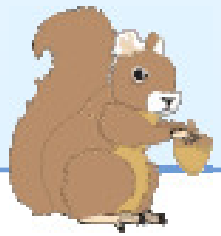
Kartenspielen ohne Karten

11. August 2006

Tobias Wahl

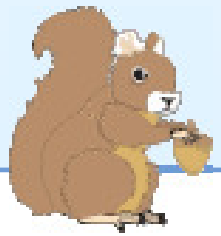
towa@towahl.de



Übersicht

Christian Schindelhauer:
“A Toolbox for Mental Card games” (1998)

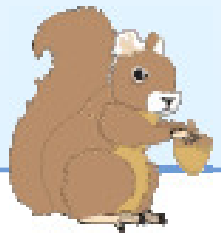
- Was?



Übersicht

Christian Schindelhauer:
“A Toolbox for Mental Card games” (1998)

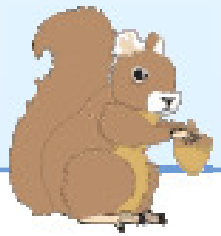
- Was?
- Zahlentheoretische Grundlagen



Übersicht

Christian Schindelhauer:
“A Toolbox for Mental Card games” (1998)

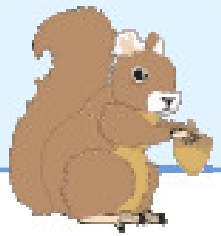
- Was?
- Zahlentheoretische Grundlagen
- Zero-Knowledge Protokolle



Übersicht

Christian Schindelhauer:
“A Toolbox for Mental Card games” (1998)

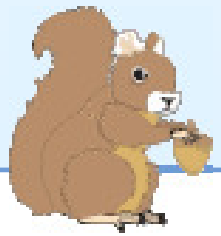
- Was?
- Zahlentheoretische Grundlagen
- Zero-Knowledge Protokolle
- Operationen auf Bits



Übersicht

Christian Schindelhauer:
“A Toolbox for Mental Card games” (1998)

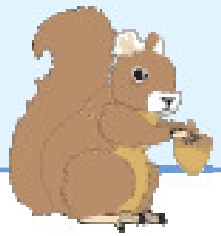
- Was?
- Zahlentheoretische Grundlagen
- Zero-Knowledge Protokolle
- Operationen auf Bits
- Karten



Übersicht

Christian Schindelhauer:
“A Toolbox for Mental Card games” (1998)

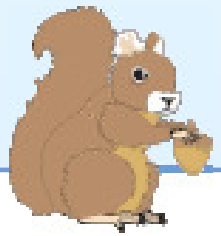
- Was?
- Zahlentheoretische Grundlagen
- Zero-Knowledge Protokolle
- Operationen auf Bits
- Karten
- Kartenstapel



Übersicht

Christian Schindelhauer:
“A Toolbox for Mental Card games” (1998)

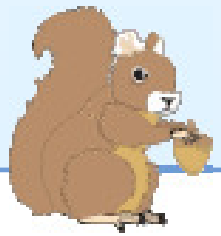
- Was?
- Zahlentheoretische Grundlagen
- Zero-Knowledge Protokolle
- Operationen auf Bits
- Karten
- Kartenstapel
- Nachteile



Was?

- Die Karten werden durch eine geeignete Datenstruktur dargestellt.

$$\underline{0 \quad 1 \quad 1 \quad \cdots \quad 0} \quad \text{Karte}$$

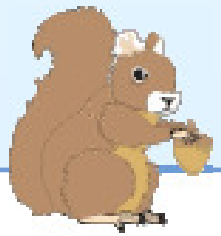


Was?

- Die Karten werden durch eine geeignete Datenstruktur dargestellt.

0	0	1	$\cdots$	1	1. Spieler
1	0	0	$\cdots$	1	2. Spieler
1	1	0	$\cdots$	0	3. Spieler
0	1	1	$\cdots$	0	Karte

- Jeder Spieler kennt eine Zeile aus der Karten-Matrix. Der Typ der Karte ergibt sich durch xor der einzelnen Zeilen.

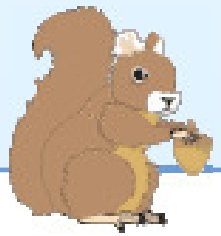


Was?

- Die Karten werden durch eine geeignete Datenstruktur dargestellt.

