



Robuste MIX-Netze:

Randomized Partial Checking,
Neffs verifizierbarer ElGamal-Shuffle

Referent:

Philippe Stellwag

philippe@stellwag.eu

[Gliederung]

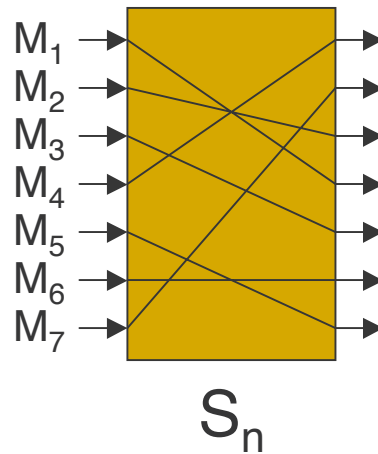
- i. MIXe
 - a. Definition
 - b. Grundfunktionen
 - c. Beispiel
 - d. Ansprüche an MIX-Netze
 - e. Anwendungen
- ii. Randomized Partial Checking
- iii. Neffs verifizierbarer ElGamal-Shuffle
- iv. Zusammenfassung



Allgemeines zu MIX-Netzen

[MIXe: Definition]

MIXe verstehen die Vermittlungsstationen S_n eines Anonymisierungskonzepts von Nachrichten M_n zur Kommunikation innerhalb von unsicheren Netzwerken (D. Chaum, 1981).

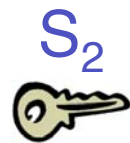
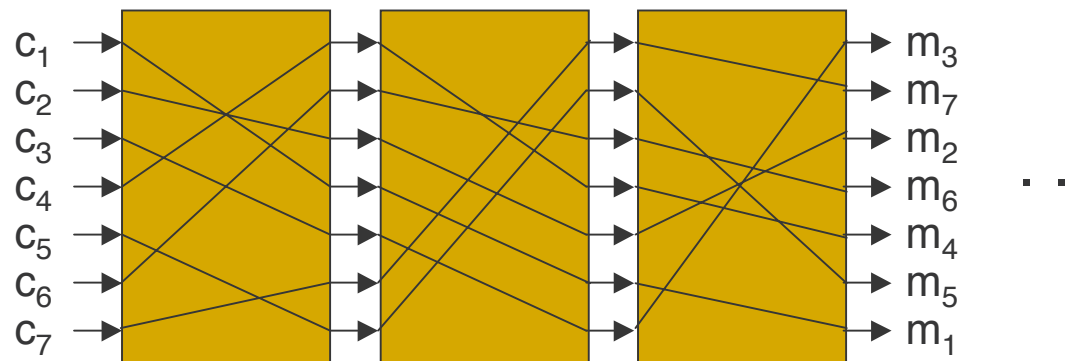


[MIXe: Grundfunktionen]

1. Sammlung/ Anhäufung von Nachrichten
2. Probabilistische Verschlüsselung (z. B. ElGamal)
 - Verhinderung von Duplikaten
 - Verhinderung von Klartext
3. Permutation der Nachrichten

[MIXe: Beispiel]

MIX-Netz (nach D. Chaum, 1981):



$$C_n = E_{pk1}(E_{pk2}(E_{pk3}(m_n)))$$

$$C_n = E_{pk2}(E_{pk3}(m_n))$$

$$C_n = E_{pk3}(m_n)$$

MIXe: Ansprüche an MIX-Netze

1. Sicherstellung der Geheimhaltung:
Ein Beobachter darf keine Beziehung zwischen Input und Output (und umgekehrt) ermitteln können (außer durch ordinäres Raten).
2. Korrekte Arbeitsweise:
Der Output muss eine zufällige, geheime Permutation des Inputs sein.
3. Robustheit:
Lieferung eines Beweises (oder eines überzeugenden Hinweises) für die korrekte Arbeitsweise. Wünschenswert wäre eine öffentliche Überprüfbarkeit!

