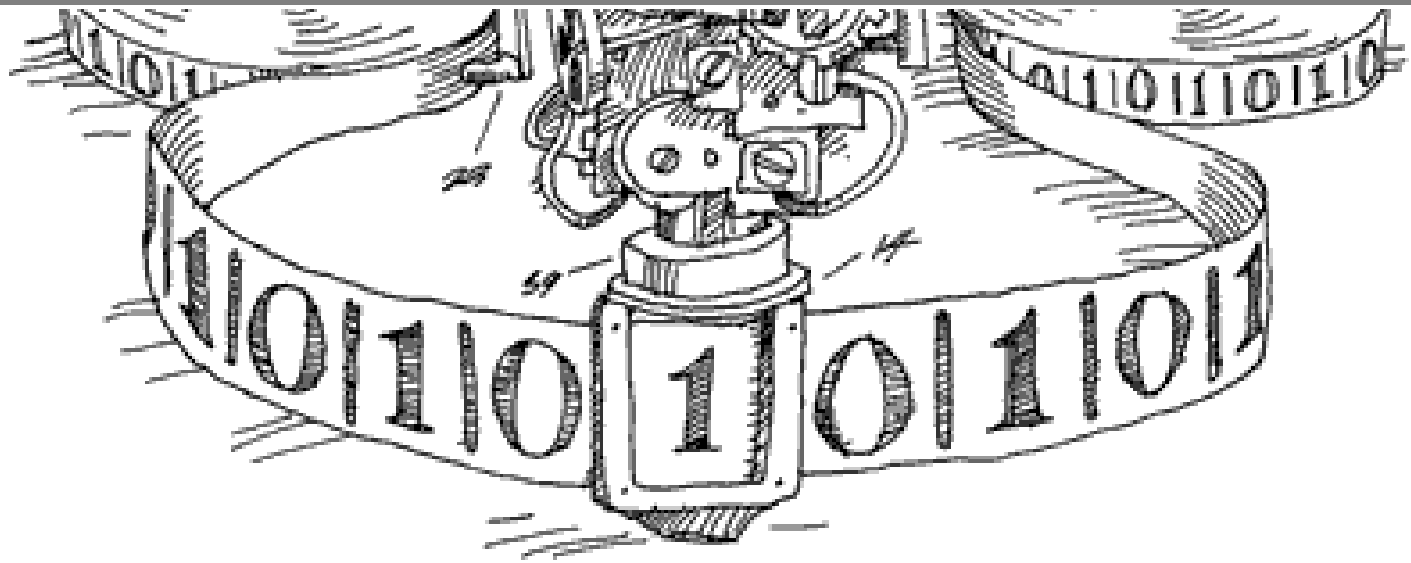
Karlsruher Institut für Technologie

Theoretische Grundlagen der Informatik

Tutorium 13

Institut für Kryptographie und Sicherheit



Best of Übungsblatt 7

- Abgaben: 5 von 21 (23.8%)
- Mindestens 50% Punkte: 5 von 5 (100%)
- Durchschnittliche erreichte Punktzahl: 11.5

Aufgabenverteilung

Aufgabe 1: Gesamt: 18.25P (durchscnittlich 3.65)

Aufgabe 2: Gesamt: 11.5P (durchschnittlich 2.3)

Aufgabe 3: Gesamt: 15.75P (durchschnittlich 3.15)

Aufgabe 4: Gesamt: 5.5P (durchschnittlich 1.1)

Aufgabe 5: Gesamt: 3.75P (durchschnittlich 0.75)

Übungsschein erhalten: 9 von 21 (42%)

Systematische Codes

Damit überprüft werden kann, ob ein Wort richtig übertragen wurde, müssen zusätzlich Daten übermittelt werden.
Eine Möglichkeit der Fehlerprüfung ist die Verwendung von Generatormatrizen.
Dabei werden Wörter der festen Länge k mit Wörter der Länge k+r kodiert. Dazu wird eine Generatormatrix der Form

$$G = \begin{pmatrix} I_k \\ A \end{pmatrix}$$

verwendet, wobei I_k die $k \times k$-Einheitsmatrix ist und A eine $r \times k$-Matrix. Das kodierte Wort ω_{codiert} erhält man aus dem dazugehörigen Wort ω mittels der Formel $G\omega = \omega_{\text{codiert}}$.

