Das c-mix Verfahren

Merlin Koglin, Maik Graaf

Agenda

1. Motivation
   - Chaumsche Mixe
   - Probleme
   - Idee
2. c-mix Verfahren
   - Übersicht
   - Precomputation Phase
   - Realtime Phase
3. Analyse
   - Sicherheit
   - Performance
4. Andwendung
   - Privategrety
   - Kritik

Ansatz der Chaumische Mixe

- Gewährleistung der Anonymität
  - Mixe werden durchlaufen um die Beziehung zwischen Sender und Empfänger zu verschleiern
  - Im Mixnetz findet eine „Zwiebelschalen“ artige Entschlüsselung statt
  - Mithilfe der Rückadresse wird der Empfänger einer Nachricht bestimmt

Probleme bisheriger Mix Verfahren

- In Echtzeitsystemen
  - Lange warte Zeiten beim Sammeln der Nachrichten
  - Der Sammelschritt wird deshalb kurz gehalten oder sogar weggelassen
  - Das Verfahren wird somit angreifbarer
  - Für mobile Geräte ungeeignet aufgrund des großen Zeit und Energie Aufwands

Idee von David Chaum

- Vermeidung von Schlüsselberechnungen in Echtzeit
  - Steigerung der Effizienz von Mix-Netzen
  - Energieaufwand verringern
  - Schlüsselberechnung vor der Kommunikation
  - Schlüssel Austausch zwischen Sender und Mixknoten

Kommunkationsübersicht

- 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, ... N_n erzeugen einen zufälligen Vektor r_i
  - Vektor enthält einen zufälligen Wert für jeden Nachrichtenslot
- Verschlüsselung mittes ElGamal $\rightarrow E(r_i^{-1})$, Resultat wird an den Network Handler gesendet
  - Diese Verschlüsselung muss dann in der Echzeitphase 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 gleichzeit 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. Das 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 $E(s'^{-1}_i)$ (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 Entschüsselungsanteile für $E(S'^{-1}_1)$ von allen Knoten berechnet

Echzeit Phase - Step 1

- Preprocessing
  1. Ein User verschüsselt seine Nachricht M mit seinem Schüssel $M \times ka_i^{-1}$ 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

- Anhand eines Modells
  - Private Kommunikation der Mixknoten untereinander und eines vertraulichen dritten Punktes
  - Keine Kryptographischen Operation, Sicherstellung durch den vertraulichen dritten Punkt
  - „reale Simulation“ des Modells mit Eigenschaften des cMix Protokolls zeigt Anonymität

Integrität

- Die Integrität ist gegeben wenn
  - Die Nachricht M unmodifiziert und an den Empfänger weitergeleitet wird oder...
  - Alle Mixknoten wissen, dass das cMix Protokoll nicht richtig durchgeführt wurde
    - Sicherstellung durch den Mechanismus „Randomized Partial Checking“

Vertraulichkeit

- Schutzziel Anonymität
  - Wird sichergestellt indem Nachrichten vom Sender verschlüsselt werden
    - z.B durch einen öffentlichen Schlüssel einer asymmetrischen Verschlüsselung
  - Diese Verschlüsselung vermeidet aufwändige Publickey-Operationen

Protoyp

- Performance Messung
  - In Python implementiert
  - Auf Instanzen des Amazon Web Service EC2 getestet
  - Jeder Mixknoten hatte zwei Intel Xeon E5-2680 und 3,75 GB Arbeitsspeicher zur Verfügung
  - Bei einer 1024-bit ElGamal-Verschlüsselung
  - Starke Verbesserung das re-encryption Mixnet ist bis zu 8 mal langsamer

Anzahl Nachrichten	Vorberechnung (Durchschnitt in Sekunden)	Echtzeit (Durchschnitt in Sekunden)
50	1.56	0.20
100	3.02	0.33
500	14.59	1.51
1000	28.87	3.09

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