[MIXe: Anwendungen]

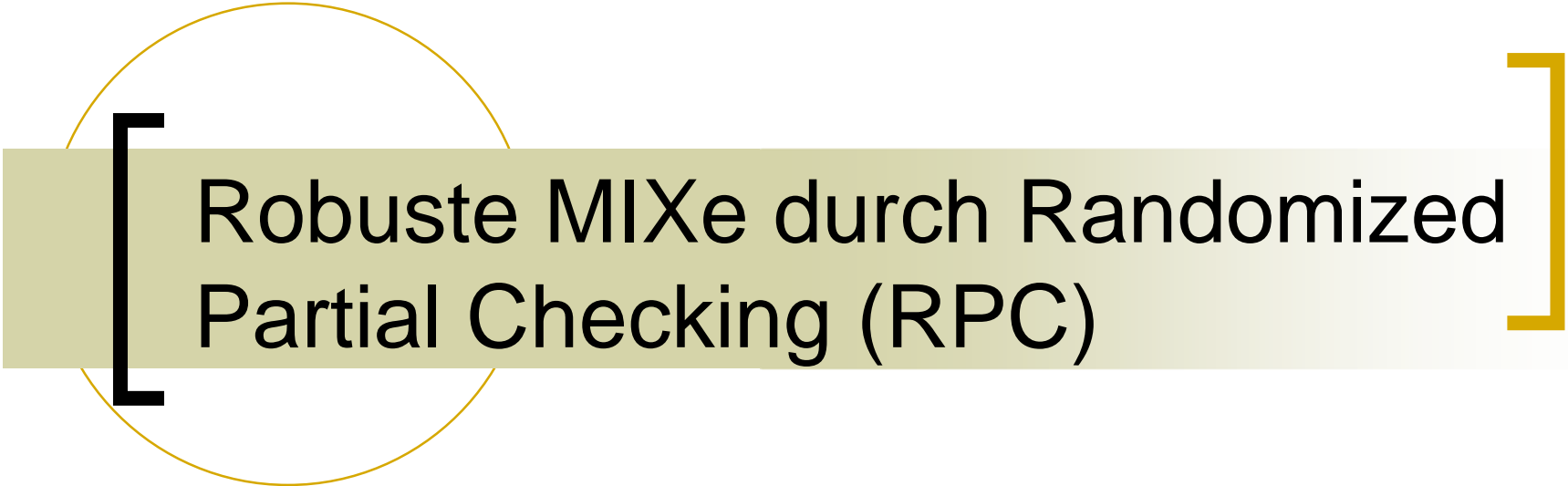
- Elektronische Wahlen
- Anonymisierungsdienste im WWW
(z. B. JAP¹ oder TOR²)

Probleme von „Echtzeitsystemen“:

- Sammlung/ Anhäufung von Daten
- Löschung von Duplikaten (Replay-Erkennung)

