

Heiko Stamer <stamer@gaos.org>

Zero-Knowledge-Beweise

Mein Vortrag stellt anfangs das allgemeine Konzept der Zero-Knowledge-Beweise (Beweis von Wissen ohne seine Preisgabe/Transfer) am Modell der sogenannten Interaktiven-Turing-Maschinen (ITM) vor. Wir werden eine nützliche Klassifikation solcher probabilistischer Beweissysteme kurz kennenlernen und ihre Voraussetzungen am ITM-Modell klären. Über klassische Anwendungsbeispiele wie Ali Baba's Höhle, Graphenisomorphie oder den Identitätsbeweis von Feige-Fiat-Shamir wird der Bezug zur Kryptographie deutlich gemacht. Im zweiten Teil gehen wir auf die Verwendung solcher Beweise in 'Mentalen Kartenspielen' ein. Den Abschluß bildet die Präsentation einer Spielimplementierung für das bekannte Kartenspiel Skat.

Zero-Knowledge-Beweise: Keine Preisgabe des Beweisgegenstands

Fragestellung: Ist es möglich, eine Aussage/ Eigenschaft zu beweisen, ohne Informationen über den Beweisgegenstand zu offenbaren (ausgenommen ist die Gültigkeitsinformation der Aussage ansich)?

Antwort: Ja, es gibt probabilistische Beweissysteme, welche genau diese Forderung erfüllen → Zero-Knowledge-Protokolle.

Geschichte: 1535 Tartaglia (Lösungsverfahren Gl. dritten Grades), formal: 1985 Goldwasser, Micali, Rackoff [GoMR85]; 1985 Babai

Anwendung: Kryptographie (Identitätsbeweise, sichere Mehrpartei-berechnung, ‚trusted computing‘, mentale Spiele, ...)

Zero-Knowledge-Beweise: menschlich-intuitiv statt mathematisch-streng

- kein mathematisch-strenger Beweis (statisch und formal), sondern ‚Überzeugen‘ durch interaktive Kommunikation (vgl. dynamisches Frage-Antwort-Spiel, Challenge-Response-Systeme)
- handelnde Parteien in Zero-Knowledge-Beweisen:
 - **Peggy** (P), die Beweisführende
Möchte Victor von einer Aussage überzeugen, ohne ihren Beweisgegenstand bzw. das Geheimnis preisgeben zu müssen.
 - **Victor** (V), der Beweisverifizierende
Stellt Peggy eine Reihe von Fragen um herauszufinden, ob ihre Aussage bzgl. des Beweisgegenstandes gültig ist oder nicht.

Zero-Knowledge-Protokolle: fünf wichtige Eigenschaften

1. *Kein Wissenstransfer*: Viktor kann keine Informationen über das Geheimnis ableiten. (sonst Minimum-Disclosure-Protokoll)
2. *Korrektheit*: Peggy kann Viktor (mit hoher Wahrscheinlichkeit) nicht betrügen. (d. h. sie muß das Geheimnis wirklich kennen)
3. *Robustheit*: Viktor kann durch protokollunkonformes Verhalten keine zusätzlichen Informationen über das Geheimnis gewinnen.

Zero-Knowledge-Protokolle: fünf wichtige Eigenschaften (2)

4. Viktor kann sich gegenüber einer dritten Partei nicht als Peggy ausgeben oder die Partei von der Beweiskorrektheit überzeugen. (z. B. Videoaufzeichnung; aber ggf. man-in-the-middle Angriff)
5. *Vollständigkeit*: Viktor kann von der Gültigkeit aller wahren Aussagen (über einer problembezogenen Menge) überzeugt werden.

Zero-Knowledge-Beweise: Formalisierung durch ITM-Modell

Definition 1 (Interaktive-Turing-Maschine)

Sechsbändige Turing-Maschine mit

- *einem nur-lesbarem Eingabeband, einem nur-schreibbarem Ausgabeband*
- *einem les- und schreibbarem Arbeitsband,*
- *einem nur-lesbarem Zufallsband,
(enthält Sequenz von Zufallszeichen; Band nur unidirektional lesbar)*
- *einem nur-lesbarem und einem nur-schreibbarem Kommunikationsband.
(Interpretation: empfangene und gesendete Nachrichten der Maschine)*

Zero-Knowledge-Beweise: Formalisierung durch ITM-Modell

Definition 2 (interaktives Protokoll)

Geordnetes Paar (P, V) Interaktiver-Turing-Maschinen, mit dem selben Eingabeband, aber inversen Kommunikationsbändern.

P, V sind immer abwechselnd aktiv (V zuerst). Jeweils Schritte:

- 1. private Berechnung auf Grundlage der Bandinhalte*
- 2. eine Zeichenkette auf das nur-schreibare Kommunikationsband ausgeben*

Termination: Vermeiden von Schritt 2. (durch P und V auslösbar)

V hat akzeptierenden und ablehnenden Zustand (Protokollergebnis)

Zeitkomplexität: P unbeschränkt, V polynomial bzgl. Eingabelänge

Zero-Knowledge-Beweise: Formalisierung durch ITM-Modell

(P, V) akzeptiert Eingabe x , gdw. V im akzeptierenden Zustand hält.