0	0	1	$\cdots$	1	1. Spieler
1	0	0	$\cdots$	1	2. Spieler
1	1	0	$\cdots$	0	3. Spieler
0	1	1	$\cdots$	0	Karte

- Jeder Spieler kennt eine Zeile aus der Karten-Matrix. Der Typ der Karte ergibt sich durch xor der einzelnen Zeilen.
- Um Mogeln zu verhindern, sind diese Zeilen öffentlich aber verschlüsselt jedem Spieler zugänglich.

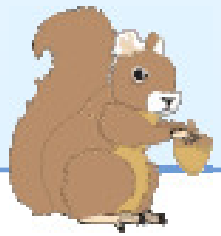


Was?

- Die Karten werden durch eine geeignete Datenstruktur dargestellt.

0	0	1	$\cdots$	1	1. Spieler
1	0	0	$\cdots$	1	2. Spieler
1	1	0	$\cdots$	0	3. Spieler
0	1	1	$\cdots$	0	Karte

- Jeder Spieler kennt eine Zeile aus der Karten-Matrix. Der Typ der Karte ergibt sich durch xor der einzelnen Zeilen.
- Um Mogeln zu verhindern, sind diese Zeilen öffentlich aber verschlüsselt jedem Spieler zugänglich.
- Zu gegebener Zeit muss ein Spieler seine Zeile entschlüsseln.

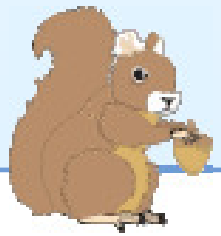


Was?

- Die Karten werden durch eine geeignete Datenstruktur dargestellt.

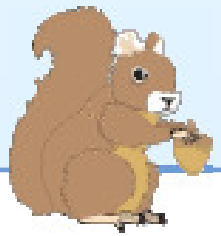
0	0	1	$\cdots$	1	1. Spieler
1	0	0	$\cdots$	1	2. Spieler
1	1	0	$\cdots$	0	3. Spieler
0	1	1	$\cdots$	0	Karte

- Jeder Spieler kennt eine Zeile aus der Karten-Matrix. Der Typ der Karte ergibt sich durch xor der einzelnen Zeilen.
- Um Mogeln zu verhindern, sind diese Zeilen öffentlich aber verschlüsselt jedem Spieler zugänglich.
- Zu gegebener Zeit muss ein Spieler seine Zeile entschlüsseln.
- Keine Koalition kann eine Karte gegen den Willen eines Spieler entschlüsseln.



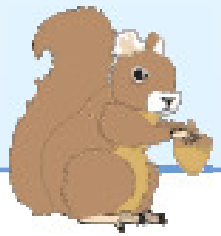
etwas Zahlentheorie

- $\mathbb{Z} := \{0, \pm 1, \pm 2, \ldots\}$ die Menge der *ganzen Zahlen* (Ring).



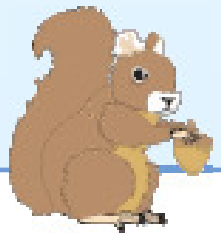
etwas Zahlentheorie

- $\mathbb{Z} := \{ 0, \pm 1, \pm 2, \ldots \}$ die Menge der *ganzen Zahlen* (Ring).
- $\mathbb{Z}_n := \mathbb{Z}/n\mathbb{Z} \cong \{ 0, 1, \ldots, n-1 \}$ die *Restklassen modulo* n (Ring).



etwas Zahlentheorie

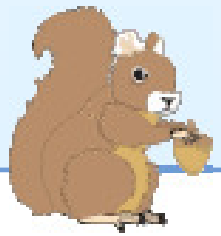
- $\mathbb{Z} := \{ 0, \pm 1, \pm 2, \ldots \}$ die Menge der *ganzen Zahlen* (Ring).
- $\mathbb{Z}_n := \mathbb{Z}/n\mathbb{Z} \cong \{ 0, 1, \ldots, n-1 \}$ die *Restklassen modulo* n (Ring).
- $\mathbb{Z}_n^\star := \{ x + n\mathbb{Z} \mid 1 \in \mathrm{ggt}(x, n) \}$ die *primen Restklassen modulo* n (Gruppe)



etwas Zahlentheorie

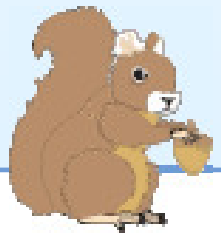
- $\mathbb{Z} := \{ 0, \pm 1, \pm 2, \ldots \}$ die Menge der *ganzen Zahlen* (Ring).
- $\mathbb{Z}_n := \mathbb{Z}/n\mathbb{Z} \cong \{ 0, 1, \ldots, n-1 \}$ die *Restklassen modulo* n (Ring).
- $\mathbb{Z}_n^\star := \{ x + n\mathbb{Z} \mid 1 \in \mathrm{ggt}(x, n) \}$ die *primen Restklassen modulo* n (Gruppe)
- Für p prim: $\left(\frac{x}{p}\right) := \begin{cases} 0, & p|x \\ 1, & \text{es gibt } y \text{ mit } y^2 \equiv x \bmod p, \\ -1, & \text{es gibt kein } y \text{ mit } y^2 \equiv x \bmod p \end{cases}$

  das *Legendre-Symbol*.



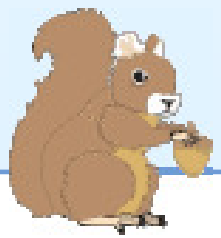
etwas Zahlentheorie

- $\mathbb{Z} := \{ 0, \pm 1, \pm 2, \ldots \}$ die Menge der *ganzen Zahlen* (Ring).
- $\mathbb{Z}_n := \mathbb{Z}/n\mathbb{Z} \cong \{ 0, 1, \ldots, n-1 \}$ die *Restklassen modulo* n (Ring).
- $\mathbb{Z}_n^\star := \{ x + n\mathbb{Z} \mid 1 \in \mathrm{ggt}(x, n) \}$ die *primen Restklassen modulo* n (Gruppe)
- Für p prim: $\left(\frac{x}{p}\right) := \begin{cases} 0, & p|x \\ 1, & \text{es gibt } y \text{ mit } y^2 \equiv x \bmod p, \\ -1, & \text{es gibt kein } y \text{ mit } y^2 \equiv x \bmod p \end{cases}$ das *Legendre-Symbol*.
- $\left(\frac{x}{nm}\right) := \left(\frac{x}{n}\right)\left(\frac{x}{m}\right)$ das *Jacobi-Symbol*.



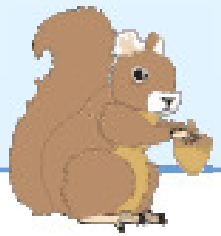
etwas Zahlentheorie

- $\mathbb{Z} := \{0, \pm 1, \pm 2, \ldots\}$ die Menge der *ganzen Zahlen* (Ring).
- $\mathbb{Z}_n := \mathbb{Z}/n\mathbb{Z} \cong \{0, 1, \ldots, n-1\}$ die *Restklassen modulo* n (Ring).
- $\mathbb{Z}_n^\star := \{x + n\mathbb{Z} \mid 1 \in \mathrm{ggt}(x, n)\}$ die *primen Restklassen modulo* n (Gruppe)
- Für p prim: $\left(\frac{x}{p}\right) := \begin{cases} 0, & p|x \\ 1, & \text{es gibt } y \text{ mit } y^2 \equiv x \bmod p, \\ -1, & \text{es gibt kein } y \text{ mit } y^2 \equiv x \bmod p \end{cases}$ das *Legendre-Symbol*.
- $\left(\frac{x}{nm}\right) := \left(\frac{x}{n}\right)\left(\frac{x}{m}\right)$ das *Jacobi-Symbol*.
- $\mathrm{qr}(x, n) := \begin{cases} 0, & \text{falls es ein } y \text{ mit } y^2 \equiv x \bmod n, \\ 1, & \text{sonst.} \end{cases}$



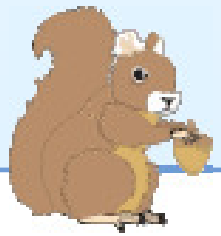
etwas Zahlentheorie

- $\mathbb{Z} := \{0, \pm 1, \pm 2, \ldots\}$ die Menge der *ganzen Zahlen* (Ring).
- $\mathbb{Z}_n := \mathbb{Z}/n\mathbb{Z} \cong \{0, 1, \ldots, n-1\}$ die *Restklassen modulo* n (Ring).
- $\mathbb{Z}_n^\star := \{x + n\mathbb{Z} \mid 1 \in \mathrm{ggt}(x, n)\}$ die *primen Restklassen modulo* n (Gruppe)
- Für p prim: $\left(\frac{x}{p}\right) := \begin{cases} 0, & p|x \\ 1, & \text{es gibt } y \text{ mit } y^2 \equiv x \bmod p, \\ -1, & \text{es gibt kein } y \text{ mit } y^2 \equiv x \bmod p \end{cases}$ das *Legendre-Symbol*.
- $\left(\frac{x}{nm}\right) := \left(\frac{x}{n}\right)\left(\frac{x}{m}\right)$ das *Jacobi-Symbol*.
- $\mathrm{qr}(x, n) := \begin{cases} 0, & \text{falls es ein } y \text{ mit } y^2 \equiv x \bmod n, \\ 1, & \text{sonst.} \end{cases}$
- $\mathbb{Z}_n^\circ := \{x + n\mathbb{Z} \mid 1 \in \mathrm{ggt}(x, n),\ \left(\frac{x}{n}\right) = 1\}$ (Gruppe)



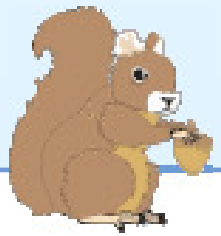
Zero-Knowledge

- Unterscheidung zwischen *Information* (eventuell implizit) und *Wissen* (explizit bzw. “schnell” berechenbar).



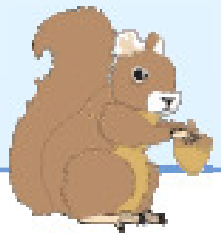
Zero-Knowledge

- Unterscheidung zwischen *Information* (eventuell implizit) und *Wissen* (explizit bzw. “schnell” berechenbar).
- Ein *Zero-Knowledge-Protokoll* ist eine *Zwei-Teilnehmer-Protokoll*, mittels dem ein *Beweiser* dem *Verifizierer* von etwas überzeugen kann, aber sonst kein Wissen übermittelt.



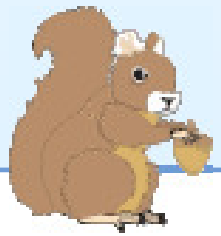
Zero-Knowledge

- Unterscheidung zwischen *Information* (eventuell implizit) und *Wissen* (explizit bzw. “schnell” berechenbar).
- Ein *Zero-Knowledge-Protokoll* ist eine *Zwei-Teilnehmer-Protokoll*, mittels dem ein *Beweiser* dem *Verifizierer* von etwas überzeugen kann, aber sonst kein Wissen übermittelt.
- Beispiel: Graph-Isomorphismus.
  G_1 und G_2 zwei Graphen (öffentlich), der Beweiser kennt einen Isomorphismus σ. Er will den Verifizieren davon überzeugen, dass G_1 und G_2 isomorph sind, ohne den Isomorphismus zu verraten.



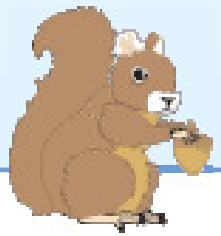
Zero-Knowledge

- Unterscheidung zwischen *Information* (eventuell implizit) und *Wissen* (explizit bzw. “schnell” berechenbar).
- Ein *Zero-Knowledge-Protokoll* ist eine *Zwei-Teilnehmer-Protokoll*, mittels dem ein *Beweiser* dem *Verifizierer* von etwas überzeugen kann, aber sonst kein Wissen übermittelt.
- Beispiel: Graph-Isomorphismus.
  G_1 und G_2 zwei Graphen (öffentlich), der Beweiser kennt einen Isomorphismus σ. Er will den Verifizieren davon überzeugen, dass G_1 und G_2 isomorph sind, ohne den Isomorphismus zu verraten.
- Warum ist das Zero-Knowledge?



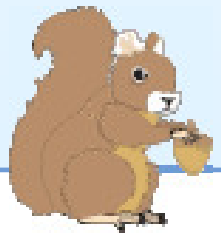
Operationen auf Bits

- Jeder Spieler i hat einen geheimen Schlüssel (p_i, q_i) aus zwei (großen) Primzahlen. Außerdem hat er einen öffentlichen Schlüssel (n_i, y_i) mit $n_i = p_i q_i$ (*Blum-Zahl*) und $y_i \in \mathbb{Z}^{\circ}_{n_i}$ aber $\mathsf{qr}(y_i, n_i) = 1$.



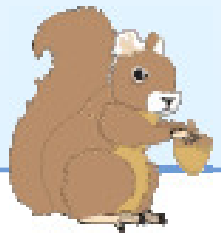
Operationen auf Bits

- Jeder Spieler i hat einen geheimen Schlüssel (p_i, q_i) aus zwei (großen) Primzahlen. Außerdem hat er einen öffentlichen Schlüssel (n_i, y_i) mit $n_i = p_i q_i$ (*Blum-Zahl*) und $y_i \in \mathbb{Z}^{\circ}_{n_i}$ aber $\mathsf{qr}(y_i, n_i) = 1$.
- Dass der öffentliche Schlüssel korrekt gebildet wurde, wird mit einem interaktiven Beweis gezeigt.



Operationen auf Bits

- Jeder Spieler i hat einen geheimen Schlüssel (p_i, q_i) aus zwei (großen) Primzahlen. Außerdem hat er einen öffentlichen Schlüssel (n_i, y_i) mit $n_i = p_i q_i$ (*Blum-Zahl*) und $y_i \in \mathbb{Z}^\circ_{n_i}$ aber $\mathsf{qr}(y_i, n_i) = 1$.
- Dass der öffentliche Schlüssel korrekt gebildet wurde, wird mit einem interaktiven Beweis gezeigt.
- Ein Bit b wird verschlüsselt durch ein $x \in \mathbb{Z}^\circ_{n_i}$ mit $\mathsf{qr}(x, n_i) = b$. Der i-te Spieler kann diese Bit "lesen" die anderen nicht.

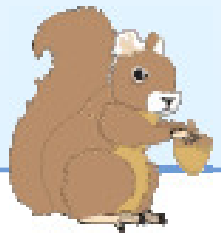


Operationen auf Bits

- Jeder Spieler i hat einen geheimen Schlüssel (p_i, q_i) aus zwei (großen) Primzahlen. Außerdem hat er einen öffentlichen Schlüssel (n_i, y_i) mit $n_i = p_i q_i$ (*Blum-Zahl*) und $y_i \in \mathbb{Z}^\circ_{n_i}$ aber $\mathsf{qr}(y_i, n_i) = 1$.
- Dass der öffentliche Schlüssel korrekt gebildet wurde, wird mit einem interaktiven Beweis gezeigt.
- Ein Bit b wird verschlüsselt durch ein $x \in \mathbb{Z}^\circ_{n_i}$ mit $\mathsf{qr}(x, n_i) = b$. Der i-te Spieler kann diese Bit "lesen" die anderen nicht.
- Die anderen Spieler können aber mit diesem Bit rechnen!

$$\mathsf{qr}(x_1 x_2) = \mathsf{qr}(x_1) \text{ xor } \mathsf{qr}(x_2).$$

Zu gegebenen b kann man sich leicht ein "zufälliges" z mit $\mathsf{qr}(z) = b$ beschaffen: $z = r^2 y_i^b$ (mit einem "zufälligem" r).

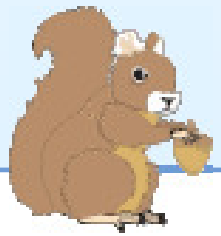


Operationen auf Bits

- Jeder Spieler i hat einen geheimen Schlüssel (p_i, q_i) aus zwei (großen) Primzahlen. Außerdem hat er einen öffentlichen Schlüssel (n_i, y_i) mit $n_i = p_i q_i$ (*Blum-Zahl*) und $y_i \in \mathbb{Z}^\circ_{n_i}$ aber $\mathsf{qr}(y_i, n_i) = 1$.
- Dass der öffentliche Schlüssel korrekt gebildet wurde, wird mit einem interaktiven Beweis gezeigt.
- Ein Bit b wird verschlüsselt durch ein $x \in \mathbb{Z}^\circ_{n_i}$ mit $\mathsf{qr}(x, n_i) = b$. Der i-te Spieler kann diese Bit "lesen" die anderen nicht.
- Die anderen Spieler können aber mit diesem Bit rechnen!

$$\mathsf{qr}(x_1 x_2) = \mathsf{qr}(x_1) \text{ xor } \mathsf{qr}(x_2).$$

  Zu gegebenen b kann man sich leicht ein "zufälliges" z mit $\mathsf{qr}(z) = b$ beschaffen: $z = r^2 y_i^b$ (mit einem "zufälligem" r).
- Entschlüsseln eines Bits durch Zero-Knowledge-Protokoll.

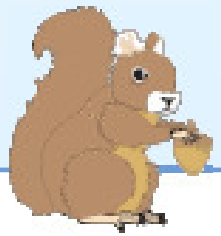


Karten (1)

- Eine *offene Karte* wird wie folgt erstellt:

$$\begin{array}{ccccc} y_1^{b_0} & y_1^{b_1} & y_1^{b_2} & \cdots & y_1^{b_n} \\ 1 & 1 & 1 & \cdots & 1 \\ 1 & 1 & 1 & \cdots & 1 \end{array}$$

Dabei ist $x = \sum_{i=0}^{n} b_i 2^i$ der *Typ* der Karte. (Beachte $\mathrm{qr}(1, n_j) = 0$ und $\mathrm{qr}(y_1^{b_i}, n_1) = b_i$).



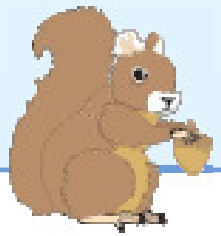
Karten (1)

- Eine *offene Karte* wird wie folgt erstellt:

$$\begin{array}{ccccc} y_1^{b_0} & y_1^{b_1} & y_1^{b_2} & \cdots & y_1^{b_n} \\ 1 & 1 & 1 & \cdots & 1 \\ 1 & 1 & 1 & \cdots & 1 \end{array}$$

  Dabei ist $x = \sum_{i=0}^{n} b_i 2^i$ der *Typ* der Karte. (Beachte $\mathrm{qr}(1, n_j) = 0$ und $\mathrm{qr}(y_1^{b_i}, n_1) = b_i$).

- Verschlüsseln einer Karte: Ein Spieler multipliziert alle Zahlen mit Werten der Art $r^2 y_i^b$, wobei in jeder Spalte eine gerade Anzahl an “Flips” ($b = 1$) stattfinden.



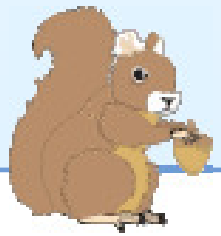
Karten (1)

- Eine *offene Karte* wird wie folgt erstellt:

$$\begin{array}{ccccc} y_1^{b_0} & y_1^{b_1} & y_1^{b_2} & \cdots & y_1^{b_n} \\ 1 & 1 & 1 & \cdots & 1 \\ 1 & 1 & 1 & \cdots & 1 \end{array}$$

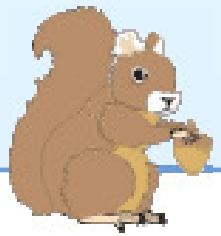
  Dabei ist $x = \sum_{i=0}^{n} b_i 2^i$ der *Typ* der Karte. (Beachte $\mathrm{qr}(1, n_j) = 0$ und $\mathrm{qr}(y_1^{b_i}, n_1) = b_i$).

- Verschlüsseln einer Karte: Ein Spieler multipliziert alle Zahlen mit Werten der Art $r^2 y_i^b$, wobei in jeder Spalte eine gerade Anzahl an “Flips” ($b = 1$) stattfinden.
- Nachweis, dass eine Karte durch Verschlüsseln aus einer anderen entstanden ist, ähnlich wie bei Graph-Isomorphie.



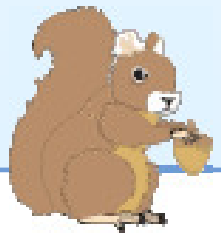
Karten (2)

- Die Karte einem Spieler zeigen: Alle anderen Spieler entschlüsseln ihre Zeile und beweisen, dass sie nicht mogeln.



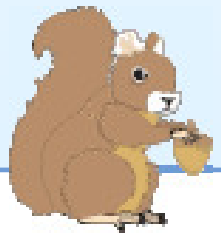
Karten (2)

- Die Karte einem Spieler zeigen: Alle anderen Spieler entschlüsseln ihre Zeile und beweisen, dass sie nicht mogeln.
- Die Karte aufdecken: Alle Spieler entschlüsseln ihre Zeile und beweisen, dass sie nicht mogeln.