1) JAP: <http://anon.inf.tu-dresden.de/>

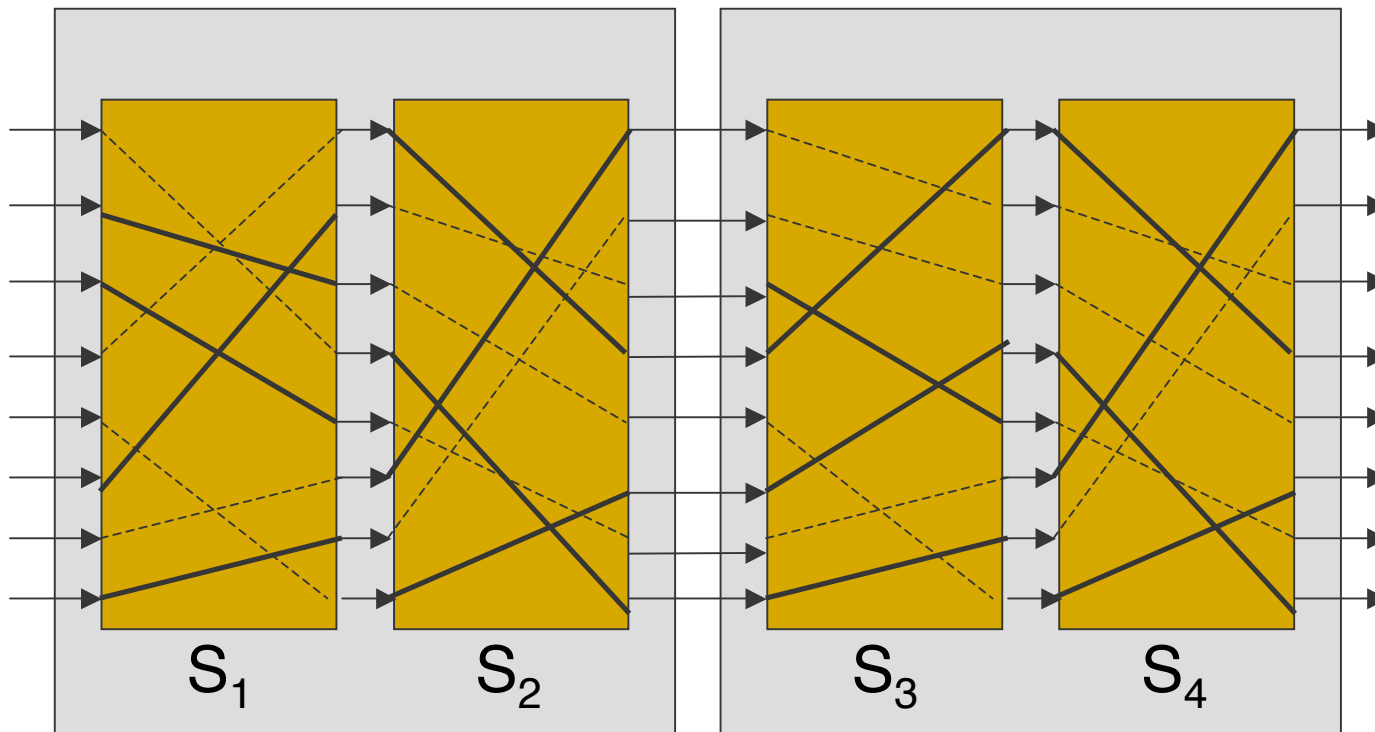
2) TOR: <http://tor.eff.org/>



Robuste MIXe durch Randomized Partial Checking (RPC)

- a. Partielle Offenlegung
- b. Bulletin Board
- c. Commitment zu einer Permutation
- d. Praktische Anwendungsmöglichkeit
- e. Wahlschema eines RPC-MIX-Netzes
- f. Boundary Probability

[Partielle Offenlegung]



Offenlegung von $n/2$ Nachrichten!

[Bulletin Board]

- Öffentliches schwarzes Brett
- Enthält verschlüsselten In- und Output der MIX-Server und am Schluss die entschlüsselte Nachricht (z. B. Wählerstimme).
- Öffentliche Nachrichten werden im „append-only“-Modus signiert angehängt.

[Commitment zu einer Permutation]

- MIX-Server ergänzen die Output-Liste mit einem Commitment zur privaten Permutation π_j
- Sei $\zeta_w[i]$ das Commitment zur Zahl i unter dem Zeugen w .
- Je nach Rolle gibt es zwei Mglk.:
 - $\Gamma_j^{(\text{in})} = \{ \zeta_{wji} [\pi_j(i)] \}_{i=1}^n$
 - $\Gamma_j^{(\text{out})} = \{ \zeta_{wji} [\pi_j^{-1}(i)] \}_{i=1}^n$

Praktische Anwendungsmglk.

- ζ_w wird durch eine kryptografische Hash-Funktion h instanziiert, wie z. B. SHA-1
- Wählen eines zufälligen Bit-Streams w und diesen mit der als String ausgedrückten Integer-Zahl i konkatenieren: $\zeta_w[i] = h(w || i)$
- Hiding Property: Bei entsprechender Länge des zufällig gewählten Bitstreams w bleibt die Länge des Inputs i verborgen.

