabe 4

- Fall 2: Die Matrix A ist nicht invertierbar.
  - Dann gibt es ein (kleinstes) j, sodass die j-te Zeile Z_j von A eine Linearkombination der Zeilen $Z_1, \ldots, Z_{j-1}$ ist.

$$Z_j = \sum_{i=1}^{j-1} \gamma_i \cdot Z_i$$

  - Dann können wir das Ausgabe-Bit c_{k+j} vorhersagen!

$$c_{k+j} = Z_j \cdot \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{pmatrix} = (\sum_{i=1}^{j-1} \gamma_i \cdot Z_i) \cdot \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{pmatrix}$$

$$= \sum_{i=1}^{j-1} \gamma_i \cdot \underbrace{Z_i \cdot \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{pmatrix}}_{=c_{k+i}}.$$

$$\begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{pmatrix} \quad \sum_{i=1}^{j-1} \quad \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{pmatrix}$$

Wann ist ein Baustein sicher?

- Problem:
  - In vielen Fällen nicht offensichtlich, ob ein Baustein sicher ist.
  - (Im Gegensatz zu z.B. der Effizienz von Algorithmen.)
- Lösung: Beweisbare Sicherheit

Was soll ein kryptographischer Baustein eigentlich leisten?

- Es ist wichtig, klar zu definieren, was wir mit “sicher” meinen.
  - In vielen Fällen fordern wir von einem “sicheren” Baustein, dass es für effiziente Angreifer nicht möglich ist ein bestimmtes Ziel zu erreichen.
  - (Mit “effizienten Angreifern” meinen wir Algorithmen mit polynomieller Laufzeit.)

Was soll ein kryptographischer Baustein eigentlich leisten?

Beispiel:

- Von einem Verschlüsselungsverfahren werden wir fordern, dass effiziente Angreifer Chiffrate nicht entschlüsseln können.
  - Um diese Eigenschaft zu beweisen, müssen wir mindestens annehmen, dass $\mathcal{P} \neq \mathcal{NP}$ ist.
  - Wieso?

Wie können wir Sicherheit beweisen?

- In vielen Fällen sind komplexitätstheoretische **Annahmen** nötig, um die Sicherheit von Bausteinen zu beweisen.
  - Die verwendeten Annahmen sind lange bekannt und gut untersucht (implizieren aber in vielen Fällen $\mathcal{P} \neq \mathcal{NP}$, können also mit aktuellen Techniken nicht bewiesen werden).
- Ein **Sicherheitsbeweis** sieht dann so aus:
  - Zeige, dass man aus einem ein erfolgreicher Angreifer auf den Baustein einen erfolgreichen Angreifer auf die zugrunde liegende Annahme konstruieren kann (Reduktion).
  - Mehr dazu in der nächsten Vorlesung

OTP – mehrfache Verwendung eines Schlüssels