

Ausgangspunkt:

- ▶ NP-vollständige Probleme zur Konstruktion von *trapdoor*-Funktionen verwenden!
- ▶ Das (Syndrom-)Decodierungsproblem für lineare Codes ist NP-vollständig – also mutmasslich hoffnungslos schwierig!
- ▶ Man kennt grosse Klassen von linearen Codes (BCH-Codes, Goppa-Codes), für die es effiziente Decodieralgorithmen gibt!

Vorsicht! Andere Kryptosysteme auf der Basis von NP-vollständigen Problemen haben sich als nicht sicher erwiesen, so das auf dem Knapsack-Problem basierende System von MERKLE und HELLMAN.

Realisierung (setup):

- ▶ *Bob*
  - ▶ wählt einen  $t$ -Fehler korrigierenden linearen  $[n, k]$ -Code  $\mathcal{C}$  mit effizienter Decodierung — z.B. GOPPA-Code mit Parametern

$$n = 2^m, k = n - m \cdot t, d \geq 2t + 1,$$

realistischerweise etwa (McELIECE)

$$m = 10, n = 2^{10} = 1024, t = 50, k = 524.$$

$G \in \mathbb{B}^{k \times n}$  sei Generatormatrix für diesen Code.

NB: es gibt *vielen* GOPPA-Codes mit diesen Parametern!

- ▶ wählt ferner eine invertierbare Matrix ("scrambler")  $S \in \mathbb{B}^{k \times k}$  und eine Permutationsmatrix  $P \in \mathbb{B}^{n \times n}$ .
- ▶ berechnet  $\tilde{G} = S \cdot G \cdot P$  und gibt diese Matrix als öffentlichen Schlüssel bekannt.

Szenario:

- ▶ Teilnehmer *Bob* wählt – privat – einen  $t$ -Fehler korrigierenden linearen Code  $\mathcal{C}$ , für den es einen effizienten Decodieralgorithmus gibt.
- ▶ *Bob* "verdreh" (*scramble*) diesen Code  $\mathcal{C}$  mit einer nur ihm bekannten reversiblen Transformationen  $T$  zu einem linearen Code  $\tilde{\mathcal{C}} = T(\mathcal{C})$  mit mutmasslich NP-harter Decodierung, dessen Daten (Generatormatrix  $\tilde{G}$ ) er öffentlich bekannt gibt.
- ▶ *Alice* codiert Nachrichten für *Bob* mittels  $\tilde{G}$  und zufällig gewähltem Fehlervektor mit Gewicht  $\leq t$ .
- ▶ *Bob* rekonstruiert Nachricht mittels effizienter Decodierung für  $\mathcal{C}$ .
- ▶ *Eve* müsste das NP-harte Decodierungsproblem für  $\tilde{\mathcal{C}}$  lösen, um Nachricht zu erfahren!

Beachte:

- ▶ Die Matrix  $\tilde{G} = S \cdot G \cdot P$  ist Generatormatrix eines linearen Codes  $\tilde{\mathcal{C}}$ 
  - ▶ mit den gleichen Parametern  $[n, k, d]$  wie  $\mathcal{C}$
  - ▶ ohne (für Angreifer) erkennbare Struktur: für  $\tilde{\mathcal{C}}$  kommt nur Syndrom-Decodierung in Frage – und dies ist ein NP-vollständiges Problem!

### Realisierung (Übertragung)

- ▶ Alice will  $\mathbf{a} \in \mathbb{B}^k$  sicher zu Bob übertragen. Sie
  - ▶ wählt zufällig einen Vektor  $\mathbf{f} \in \mathbb{B}^n$  mit  $\|\mathbf{f}\| \leq t$ .
  - ▶ berechnet

$$\mathbf{b} = \mathbf{a} \cdot \tilde{G} \oplus \mathbf{f}$$

und überträgt  $\mathbf{b}$  zu Bob.

- ▶ Bob
  - ▶ berechnet

$$\mathbf{y} = \mathbf{b} \cdot P^{-1} = \underbrace{\mathbf{a} \cdot S \cdot G}_{\mathbf{x}} \oplus \underbrace{\mathbf{f} \cdot P^{-1}}_{\mathbf{z}}$$

- ▶ beachtet  $\mathbf{x} \cdot G \in \mathcal{C}$  und  $\|\mathbf{z}\| = \|\mathbf{f}\| \leq t$  und kann aus  $\mathbf{y}$  effizient  $\mathbf{x} \cdot G \in \mathcal{C}$  berechnen (Decodierungsalgorithmus für  $\mathcal{C}$  !!)
  - und somit auch  $\mathbf{x}$ .
- ▶ berechnet  $\mathbf{a} = \mathbf{x} \cdot S^{-1}$ .

### Realisierung (Angriff)

- ▶ Eve
  - ▶ steht vor dem Problem, aus Kenntnis von  $\mathbf{b}$  und  $\tilde{G}$  die ursprüngliche Nachricht  $\mathbf{a}$  zu berechnen, wobei nur  $d(\mathbf{a} \cdot \tilde{G}, \mathbf{b}) \leq t$  bekannt ist:
    - das ist das Decodierungsproblem für den Code  $\tilde{\mathcal{C}}$ !
  - ▶ mit den Beispieldaten von oben:

$$\sum_{s=0}^{50} \binom{1024}{s} = 3.362 \dots \cdot 10^{85}$$

Beispiel:  $\mathcal{C}$  = der [7,4]-HAMMING-Code (systematische Form)

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

$$S = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \quad P = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\tilde{G} = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\begin{aligned} \mathbf{a} &= [1 \ 0 \ 1 \ 1] \\ \mathbf{f} &= [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0] \\ \mathbf{b} &= \mathbf{a} \cdot \tilde{G} \oplus \mathbf{f} = [0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0] \\ \mathbf{y} &= \mathbf{b} \cdot P^{-1} = [0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1] \\ \mathbf{x} \cdot G &= [0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1] \\ \mathbf{x} &= [0 \ 0 \ 1 \ 0] \\ \mathbf{a} &= \mathbf{x} \cdot S^{-1} = [1 \ 0 \ 1 \ 1] \end{aligned}$$