Wahlschema eines RPC-MIX-Netzes

1. System Setup
2. Ballot Preparation and Encryption
3. Initial Ballot Checking
4. Permutation Commitment
5. Mix Net Processing
6. Correctness Check
7. Ballot Decryption
8. Boundary Check

[Boundary Probability]

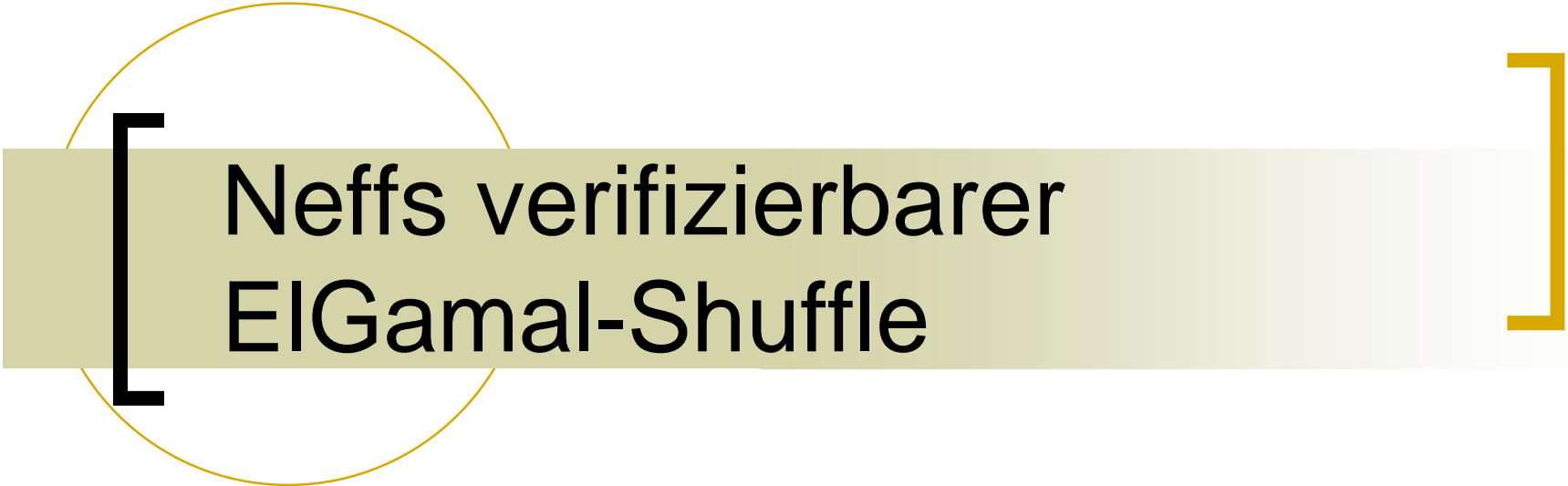
Elementare Frage: Wie groß ist die WSK, dass ein Angreifer durch Verfälschung von Stimmen das Wahlergebnis so beeinflusst hat, dass dieses nicht das korrekte ist?

■ Annahmen:

- Zentraler Angreifer
- Minderheit an Wählern und MIX-Servern betrügen (der Rest ist ehrlich)

■ Keine Anzeichen eines Betrugs:

- Behauptung: WSK, dass das Wahlergebnis k Stimmen abweicht ist $1/2^k$



Neffs verifizierbarer ElGamal-Shuffle

- a. Notationen
- b. Iterated Logarithmic Multiplication Proof Protocol (ILMPP)
- c. Simple k-Shuffle Proof Protocol
- d. Multi-Authority Voting Application

[Notationen]

- k = Anzahl der Stimmberechtigten
- P = prover (*shuffler*) und V = verifier (*auditor*)
- p, q sind Primzahlen, (wobei, $q \mid (p - 1)$), öffentlich bekannt
- Alle arithmetischen Operationen in \mathbb{Z}_p
- $G, X, H, Y \in \mathbb{Z}_p, g \in \mathbb{Z}_p$ mit primer multiplikativer Ordnung p
- Chaum-Pedersen:

„Beweis der Gleichheit diskreter Logarithmen“:

$$\log_G X = \log_H Y$$

- g ist fix, \otimes_g ist Binäroperator auf $\langle g \rangle \times \langle g \rangle$,

$$\log_g (x \otimes_g y) = \log_g x \log_g y$$

- $\bigotimes_{i=1}^k X_i = X_0 \otimes_g X_1 \otimes_g \cdots \otimes_g X_k$

- $\log_G X = \log_H Y \iff G \otimes_g Y = H \otimes_g X$

Iterated Logarithmic Multiplication Problem

- Basis g ist für alle logarithmischen Operationen fix
- $x_i = \log_g X_i$; $y_i = \log_g Y_i$
- Peggy Public Known Vic
 x_i $\{X_i\}_{i=1}^k$
 y_i $\{Y_i\}_{i=1}^k$
- Peggy muss Vic davon überzeugen, dass $\bigotimes_{i=1}^k X_i = \bigotimes_{i=1}^k Y_i$ ohne etwas über x_i und y_i zu verraten.
- Für den Fall $k=2$ entspricht dies genau dem Chaum-Pedersen-Beweis.

Iterated Logarithmic Multiplication Proof Protocol (ILMPP)

Peggy

Vic

1. Peggy generiert zufällig und unabhängig $k-1$ Elemente $\theta_1, \dots, \theta_{k-1}$ aus \mathbf{Z}_q und berechnet:

$$\begin{aligned} A_1 &= Y_1^{\theta_1} \\ A_2 &= X_2^{\theta_1} Y_2^{\theta_2} \\ &\vdots = \vdots \\ A_i &= X_i^{\theta_{i-1}} Y_i^{\theta_i} \\ &\vdots = \vdots \\ A_k &= X_k^{\theta_{k-1}} \end{aligned}$$

A_1, \dots, A_k \longrightarrow

2. \longleftarrow Vic generiert eine zufällige Challenge $\gamma \in \mathbf{Z}_q$

Iterated Logarithmic Multiplication Proof Protocol (ILMPP)

Peggy

Vic

3. Peggy berechnet $k-1$ Elemente, r_1, \dots, r_{k-1} in \mathbf{Z}_q :

$$\begin{aligned}
 Y_1^{r_1} &= A_1 X_1^{-\gamma} & (10) \\
 X_2^{r_1} Y_2^{r_2} &= A_2 \\
 \vdots &= \vdots \\
 X_i^{r_{i-1}} Y_i^{r_i} &= A_i \\
 \vdots &= \vdots \\
 X_k^{r_{k-1}} &= A_k Y_k^{(-1)^{(k-1)}\gamma}
 \end{aligned}$$

r_1, \dots, r_{k-1} \longrightarrow

4. Vic akzeptiert den Beweis nur, wenn die Gleichungen in (10) stimmen.

[Simple k-Shuffle Problem]

Peggy

$$x_i = \log_g X_i$$

$$y_i = \log_g Y_i$$

$$c, d \in \mathbf{Z}_p$$

Public Known

$$X_1, \dots, X_k \in \mathbf{Z}_p$$

$$Y_1, \dots, Y_k \in \mathbf{Z}_p$$

$$C = g^c \text{ und } D = g^d$$

Vic

- Peggy muss Vic davon überzeugen, dass es eine Permutation $\pi \in \Sigma_k$ mit der Eigenschaft $Y_i^d = X_{\pi(i)}^c$ für alle $1 \leq i \leq k$ gibt, ohne etwas über x_i , y_i , π , c oder d zu verraten.
- Anmerkung: $x_i \neq x_j$ für $i \neq j$ und $y_i \neq y_j$ für $i \neq j$ und $x_i \neq 1$ für alle $1 \leq i \leq k$.