Definition 3 (interaktives Beweissystem für Sprache L)

Interaktives Protokoll (P, V) mit

1. *Vollständigkeit: für jede Konstante $c > 0$ und genügend langes $x \in L$*

$$\Pr[(P, V) \text{ akzeptiert } x] > 1 - |x|^{-c} \quad (= 1 \text{ one-side-error})$$

2. *Korrektheit: für jede Konstante $c > 0$ und genügend langes $x \notin L$*

$$\forall P' : \Pr[(P', V) \text{ akzeptiert } x] < |x|^{-c}$$

(Je nach Literatur auch konstante Wahrscheinlichkeiten, weil parallele Ausführbarkeit des Beweises möglich. [hier ggf. Probleme mit Zero-Knowledge Eigenschaft])

Zero-Knowledge-Beweise: Fakten über interaktive Beweissysteme

Definition 4 (Klasse \mathcal{IP})

$$\mathcal{IP} = \{L : L \text{ hat interaktives Beweissystem}\}$$

Bemerkung 1

*Jede Sprache aus \mathcal{NP} hat ein interaktives Beweissystem. $\mathcal{NP} \subseteq \mathcal{IP}$
(Zufallsband von P, V bleibt unbenutzt; Interaktion immer unidirektional $P \rightarrow V$)*

Theorem 1 (Shamir 1990)

$$\mathcal{IP} = \mathcal{PSPACE} = \text{POLY-Arthur-Merlin-Spiele}$$

Zero-Knowledge-Beweise: Graphen Nicht-Isomorphie $\in \mathcal{IP}$

gemeinsame Eingabe: Graphen $G_1 = (V_1, E_1)$ und $G_2 = (V_2, E_2)$
o. B. d. A. $|V_1| = |V_2|$, $|E_1| = |E_2|$, $V_1 = V_2 = \{1, 2, \dots\}$

P -Behauptung: $G_1 \not\cong G_2$, **interaktiver Beweis** (P, V) :

V : $\sigma \stackrel{R}{:=} \{1, 2\}$, zufällige Knotenpermutation π , bildet isomorphe Kopie $H = \pi(G_\sigma)$

sendet H an P

P : empfängt H und sucht $\tau \in \{1, 2\}$ mit $H \cong G_\tau$ ($\tau := 0$, falls $H \not\cong G_1$ und $H \not\cong G_2$), [Wenn entgegen der Behauptung doch $G_1 \cong G_2$ gelten sollte, kann P die isomorphe Kopie nicht zuordnen und rät mit Wahrscheinlichkeit $1/2$ falsch.]

sendet τ an V

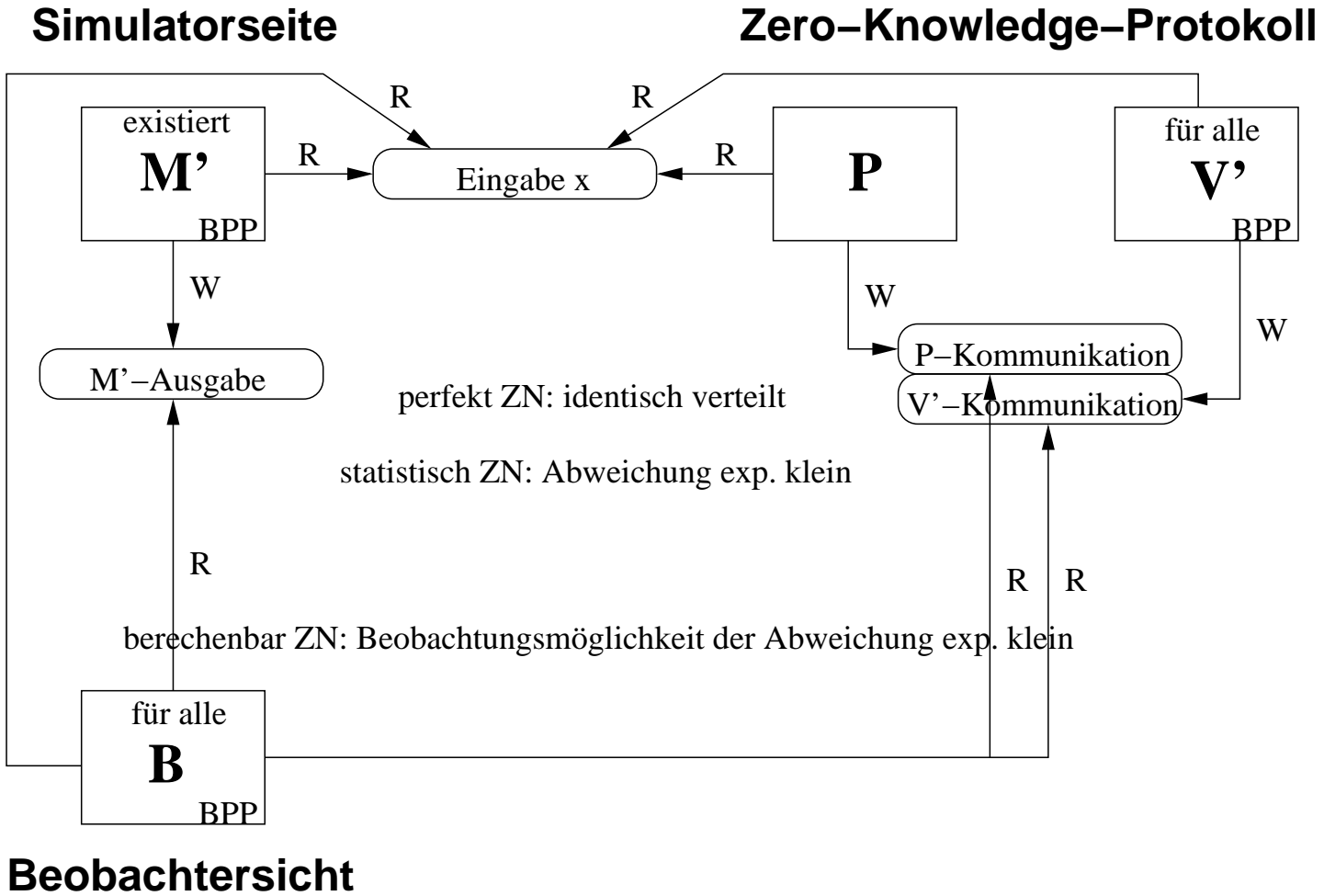
V : falls $\sigma = \tau$ akzeptiert V , ansonsten lehnt er den Beweis ab

