

IT-Sicherheit Seminar

Reimer: Typ-I bis Typ-III

Mervyn McCreight

FH-Wedel

4. November 2015

Table of Contents

1 Motivation

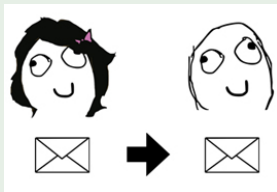
2 Cypherpunk-Remailer

- Funktionsweise
- Sicherheitsanalyse
- Zuverlässigkeitsanalyse

3 Mixmaster-Remailer

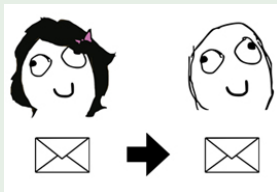
- Funktionsweise
- Analyse

Sitzung



- Alice möchte Bob Nachricht senden
- Normal: Schutz des Inhalts
- Jetzt: Schutz der Identitäten

Sitzung



- Alice möchte Bob Nachricht senden
- Normal: Schutz des Inhalts
- Jetzt: Schutz der Identitäten

Angreifer Eve möchte Ziele gefährden



- Netzwerk beobachten
- Einsicht in Traffic
- Pakete abfangen, senden, manipulieren und senden

Table of Contents

1 Motivation

2 Cypherpunk-Remailer

- Funktionsweise
- Sicherheitsanalyse
- Zuverlässigkeitsanalyse

3 Mixmaster-Remailer

- Funktionsweise
- Analyse

Wesentliche Eigenschaften

- Klassifizierung: Typ-I Remailer
- "Cipher", "Cyber", "Punk"
- Anonymisierend
- Inspiration: Mix-Netzwerke (*David Chaum*)
- E-Mail Protokoll

Wesentliche Eigenschaften

- Klassifizierung: Typ-I Remailer
- "Cipher", "Cyber", "Punk"
- Anonymisierend
- Inspiration: Mix-Netzwerke (*David Chaum*)
- E-Mail Protokoll

Wesentliche Eigenschaften

- Klassifizierung: Typ-I Remailer
- "Cipher", "Cyber", "Punk"
- Anonymisierend
- Inspiration: Mix-Netzwerke (*David Chaum*)
- E-Mail Protokoll

Wesentliche Eigenschaften

- Klassifizierung: Typ-I Remailer
- "Cipher", "Cyber", "Punk"
- Anonymisierend
- Inspiration: Mix-Netzwerke (*David Chaum*)
- E-Mail Protokoll

Wesentliche Eigenschaften

- Klassifizierung: Typ-I Remailer
- "Cipher", "Cyber", "Punk"
- Anonymisierend
- Inspiration: Mix-Netzwerke (*David Chaum*)
- E-Mail Protokoll

Basis des Protokolls

Netzwerk von mehreren verschiedenen Cypherpunk-Remailern

Cypherpunk-Remailer C



- öffentlicher Schlüssel D_C
- privater Schlüssel E_C
- Nachricht entschlüsseln und weiterleiten
- Nachrichten-Header modifizieren

Alice kennt:

- Remailer-Netzwerk C_1, C_2, \dots, C_n
- öffentliche Schlüssel $E_{C_1}, E_{C_2}, \dots, E_{C_n}$

Alice muss

- Auswahl Remailer
- Reihenfolge bestimmen

Ziel

Nachricht wird über Pfad an Bob gesendet

Inhalt einer Nachricht

- Adresse A
- Nachricht N

schichtenweise Verschlüsselung

$$N' = (A_1, E_{C_1}(A_2, E_{C_2}(\dots(A_n, E_{C_n}(A_{Bob}, E_{Bob}(N))))) \quad (1)$$

Beispiel



Ablauf Sendevorgang

- Alice sendet N' an C_1
- C_1 erhält A_2 und verschlüsselte Nachricht
- C_1 sendet Nachricht an Adresse in A_2
- C_2 erhält A_3 und verschlüsselte Nachricht
- (...)
- C_n sendet Nachricht an Adresse von Bob

Was haben wir erreicht?