Systematische Codes

Zu der Generatormatrix gehört eine Prüfmatrix

$$H = (A|I_r)$$

mit deren Hilfe sich das Syndrom $s = H\omega_{\text{codiert}}$ ausrechnen lässt.
Ist $s = 0$, wurde die Information im Rahmen der Fehlerkorrektur richtig übertragen. Ist $s \neq 0$ vergleicht man s mit den Spalten von H.
Sei H_k die k-te Spalte von H.

- Gilt $H_k = s$ für exakt ein k, dann ist das k-te Bit im gesendeten Wort falsch.
- Gilt $H_k = s$ für mehrere k, dann ist eine ungerade Anzahl der dazugehörigen Bits falsch.
- Gilt $H_k \neq s$ für alle k, dann sind definitiv mehrere Bits falsch übertragen worden.

Aufgabe B13 A2

Sei $\mathcal{C}$ ein binärer Code, der durch die folgende Generatormatrix gegeben ist:

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Dekodieren Sie die folgenden empfangenen Wörter!

1. $w_1 = (\ 1 \quad 1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1\ )$
2. $w_2 = (\ 0 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1\ )$
3. $w_3 = (\ 0 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0\ )$

Hamming-Code

- Eine Codierung zur Erkennung von bis zu 2-bit-Fehlern und Korrektur von 1-bit-Fehlern
- n Paritätsbits sichern 2^n bits - 1 (Code für Fehlerfrei) - n

Hamming-Code

- Eine Codierung zur Erkennung von bis zu 2-bit-Fehlern und Korrektur von 1-bit-Fehlern
- n Paritätsbits sichern 2^n bits - 1 (Code für Fehlerfrei) - n

Bildlich:

p_1 p_2 d_3

p_1 p_2 d_3

p_1 p_2 d_3

Hamming-Code

- Eine Codierung zur Erkennung von bis zu 2-bit-Fehlern und Korrektur von 1-bit-Fehlern
- n Paritätsbits sichern 2^n bits - 1 (Code für Fehlerfrei) - n

Bildlich:

$p_1\ p_2\ d_3\ p_4\ d_5\ d_6\ d_7$
$p_1\ p_2\ d_3\ p_4\ d_5\ d_6\ d_7$
$p_1\ p_2\ d_3\ p_4\ d_5\ d_6\ d_7$
$p_1\ p_2\ d_3\ p_4\ d_5\ d_6\ d_7$

Hamming-Code

- Eine Codierung zur Erkennung von bis zu 2-bit-Fehlern und Korrektur von 1-bit-Fehlern
- n Paritätsbits sichern 2^n bits - 1 (Code für Fehlerfrei) - n

Bildlich:

$p_1\ p_2\ d_3\ p_4\ d_5\ d_6\ d_7\ p_8\ d_9\ d_{10}\ d_{11}\ d_{12}\ d_{13}\ d_{14}\ d_{15}$

$p_1\ p_2\ d_3\ p_4\ d_5\ d_6\ d_7\ p_8\ d_9\ d_{10}\ d_{11}\ d_{12}\ d_{13}\ d_{14}\ d_{15}$

$p_1\ p_2\ d_3\ p_4\ d_5\ d_6\ d_7\ p_8\ d_9\ d_{10}\ d_{11}\ d_{12}\ d_{13}\ d_{14}\ d_{15}$

$p_1\ p_2\ d_3\ p_4\ d_5\ d_6\ d_7\ p_8\ d_9\ d_{10}\ d_{11}\ d_{12}\ d_{13}\ d_{14}\ d_{15}$

$p_1\ p_2\ d_3\ p_4\ d_5\ d_6\ d_7\ p_8\ d_9\ d_{10}\ d_{11}\ d_{12}\ d_{13}\ d_{14}\ d_{15}$