Zero-Knowledge-Beweise: Formalisierung durch ITM-Modell

Definition 5 (Goldwasser, Micali, Rackoff 1985)

Ein interaktives Beweissystem (P, V) hat bezüglich der Sprache L die Zero-Knowledge-Eigenschaft, falls für alle Eingaben $x \in L$ der Verifizierer V nach seiner Interaktion mit P nicht mehr berechnen/wissen kann, als eine probabilistische polynomialzeitbeschränkte Turingmaschine bei Eingabe x auch alleine hätte erfahren können.

Insbesondere soll auch ein polynomialzeitbeschränkter Beobachter, welcher nur die Kommunikationsbänder von P und V belauscht, nicht erkennen können, ob der Beweis korrekt war. (vgl. Videoaufzeichnung mit Absprache) \rightsquigarrow Simulierbarkeit



Zero-Knowledge-Beweise: Klassifikation hinsichtlich P

Definition 6 (perfekte Zero-Knowledge-Eigenschaft)

P ist perfekt ZN bzgl. L , wenn für jeden probabilistischen polynomialzeitbeschränkten Verifizierer V' eine probabilistische polynomialzeitbeschränkte Turingmaschine M' existiert, welche die Kommunikation des interaktiven Protokolls (P, V') identisch simuliert, d. h.

$$\forall x \in L : C((P, V'), x) = C(M', x)$$

$C((P, V'), x)$ Zufallsvariable für Kommunikationsbänder von P, V' bei Eingabe von x

$C(M', x)$ Zufallsvariable für Ausgabeverhalten von M' bei Eingabe von x

(identische Wahrscheinlichkeitsverteilung)

Zero-Knowledge-Beweise: Klassifikation hinsichtlich P

Definition 7 (statistische Zero-Knowledge-Eigenschaft)

P ist statistisch ZN bzgl. L , wenn für jeden probabilistischen polynomialzeitbeschränkten Verifizierer V' eine probabilistische polynomialzeitbeschränkte Turingmaschine M' (Simulator) existiert, so daß

$$\forall x \in L : \sum_{\alpha} |\Pr[C((P, V'), x) = \alpha] - \Pr[C(M', x) = \alpha]| \leq |x|^{-c}$$

für alle Konstanten $c > 0$ und genügend langes x gilt.

$C((P, V'), x)$ Zufallsvariable für Kommunikationsbänder von P, V' bei Eingabe von x

$C(M', x)$ Zufallsvariable für Ausgabeverhalten von M' bei Eingabe von x

(Abweichung der Wahrscheinlichkeitsverteilungen ist exp. klein)

Zero-Knowledge-Beweise: Klassifikation hinsichtlich P

Definition 8 (berechenbare Zero-Knowledge-Eigenschaft)

P ist berechenbar ZN bzgl. L , wenn für jeden probabilistischen polynomialzeitbeschränkten Verifizierer V' eine probabilistische polynomialzeitbeschränkte Turingmaschine M' (Simulator) existiert, so daß für alle probabilistische polynomialzeitbeschränkte Turingmaschinen B (Beobachter)

$$\forall x \in L : |p_B((P, V'), x) - p_B(M', x)| \leq |x|^{-c}$$

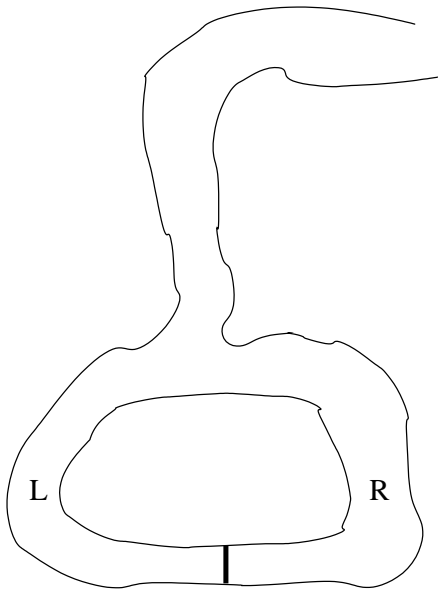
für alle Konstanten $c > 0$ und genügend langes x gilt.