- C_x kennt nur unmittelbaren Nachfolger und Vorgänger
- Bob kennt nur letzten Remailer
- Alice kennt als Einzige gesamten Pfad



Traffic Analyse

- Nachrichtengröße
- leitet Nachrichten sofort weiter

Replay Angriff

- Eve kann Nachrichten abfangen und wieder einspielen
- Duplikate werden nicht erkannt



Remailer im Pfad fällt aus

- Jeder Remailer kennt nur unmittelbaren Nachfolger und Vorgänger
- Nachricht verschwindet
- Fehler bleibt unbemerkt

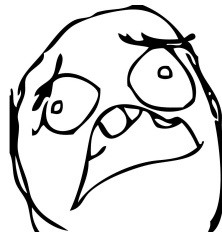


Table of Contents

1 Motivation

2 Cypherpunk-Remailer

- Funktionsweise
- Sicherheitsanalyse
- Zuverlässigkeitsanalyse

3 Mixmaster-Remailer

- Funktionsweise
- Analyse

- Baut auf Cytherpunk-Remailer auf
- Basis: "Mixmaster and remailer attacks"(Lance Cottrell)
 - deckt Sicherheitsschwächen auf
 - bietet Lösungsansätze

Ziel

Weiterentwicklung des existierenden Protokolls, um Sicherheitslücken zu schließen.

Was bleibt gleich?

- Remailernetzwerk
- schichtenweise Verschlüsselung
- E-Mail Protokoll

Was bleibt gleich?

- Remailernetzwerk
- schichtenweise Verschlüsselung
- E-Mail Protokoll

Ausblick: Was muss sich ändern?

- Traffic transparent gestalten
- Zeitliche Zuordnung
- Duplikate erkennen
- Manipulierte Nachrichten erkennen

Ziel

Traffic transparent gestalten

- Nachricht in gleich große Chunks aufteilen
- Auffüllen mit zufälligen Dummy-Daten
- Chunks statt Nachricht über Netzwerk verteilen
- Jeder Chunk (möglichst) auf verschiedenem Pfad
 - letzter Remailer im Pfad
 - Zusammensetzung
 - Weiterleitung

Ziel

Zeitliche Zuordnung

- Einkommende Nachrichten sammeln
- beliebige Größe
- zufällige Reihenfolge

Ziel

Zeitliche Zuordnung

- Einkommende Nachrichten sammeln
- beliebige Größe
- zufällige Reihenfolge

Pool wird nie voll?

- individueller Zeitpunkt
- zufallsgenerierte Dummy-Nachrichten

Ziel

Duplikate und Manipulation

Signatur

- Erkennung von manipulierten Nachrichten
- werden verworfen

Identifikation

- verschlüsselte ID
- Duplikate werden erkannt
- werden verworfen

- Alice benötigt Client-Software
- Nachricht wird aufgeteilt
- Chunks werden über das Netzwerk weitergeleitet
- letzter Remailer sammelt Chunks
- entfernt evtl. Dummy-Daten
- setzt Nachricht zusammen und leitet an Bob

Traffic Analyse

- einheitliche Nachrichtengröße
- keine zeitliche Zuordnung
- konstanter Traffic-Level

Replay Angriff

- Duplikate werden erkannt und ignoriert
- Manipulationen werden erkannt und ignoriert



Angriffsmöglichkeit

- Nachricht von Alice abfangen und zurückhalten
- Überflutung des Remailernetzwerks mit eigenen Nachrichten
- Keine anderen Nachrichten außer Eves mehr im Netzwerk
- Einspielen von Alice Nachricht
- Empfänger dieser Nachricht muss Bob sein

Angriffsmöglichkeit

- Nachricht von Alice abfangen und zurückhalten
- Überflutung des Remailernetzwerks mit eigenen Nachrichten
- Keine anderen Nachrichten außer Eves mehr im Netzwerk
- Einspielen von Alice Nachricht
- Empfänger dieser Nachricht muss Bob sein

Aber:

