



# Das c-mix Verfahren

Merlin Koglin, Maik Graaf



Universität Hamburg

DER FORSCHUNG | DER LEHRE | DER BILDUNG

# Agenda

---

## 1. Motivation

- Chaumsche Mixe
- Probleme
- Idee

## 2. c-mix Verfahren

- Übersicht
- Precomputation Phase
- Realtime Phase

## 3. Analyse

- Sicherheit
- Performance

## 4. Anwendung

- Privatedegrety
- Kritik

## Probleme bisheriger Mix Verfahren

---

- Zweck
  - Nur mit **berechtigten Partnern** weiter kommunizieren
  - Verhindert unbefugte Inanspruchnahme von Betriebsmitteln

## Idee von David Chaum

---

- Zweck
  - Nur mit **berechtigten Partnern** weiter kommunizieren
  - Verhindert unbefugte Inanspruchnahme von Betriebsmitteln

# Kommunikationsübersicht

---

- Zweck
  - Nur mit **berechtigten Partnern** weiter kommunizieren
  - Verhindert unbefugte Inanspruchnahme von Betriebsmitteln

## Übersicht der zwei Phasen

---

- Zweck
  - Nur mit **berechtigten Partnern** weiter kommunizieren
  - Verhindert unbefugte Inanspruchnahme von Betriebsmitteln

## Vorbereitung

---

- Zweck
  - Nur mit **berechtigten Partnern** weiter kommunizieren
  - Verhindert unbefugte Inanspruchnahme von Betriebsmitteln

## Precomputation - Step 1

---

- Pre Processing
- Knoten  $N_1, \dots, N_n$  erzeugen einen zufälligen Vektor  $r_i$ 
  - Vektor enthält einen zufälligen Wert für jeden Nachrichtenslot
- Verschlüsselung mittels ElGamal  $\rightarrow E(r_i^{-1})$ , Resultat wird an den Network Handler gesendet
  - Diese Verschlüsselung muss dann in der Echtzeitphase nicht mehr durchgeführt werden
- NH berechnet Produkt aus allen  $E(r_n) \rightarrow E(R_n^{-1})$



## Precomputation - Step 2

---

- Mixing
  1. Jeder Knoten erzeugt einen weiteren zufälligen Vektor  $s_i$
  2.  $E(R_n^{-1})$  wird von jedem Knoten nacheinander mit der jeweils festgelegten Permutation permutiert (Mixing) und gleichzeitig der erzeugte  $s_i^{-1}$  hineinmultipliziert
  3. Der letzte Knoten erzeugt damit  $E(P_n(R_n^{-1}) \times S_n^{-1})$

## Precomputation - Step 3

---

- **Post Processing**
  1. Jeder Knoten berechnet nun aus  $E(P_n(R_n^{-1}) \times S_n^{-1})$  seinen Entschlüsselungsanteil  $D(i, r)$  für den zufälligen Vektor  $r_i$  aus Schritt 1.
  2. Da jeder Knoten einen eigenen Entschlüsselungsanteil berechnen kann, liegt an der ElGamal Verschlüsselung, die diese Möglichkeit bietet.

## Precomputation - Return Path

- Step 1
  1. Nodes erzeugen zufällige Vektoren  $E(s_i'^{-1})$  (ElGamal verschlüsselt).
  2. Permutation rückwärts, der letzte Knoten beginnt, gleichzeitig werden  $s'^{-1}$  dazumultipliziert
  3. Der erste Knoten erhält  $E(S_1'^{-1})$
- Step 2
  1. Wie vorher werden wieder Entschlüsselungsanteile für  $E(S_1'^{-1})$  von allen Knoten berechnet

## Echzeit Phase - Step 1

- Preprocessing

1. Ein User verschlüsselt seine Nachricht  $M$  mit seinem Schlüssel  $M \times ka_i^{-1}$  und sendet diese an den NH, dieser erhält also  $M \times Ka^{-1}$
2. Nun sendet jeder Knoten  $N_i$  seinen Wert  $ka_i \times r_i$  an den NH.
3. Der NH kann damit die  $Ka^{-1}$  mit den zufälligen Vektoren  $r_i$  der Knoten austauschen
4.  $M \times Ka^{-1} \times \sum_{i=1}^n ka_i \times r_i = M \times R_n$

## Echzeit Phase - Step 2

---

- Mixing
  - Jeder Knoten permutiert nacheinander  $M \times R_n$  und multipliziert den zufälligen Vektor  $S_i$  mit ein
  - Der letzte Knoten erhält damit  $P_n(M \times R_n) \times S_n$

## Echzeit Phase - Step 3

- Entschlüsselungsanteil
  - Die Knoten  $N_1$  bis  $N_i$  senden ihren Entschlüsselungsanteil  $D(i, x)$  an den NH
  - Entschlüsselung
    - Der NH Entschlüsselt  $E(P_n(R_n^{-1}) \times S_n^{-1})$  mittels  $D(n, x)$
    - $P_n(M \times R_n) \times S_n \times P_n(R_n^{-1}) \times S_n^{-1} = P_n(M)$

# Anonymität

---

- Zweck
  - Nur mit **berechtigten Partnern** weiter kommunizieren
  - Verhindert unbefugte Inanspruchnahme von Betriebsmitteln

# Integrität

---

- Zweck
  - Nur mit **berechtigten Partnern** weiter kommunizieren
  - Verhindert unbefugte Inanspruchnahme von Betriebsmitteln



# Vertraulichkeit

---

- Zweck
  - Nur mit berechtigten Partnern weiter kommunizieren
  - Verhindert unbefugte Inanspruchnahme von Betriebsmitteln

# Protoyp

---

- Zweck
  - Nur mit berechtigten Partnern weiter kommunizieren
  - Verhindert unbefugte Inanspruchnahme von Betriebsmitteln

## Einbettung in PrivaTegrity

---

- Zweck
  - Nur mit berechtigten Partnern weiter kommunizieren
  - Verhindert unbefugte Inanspruchnahme von Betriebsmitteln

# Backdoor

---

- Zweck
  - Nur mit berechtigten Partnern weiter kommunizieren
  - Verhindert unbefugte Inanspruchnahme von Betriebsmitteln

## Der Arbeitsbereich Sicherheit in Verteilten Systemen (SVS)

Lorem ipsum dolor

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

- Themen
  1. Privacy Enhancing Technologies (PET)
  2. Security Management & Risk Management
  3. Security of Mobile Systems
- Weitere Informationen
  - <http://www.informatik.uni-hamburg.de/svs>

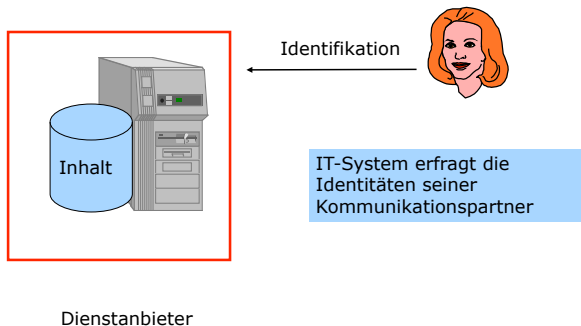
## Beispiel für eine Abbildung

---

- Zweck
  - Nur mit **berechtigten Partnern** weiter kommunizieren
  - Verhindert unbefugte Inanspruchnahme von Betriebsmitteln

## Beispiel für eine Abbildung

- Zweck
  - Nur mit **berechtigten Partnern** weiter kommunizieren
  - Verhindert unbefugte Inanspruchnahme von Betriebsmitteln



## Beispiel für eine Abbildung

- Zweck
  - Einem Kunden  $K$  einen Inhalt  $I$  in einer bestimmten Weise zugänglich machen, ihn aber daran hindern, *alles* damit tun zu können.