$$p_B((P, V'), x) = \sum_{\alpha} \Pr[B \text{ akzeptiert } x \text{ unter Eingabe } (x, \alpha)] \cdot \Pr[C((P, V'), x) = \alpha]$$

$$p_B(M', x) = \sum_{\alpha} \Pr[B \text{ akzeptiert } x \text{ unter Eingabe } (x, \alpha)] \cdot \Pr[C(M', x) = \alpha]$$

(Beobachtungsmöglichkeit der stat. Abweichung ist für B exp. klein)

Zero-Knowledge-Beweise: Ali Baba's Höhle (1)



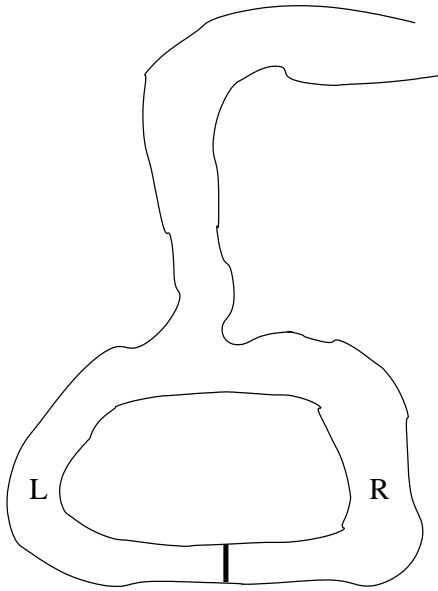
Peggy kann **magische Tür** öffnen und will es Viktor beweisen, ohne ihr Geheimnis zu verraten.

Triviales interaktives Protokoll:

1. P und V gehen zusammen in die Höhle.
2. P geht in die linke Hälfte (L),
3. und kommt rechts (R) wieder raus.

Keine Zero-Knowledge-Eigenschaft, weil kein Simulator konstruierbar!

Zero-Knowledge-Beweise: Ali Baba's Höhle (2)



perfekte Zero-Knowledge-Eigenschaft:

1. P geht in die Höhle, V bleibt draußen.
2. P geht zufällig nach links (L) oder rechts (R).
3. V kommt in den Vorraum der Höhle und ruft P zufällig L bzw. R zu.
4. P öffnet ggf. die Tür und kommt auf der gewünschten Seite heraus.

Betrugswahrscheinlichkeit: $1/2$, nach t -mal $\leq 2^{-t}$
Simulator gibt Paare (j, j) mit $j \in \{L, R\}$ aus.

Zero-Knowledge-Beweise: Graphen Isomorphie in perfektem ZN

Eingabe: Graphen G_1, G_2 , **P -Behauptung:** $G_1 \cong G_2$, **Geheimnis:** π mit $G_1 = \pi(G_2)$

P : $\sigma \stackrel{R}{:=} \{1, 2\}$, zufällige Permutation ψ der Knoten, berechnet $H := \psi(G_\sigma)$

sendet H an V

V : $\tau \stackrel{R}{:=} \{1, 2\}$ und fordert P auf, eine Permutation ρ anzugeben mit $H = \rho(G_\tau)$

sendet τ an P

P :

$$\text{berechnet } \rho := \begin{cases} \psi & \text{falls } \tau = \sigma \\ \psi \circ \pi & \text{falls } \tau \neq \sigma \text{ und } \sigma = 1 \\ \psi \circ \pi^{-1} & \text{falls } \tau \neq \sigma \text{ und } \sigma = 2 \end{cases}$$

sendet ρ an V

V : prüft, ob $H = \rho(G_\tau)$ gilt

Zero-Knowledge-Beweise: Fakten über interaktive ZN-Beweise

Theorem 2 (Goldreich, Micali, Wigderson 1986)

Falls Einwegfunktionen und damit polynomialzeit-ununterscheidbare Verschlüsselungs- bzw. Festlegeschemata existieren, hat jede Sprache aus \mathcal{NP} auch ein interaktives Beweissystem mit berechenbarer Zero-Knowledge-Eigenschaft; d. h. alles aus \mathcal{NP} ist eigentlich auch ohne beobachtbaren Wissenstransfer mit ZN-Protokollen beweisbar.

Beweisidee: Konstruktion eines interaktiven Beweissystems mit berechenbarer Zero-Knowledge-Eigenschaft für \mathcal{NP} -vollständiges Problem der Graphen Dreifärbung.

Zero-Knowledge-Beweise: Identitätsbeweis nach Feige, Fiat, Shamir

Trent erzeugt für Peggy $n := p \cdot q$ ($33 := 3 \cdot 11$) mit großen Primzahlen p, q und wählt $v \in \mathbb{Z}_n^*$ als quadratischer Rest modulo n (d. h. $\exists x : x^2 \equiv v \pmod{n}$) zufällig, so daß $v^{-1} \pmod{n}$ (16) existiert. Öffentliche Information: v (31), Peggys Geheimnis: kleinste Quadratwurzel $s := \sqrt{v^{-1}} \pmod{n}$ ($4, 7, 26, 29$)

P : Wählt $r \in \mathbb{Z}_n^*$ (13) zufällig und sendet $x := r^2 \pmod{n}$ (4) an V .

V : Sendet P ein zufälliges Bit $b \in \{0, 1\}$.

P : Falls $b = 0$, sendet sie r (13), sonst $y := r \cdot s \pmod{n}$ (19) an V .

V : Falls $b = 0$, verifiziert er $x \equiv r^2 \pmod{n}$ (4), sonst $x \equiv y^2 \cdot v \pmod{n}$ (4).

Kryptographische Grundlagen fairer mentaler Kartenspiele

mentales Kartenspiel: „Kartenspiel ohne Spielkarten“

Mentale Spiele sind ohne jegliches Spielmaterial durchführbar!

Fairness: kein Mitspieler kann „betrügen“

(höchstens mit vernachlässigbarer Wahrscheinlichkeit)

Kryptographie: Wissenschaft der Verschlüsselung (antiquiert)

moderne Aufgaben: Authentizität, Anonymität, Fairness, Geheimhaltung

Kryptographie und mentale Spiele — ein kurzer Rückblick

[Blum81] M. Blum: *Coin Flipping by Telephone*, SIGACT News, 1981

[Crép87] C. Crépeau: *A Zero-Knowledge Poker Protocol that Achieves Confidentiality of Players Strategy*, Proceedings CRYPTO'86, pp. 239-247, 1987

[KuKaOg97] K. Kurosawa, Y. Katayama, W. Ogata: *Reshufflable and Laziness Tolerant Mental Card Game Protocol*, IEICE Trans., Vol. E00-A, 1997

[Schin98] C. Schindelhauer: *A Toolbox for Mental Card Games*, Technischer Report, Universität Lübeck, 1998

Begriffe/Symbole aus Zahlentheorie und Algebra

$\mathbb{Z}_n := \{0, 1, \dots, n - 1\}$ (Menge der [positiven] ganzen Zahlen kleiner n)

$$\mathbb{Z}_{33} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, \dots, 31, 32\}$$

$\mathbb{Z}_n^* := \{a \in \mathbb{Z}_n \mid \text{ggT}(a, n) = 1\}$ (alle zu n teilerfremde Zahlen)

$$\mathbb{Z}_{33}^* = \{1, 2, 4, 5, 7, 8, 10, 13, 14, 16, 17, 19, 20, 23, 25, 26, 28, 29, 31, 32\}$$

$\text{QR}_n := \{a \in \mathbb{Z}_n^* \mid \exists x \in \mathbb{Z}_n : x^2 \equiv a \pmod{n}\}$ (quadratische Reste)

$$\text{QR}_{33} = \{1, 4, 16, 25, 31\}$$

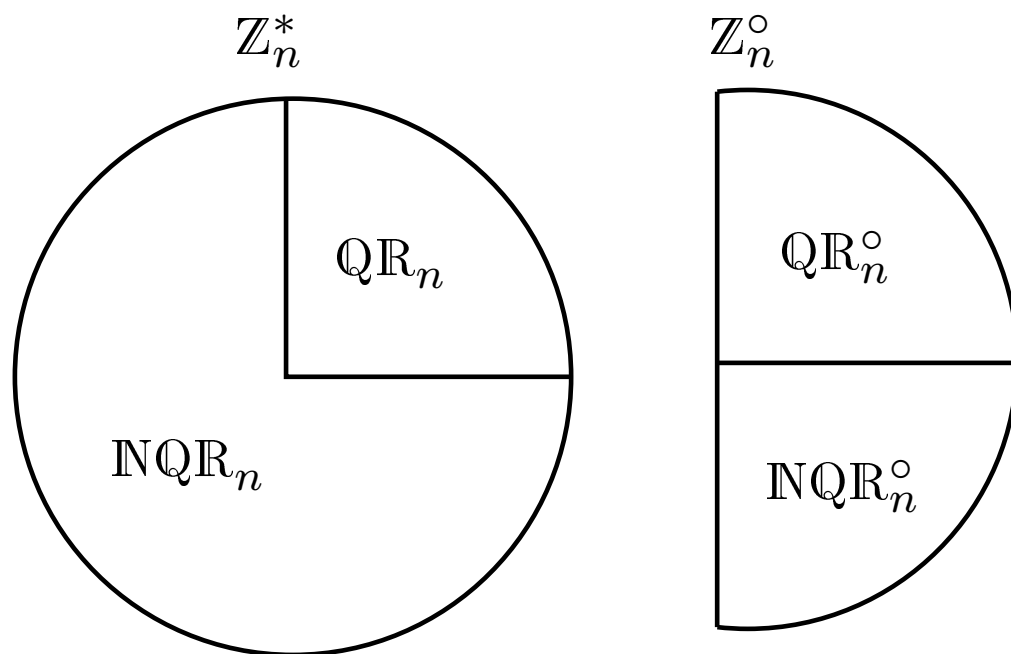
$\text{NQR}_n := \mathbb{Z}_n^* \setminus \text{QR}_n$ (quadratische nicht-Reste)

$$\text{NQR}_{33} = \{2, 5, 7, 8, 10, 13, 14, 17, 19, 20, 23, 26, 28, 29, 32\}$$

$\mathbb{Z}_n^\circ := \{a \in \mathbb{Z}_n^* \mid \left(\frac{a}{n}\right) = 1\}$, $\text{QR}_n^\circ := \text{QR}_n$, $\text{NQR}_n^\circ := \mathbb{Z}_n^\circ \cap \text{NQR}_n$

$$\mathbb{Z}_{33}^\circ = \{1, 2, 4, 8, 16, 17, 25, 29, 31, 32\}, \text{QR}_{33}^\circ = \{1, 4, 16, 25, 31\}, \text{NQR}_{33}^\circ = \{2, 8, 17, 29, 32\}$$