Beispiel

1100110 S

Beispiel

1100110	S
1100110	0

Beispiel

1100110	S
1100110	0
1100110	0

Beispiel

1100110	S
1100110	0
1100110	0
1100110	0

Beispiel

1100110 S
1100110 0
1100110 0
1100110 0
$\Rightarrow$ Keine Fehler! Datenwort = 0110

Beispiel

1100110 S
1100110 0
1100110 0
1100110 0
$\Rightarrow$ Keine Fehler! Datenwort = 0110

1100010 S

Beispiel

1100110 S
1100110 0
1100110 0
1100110 0
$\Rightarrow$ Keine Fehler! Datenwort = 0110

1100010 S
1100010 1

Beispiel

1100110	S
1100110	0
1100110	0
1100110	0

$\Rightarrow$ Keine Fehler! Datenwort = 0110

1100010	S
1100010	1
1100010	0

Beispiel

1100110	S
1100110	0
1100110	0
1100110	0

$\Rightarrow$ Keine Fehler! Datenwort = 0110

1100010	S
1100010	1
1100010	0
1100010	1

Beispiel

1100110 S
1100110 0
1100110 0
1100110 0
$\Rightarrow$ Keine Fehler! Datenwort = 0110

1100010 S
1100010 1
1100010 0
1100010 1
$\Rightarrow$ Fehler an der Stelle $2^0 + 2^2 = 5$
Korrektur: 1100110
$\Rightarrow$ repariertes Datenwort: 0110

Aufgabe B13 A3

Gegeben sei der $[7, 4]$-Hamming-Code $\mathcal{C}_H$ mit der Erzeugermatrix

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

und der Prüfmatrix

$$H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Dekodieren Sie die folgenden empfangenen Wörter!

1. $w_1 = (\ 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1\ )$
2. $w_2 = (\ 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1\ )$
3. $w_3 = (\ 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0\ )$
4. $w_4 = (\ 0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1\ )$

Verschlüsselungscodierungen

- Caesar-Chiffre: Rotiere alle Zeichen um n Stellen im Alphabet
  - $\Rightarrow$ n ist Schlüssel
- Vigenère-Chiffre: Rotiere alle Zeichen um K_n Stellen
  - K_n ist die $n$te Stelle des Schlüssels mod $|K|$
  - $|K|$ ist Schlüssel
- One-Time-Pad (OTP), wie Vigenère, aber $|K| = |M|$

Verbindung mit Informationtheorie

- Eine Chiffre ist perfect sicher, wenn $p(M|C) = p(M)$
  - $p(M)$ ist die Wahrscheinlichkeit die Nachricht M zu erraten
  - $p(M|C)$ ist die Wahrscheinlichkeit die Nachricht M zu erraten, wenn C bekannt ist.
  - Nach Satz von Bayes gilt dann: $p(C|M) = \frac{p(M|C) \cdot p(C)}{p(M)} = p(C)$
  - One-Time-Pads sind perfekt sicher
- Für perfekte Sicherheit gilt: $H(M|C) = H(M)$ also auch $I(M;C) = 0$

Commitments

Beispiel:

1. Alice sendet Bob ein verschlüsseltes Bit b (bspw. Münzwurfergebnis) ohne Schlüssel
2. Bob sendet Alice ein unverschlüsseltes Bit b' (bspw. seine Wette)
3. Alice sendet Bob den Schlüssel
4. $b \oplus b'$ könnte nun das Ergbnis des Münzwurfs sein (bei 1 gewinnt Alice, bei 0 Bob)

Begriffe:

Commitment $c = commit(b)$: c legt b fest, ohne den Wert von b offenzulegen

Unveiling $unveil(c)$: gibt den Wert von b, der durch c festegelegt wurde aus

Binding $p(unveil(commit(b)) \neq b)$ vernachlässigbar

Hiding $p(b|c) - p((1-b)|c)$ vernachläsigbar

Pedersen-Commitments

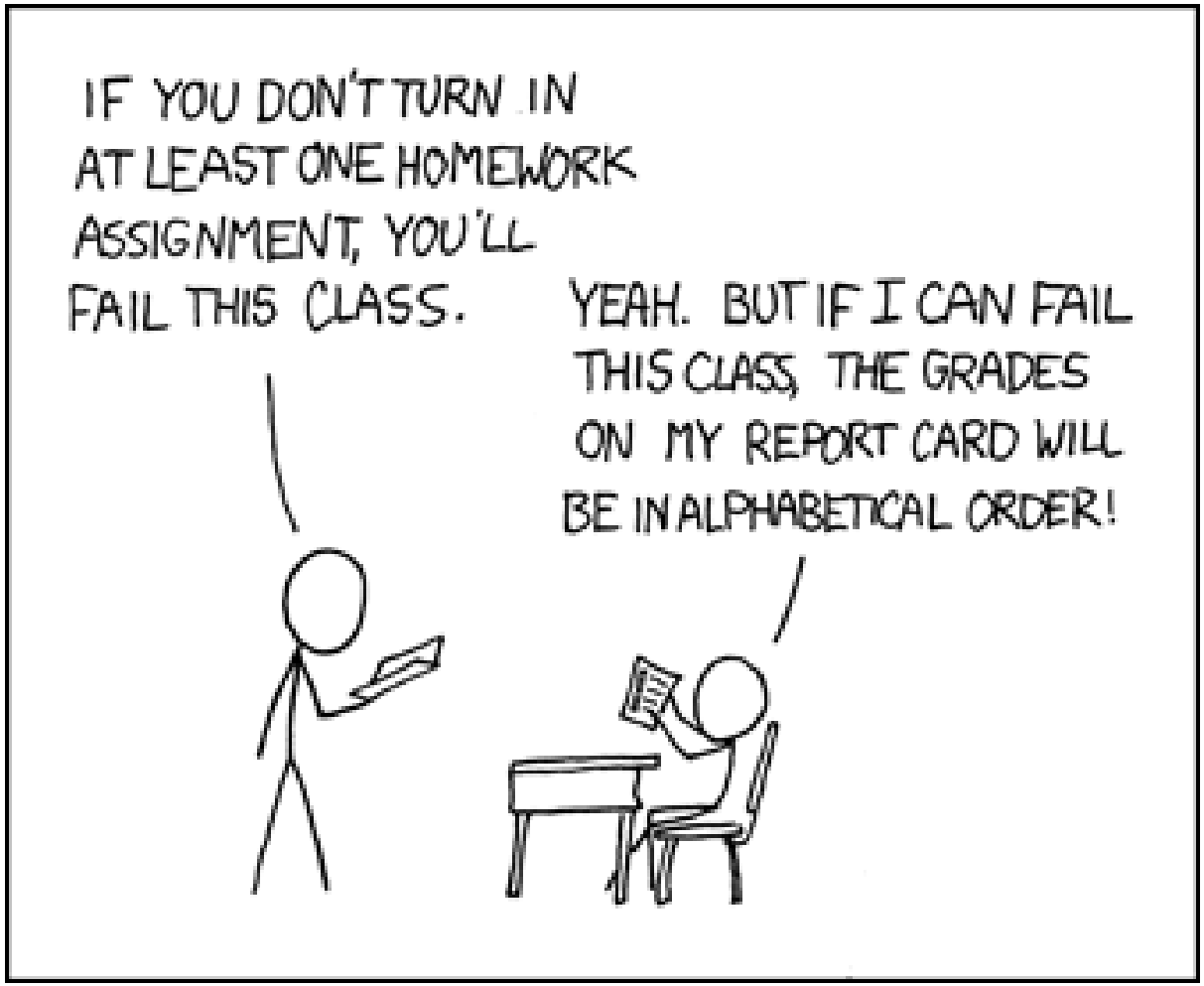
Sei G eine zyklische Gruppe, g und h Erzeuger von G und m die Nachricht. Dann sind:

- commit(m) = $g^m \cdot h^r$, wobei r zufällig gewählt wird
- unveil(c) = Gebe m und r bekannt

Pederson-Commitments erfüllen Binding und Hiding

- Die Sicherheit baut auf der Schwierigkeit des Berechnens von diskreten Logarithmen auf

Bis zum nächsten Mal!



Lizenzen