Simple k-Shuffle Proof Protocol

Peggy

Vic

1. Vic generiert zufällig eine Challenge $t \in \mathbf{Z}_q$.
←
2. Peggy und Vic berechnen öffentlich $U=D^t=g^{dt}$,
 $W=C^t=g^{ct}$, $\vec{X} = (\hat{X}_1, \dots, \hat{X}_k) = (X_1/U, \dots, X_k/U)$ und
 $\vec{Y} = (\hat{Y}_1, \dots, \hat{Y}_k) = (Y_1/W, \dots, Y_k/W)$
3. Peggy und Vic führen das ILMPP der Länge $2k$ aus:

$$\Phi = (\vec{X}, \overbrace{C, C, \dots, C}^k)$$

$$\Psi = (\vec{Y}, \overbrace{D, D, \dots, D}^k)$$

Vic akzeptiert, wenn der ILMPP von Vic erfolgreich war.

Multi-Authority Voting Application

Stimme: ElGamal-Paar $(g^{\alpha_i}, h^{\alpha_i} m)$

m = verschlüsselte Stimme

α_i = geheim durch Wähler generiert

h = öffentlicher Parameter

Nach dem Ende der Wahl werden die Stimmen sequenziell durch unabhängige Authorities permutiert:

1. β_i wird zufällig, geheim und unabhängig gewählt
2. Jede Stimme $v_i = (g^{\alpha_i}, h^{\alpha_i} m)$ wird ersetzt durch $(g^{\alpha_i + \beta_i}, h^{\alpha_i + \beta_i} m)$; ein CP-Beweis wird veröffentlicht ohne ein Geheimnis preiszugeben.

[Multi-Authority Voting Application]

3. Die verschlüsselte Stimme m wird mit dem geheimen Exponenten c überlagert m^c und permutiert.
 4. Schritte 1-2 werden wiederholt.
 5. An diesem Punkt sind die anfänglichen Stimmen m_i mit der c -th Power verschlüsselt. Über $1/c$ -th Power kann wieder m_i ermittelt werden. c ist jedoch geheim. Der nächste Verifizierer wird mit g und $C=g^c$ über den CP-Beweis überzeugt.
- Danach werden die permutierten Stimmen mit dem anfänglichen Verfahren entschlüsselt und zur Auszählung in Tabellenform aufgestellt.



Zusammenfassung

[Zusammenfassung RPC]

- Offenlegung von $n/2$ Nachrichten
- Wahrscheinlichkeit beim Betrug von k manipulierten Stimmen entdeckt zu werden ist $1-2^{-k}$
- Wahlzettel können willkürliche Größe oder Inhalte haben, d.h. keine Schwierigkeit mit Briefwahl oder langen Wahlzetteln
- Keine Zero-Knowledge Beweise notwendig

[Zusammenfassung Neff]

- Solange Diffie-Hellman-Problem als unlösbar gilt (dies ist der Fall, solange das Diskrete-Logarithmus-Problem nicht lösbar ist), keine Möglichkeit Output mit Input in Relation zu setzen.
- Cheating Probability:
 - ILMPP: $1/q$
 - Simple k -Shuffle Proof Protocol: k^{-1}/q
- Typischer Wert für k : $\ll 2^{80}$
- Typischer Wert für q : $> 2^{159}$

Literaturverzeichnis

[Chau81] D. Chaum: „Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms“, University of California, Berkeley, Feb. 1981

[DeKn02] H. Delfs, H. Knebl: „Introduction to Cryptography“, 2002

[Jako02] M. Jakobsson, Ari Juels, R. L. Rivest: „Making Mix Nets Robust For Electronic Voting By Randomized Partial Checking“, Feb. 2002

[Neff01] C. A. Neff: „A Verifiable Secret Shuffle and its Application to E-Voting“, Aug. 2001

[Neff04] C. A. Neff: „Verifiable Mixing (Shuffling) of ElGamal Pairs“, April 2004



Vielen Dank für die
Aufmerksamkeit!!!