Begriffe/Symbole aus Zahlentheorie und Algebra



$$n = p \cdot q \text{ mit } p, q \in \mathbb{P} \\ p \neq q \text{ und } p, q \neq 2$$

$$|QR_n^\circ| = |NQR_n^\circ| = \\ = \frac{1}{2}|Z_n^\circ| = \frac{1}{4}|Z_n^*|$$

$$|Z_n^*| = \varphi(n) = \\ = (p - 1)(q - 1)$$

$$\forall x, y \in Z_n^* \text{ (mul. Gruppe)} : \begin{array}{ll} x \in QR_n, y \in QR_n & \Rightarrow xy \in QR_n \\ x \in QR_n, y \in NQR_n & \Rightarrow xy \in NQR_n \\ x \in NQR_n^\circ, y \in NQR_n^\circ & \Rightarrow xy \in QR_n^\circ \end{array}$$

Schwierige Probleme?

FAKTOR: $n = p_1 \cdot p_2 \cdot \dots \cdot p_k$ (Zerlegung in Primfaktoren p_i)
z.Z. keine effizienten Algorithmen bekannt, aber kein Beweis

QWURZEL: zu gegebenen $a \in \mathbb{QR}_n$ finde x mit $x^2 \equiv a \pmod{n}$

$n \in \mathbb{P}$: effizientes Verfahren bekannt, Komplexität $O((\lg n)^4)$

$n \notin \mathbb{P}$: $\text{QWURZEL} \leq_{\mathcal{P}} \text{FAKTOR}$, $\text{FAKTOR} \leq_{\mathcal{P}} \text{QWURZEL}$

QREST: zu gegebenen $a \in \mathbb{Z}_n^\circ$ entscheide $\overset{?}{\in} \mathbb{QR}_n$, $\text{QREST} \leq_{\mathcal{P}} \text{FAKTOR}$

Anforderungen — Werkzeugkasten für mentale Kartenspiele

- beliebig viele Mitspieler und Karten(-typen) müssen möglich sein
- Bereitstellung verschiedener Karten- und Stapeloperationen
- ein Spieler kann den Typ einer Karte nur bestimmen, falls alle anderen Mitspieler einverstanden sind
- keine Koalition von Spielern kann den Typ privater Karten gegen den Willen des Inhabers bestimmen (außer trivialen Schlußfolgerungen)
- Spielstrategie soll geheim bleiben (Zero-Knowledge-Beweise)

Das Geheimnis steckt im Schlüssel

Spielvorbereitung, Schlüsselerzeugung:

- jeder Mitspieler i wählt zufällig $p_i, q_i \in \mathbb{P}_{512\text{Bit}}$ und $y_i \in \text{NQR}_{n_i}^\circ$
- $n_i := p_i \cdot q_i$ und y_i werden veröffentlicht, p_i, q_i bleiben geheim
- Teilnehmer i zeigt allen, daß n_i und y_i korrekt gewählt sind
Zero-Knowledge-Beweise: n_i hat genau zwei Primfaktoren, $y_i \in \text{NQR}_{n_i}^\circ$

Aufbau der Spielkarten

Kodierung der Spielkarten (w Stück, k Mitspieler):

- Zahlen $z_{i,j} \in \mathbb{Z}_{n_i}^\circ$ mit $i = 1, \dots, k$ und $j = 1, \dots, \lceil \log_2 w \rceil$
- Typ ist binär kodiert: $b_j \in \{0, 1\}$ mit $j = 1, \dots, \lceil \log_2 w \rceil$
- Verteilung der Typinformation ($\lceil \log_2 w \rceil$ Bit) auf **alle** Mitspieler

$$b_j = \bigoplus_{i=1}^k a_{i,j} \quad a_{i,j} := \begin{cases} 0 & z_{i,j} \in \mathbb{QR}_{n_i}^\circ \\ 1 & z_{i,j} \in \mathbb{NQR}_{n_i}^\circ \end{cases}$$

Aufbau der Spielkarten (am Beispiel)



\Leftrightarrow

Binärkodierung

$$b = 00010$$

\Leftrightarrow

Verteilung Typinformation
(Geheimnisteilung)

$$a_1 = 11001$$

\oplus

$$a_2 = 01110$$

\oplus

$$a_3 = 10101$$

$=$

$$b = 00010$$

Werkzeugkasten: Hammer, Bohrer, Meisel

Erzeugung einer offenen Karte:

1. $\forall i$ öffentlich bekannt: $1 \in \mathbb{QR}_{n_i}^\circ$ und $y_i \in \mathbb{NQR}_{n_i}^\circ$ (Schlüssel)
2. Typ festlegen; binäre Kodierung berechnen, z. B. $b = 00010$
3. $(k, \lceil \log_2 w \rceil)$ -Tupel der $z_{i,j} \in \mathbb{Z}_{n_i}^\circ$ bilden,
z. B. $z_{i,j} = ((1, 1, 1, y_1, 1), (1, 1, 1, 1, 1), (1, 1, 1, 1, 1))$ $k = 3, \lceil \log_2 w \rceil = 5$