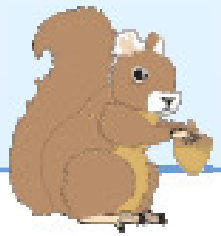
Karten (2)

- Die Karte einem Spieler zeigen: Alle anderen Spieler entschlüsseln ihre Zeile und beweisen, dass sie nicht mogeln.
- Die Karte aufdecken: Alle Spieler entschlüsseln ihre Zeile und beweisen, dass sie nicht mogeln.
- Bei entsprechender Codierung können bestimmte Eigenschaften der Karten entschlüsselt werden (nur einige Spalten entschlüsseln).



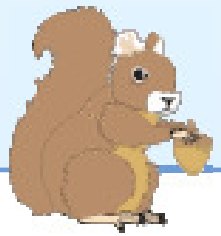
Kartenstapel

- Karten mischen: Alle Karten verschlüsseln und permutieren. Dann Nachweisen (per Zero-Knowledge), dass sich der Stapel nicht geändert hat (alle Spieler nacheinander).



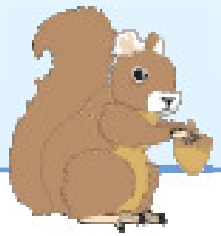
Kartenstapel

- Karten mischen: Alle Karten verschlüsseln und permutieren. Dann Nachweisen (per Zero-Knowledge), dass sich der Stapel nicht geändert hat (alle Spieler nacheinander).
- Karte in einen Stapel einfügen.



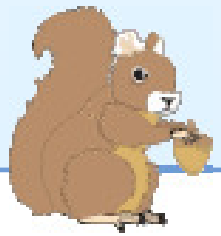
Kartenstapel

- Karten mischen: Alle Karten verschlüsseln und permutieren. Dann Nachweisen (per Zero-Knowledge), dass sich der Stapel nicht geändert hat (alle Spieler nacheinander).
- Karte in einen Stapel einfügen.
- Abheben.



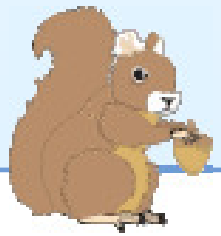
Kartenstapel

- Karten mischen: Alle Karten verschlüsseln und permutieren. Dann Nachweisen (per Zero-Knowledge), dass sich der Stapel nicht geändert hat (alle Spieler nacheinander).
- Karte in einen Stapel einfügen.
- Abheben.
- Vergleichen von Stapeln (“ich hab’ kein Pik”).



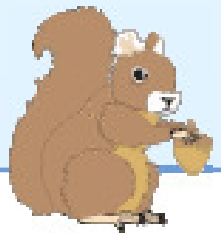
Kartenstapel

- Karten mischen: Alle Karten verschlüsseln und permutieren. Dann Nachweisen (per Zero-Knowledge), dass sich der Stapel nicht geändert hat (alle Spieler nacheinander).
- Karte in einen Stapel einfügen.
- Abheben.
- Vergleichen von Stapeln (“ich hab’ kein Pik”).
- Karten “verkleben”.



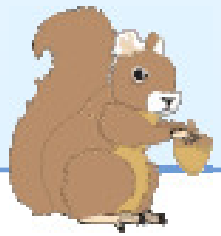
Nachteile

- unbewiesene Sicherheit



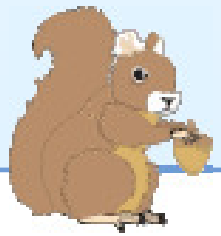
Nachteile

- unbewiesene Sicherheit
- geheimen Schlüssel “raten”



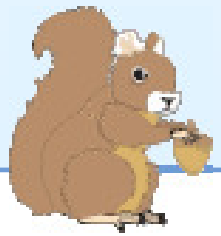
Nachteile

- unbewiesene Sicherheit
- geheimen Schlüssel “raten”
- Restrisiko



Nachteile

- unbewiesene Sicherheit
- geheimen Schlüssel “raten”
- Restrisiko
- Spieler zeigen sich gegenseitig Karten



Nachteile

- unbewiesene Sicherheit
- geheimen Schlüssel “raten”
- Restrisiko
- Spieler zeigen sich gegenseitig Karten
- Verbindungsabbruch