Werkzeugkasten: Säge, Feile, Raspel

Maskieren einer offenen oder bereits maskierten Karte:

$$z'_{i,j} = z_{i,j} \cdot r_{i,j}^2 \cdot y_i^{c_{i,j}} \pmod{n_i} \text{ mit zufälligen } r_{i,j} \in \mathbb{Z}_{n_i}^\circ, c_{i,j} \in \{0, 1\}$$

- Typerhaltung durch Festlegung einer Zeile: $c_{1,j} = \bigoplus_{i=2}^k c_{i,j}$
- „Maskieren“ ist eine Äquivalenzrelation (reflexiv, symmetrisch, transitiv)
- Zero-Knowledge-Beweis zeigt Korrektheit der „Maskierung“

Öffnen/Zeigen einer maskierten Karte für Spieler s :

1. alle Mitspieler $i = 1, \dots, k$ bestimmen, ob $z_{i,j} \stackrel{?}{\in} \mathbb{QR}_{n_i}^\circ$ für $j = 1, \dots$
2. alle Mitspieler ($\neq s$) senden ihre Ergebnisse a_i an s
3. alle Mitspieler ($\neq s$) beweisen die Korrektheit ihrer Angaben
4. Spieler s berechnet den binären Kartentyp durch $b = \bigoplus_{i=1}^k a_i$

kurzer Einschub: Zero-Knowledge-Beweis für $x \in \mathbb{QR}_n^\circ$

[Schin98]

Peggy	Viktor
geg. $p, q \in \mathbb{P}, n = p \cdot q, x \in \mathbb{QR}_n^\circ$	$n \in \mathbb{N}, x \in \mathbb{Z}_n^\circ$
1.	$\xleftarrow{\gamma}$ wählt Sicherheitsparameter $\gamma \in \mathbb{N}$
2. wählt zufällig $r_1, \dots, r_\gamma \in \mathbb{Z}_n^*$ berechnet $s_1, \dots, s_\gamma \in \mathbb{Z}_n^*$ mit $\forall 1 \leq i \leq \gamma : x \equiv r_i^2 \cdot s_i^2 \pmod{n}$ durch $s_i := \sqrt{x \cdot r_i^{-2}} \pmod{n}$ $R_i := r_i^2 \pmod{n}, S_i := s_i^2 \pmod{n}$	$\xrightarrow{R_i, S_i}$
3.	$\forall 1 \leq i \leq \gamma : x \stackrel{?}{\equiv} R_i \cdot S_i \pmod{n}$ \xleftarrow{Y} wählt zufällig $Y \subseteq \{1, \dots, \gamma\}$
4. $\forall 1 \leq i \leq \gamma :$	$\forall 1 \leq i \leq \gamma :$
4a. falls $i \in Y$ sende nur r_i	$\xrightarrow{r_i}$ prüfe $R_i \stackrel{?}{=} r_i^2 \pmod{n}$
4b. falls $i \notin Y$ sende nur s_i	$\xrightarrow{s_i}$ prüfe $S_i \stackrel{?}{=} s_i^2 \pmod{n}$

Werkzeugkasten: Glühbirnen, Steckdosen

Mischen eines Kartenstapels S :

Sequenz durch alle Mitspieler $i = 1, \dots, k$:

1. Spieler i maskiert alle Karten des Stapels (geheim, neu in S')
2. Spieler i permutiert den Stapel S' (geheim)
3. Spieler i sendet allen Mitspielern den neuen Stapel S'
4. Spieler i beweist allen Mitspielern die Korrektheit $S \simeq S'$

Werkzeugkasten: „Nützliche Dinge“

zeitnahe Spielregelprüfung $S \cap U \neq \emptyset \Rightarrow C \in U$:

- Falls der Spieler im privaten Stapel S Karten einer bestimmten Typmenge U hat, wird garantiert, daß die gespielte Karte C (ggf. maskiert) auch in U ist.

weitere Operationen für Kartenstapel:

- „Abheben“: zyklische Permutation
- Einfügen einer verdeckten Karte in einen Stapel

„Hackerethik“: **Misstrau**e **Autorität** – **fördere Dezentralisierung!**

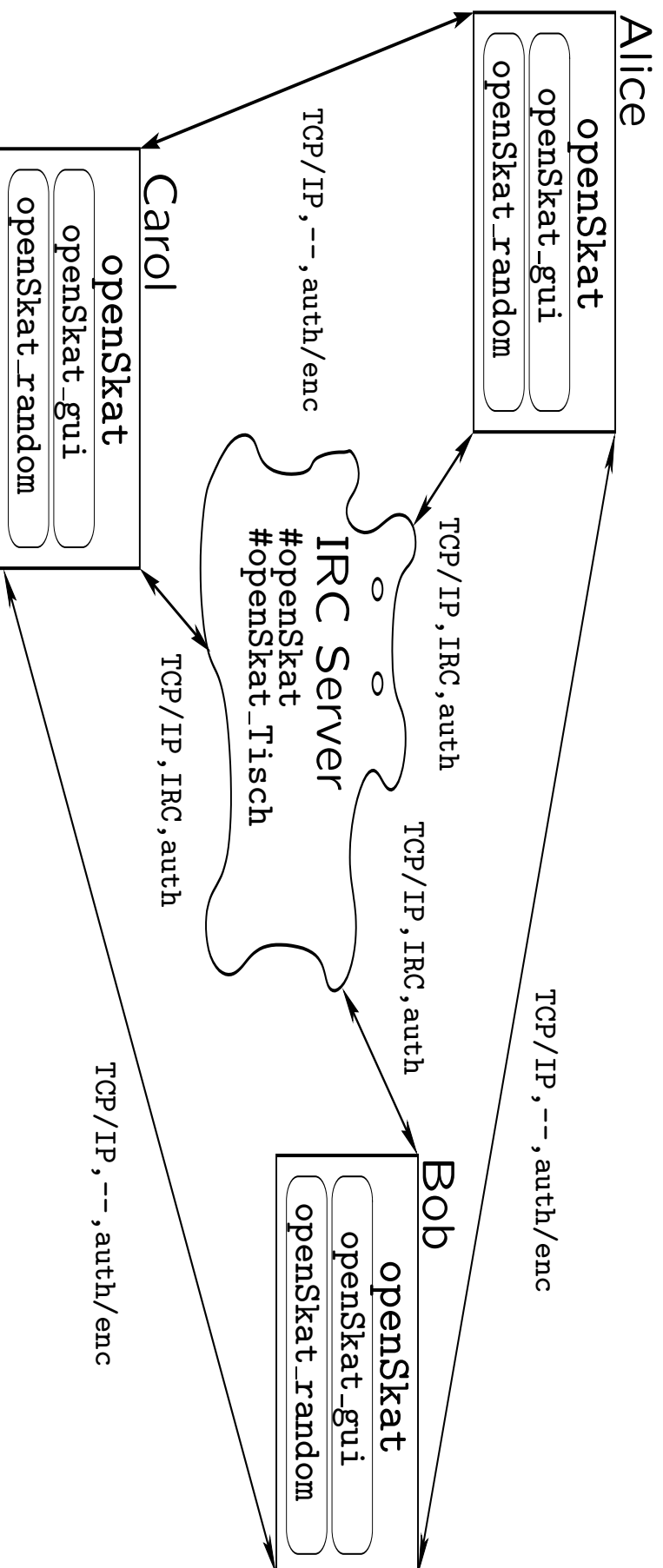
Fairness: minimaler Einfluß von Koalitionen durch TMCG

Sicherheit: nur wenige, sichere kryptographische Annahmen

Dezentralisierung: keine zentrale Vertrauensinstanz notwendig

Verifizierbarkeit: durch „Freie Software“ und offene Quelltexte

openskat — prototypischer Aufbau



openSkat — genutzte Bibliotheken („Freie Software“)

- GNU `libgmp` \geq 4.1: Rechnen mit beliebig langen/genauen Zahlen
- GNU `libgcrypt` \geq 1.1.10: kryptographische Grundfunktionen
 - Verschlüsselungsalgorithmus: *Blowfish*, symmetrisch, CFB-Modus, 128 Bit
 - Hashfunktion (message digest): *RIPEMD160*, 160 Bit
 - Zufallszahlenerzeugung (RandPool, EGD, `/dev/random`)
- `xskat` \geq 3.4: für graphische Benutzerschnittstelle `openSkat_gui`

openSkat — *Implementierung, Weiterentwicklung, Vorführung*

- Programmiersprache: C++, C [15 000 Zeilen] (mittlerweile Version 1.2 :-)
- Mit- oder Weiterentwickler (z. B. für openSkat_ai) gern gesehen:

```
cvcs -d :pserver:anonymous@gaos.org:/var/cvs login
```

```
cvcs -d :pserver:anonymous@gaos.org:/var/cvs co openSkat
```

Danke für die Geduld: Nun wird endlich vorgeführt!

openSkat — *Wieviel kostet „Fairness“?*

Sicherheitsparameter t	Betrugswahrscheinlichkeit $p \leq 2^{-t}$	übertragene Daten pro Spiel und Spieler
0	≤ 1	$\approx 0,7$ MB
1	$\leq 0,5$	$\approx 1,9$ MB
2	$\leq 0,25$	$\approx 3,0$ MB
4	$\leq 0,0625$	$\approx 5,0$ MB
8	$\leq 0,00390625$	≈ 10 MB
16	$\leq 0,00001526$	≈ 20 MB

Literatur, Quellen

[Schin98] C. Schindelhauer: *A Toolbox for Mental Card Games*

[GeMiRa97] R. Gennaro, D. Micciancio, T. Rabin: *An Efficient Non-Interactive Zero-Knowledge Proof System for Quasi-Safe Prime Products*

[MeOoVan96] A. Menezes, P. van Oorschot, S. Vanstone: *Handbook of Applied Cryptography*, CRC Press, 1996

GNU libgmp: <http://swox.com/gmp/>

GNU libgcrypt: <ftp://ftp.gnupg.org/gcrypt/alpha/libgcrypt/>

Gunter Gerhardt xskat: <http://www.gulu.net/xskat/>

[Zero-Knowledge] <http://www.tcs.hut.fi/~helger/crypto/link/zeroknowledge/>