

IT-Sicherheit Seminar

Reimer: Typ-I bis Typ-III

Mervyn McCreight

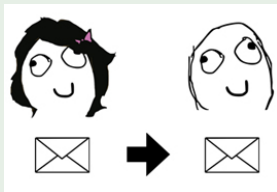
FH-Wedel

7. November 2015

Table of Contents

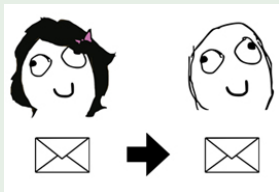
- 1 Motivation
- 2 Cypherpunk-Remailer
 - Funktionsweise
 - Sicherheitsanalyse
 - Zuverlässigkeitsanalyse
- 3 Mixmaster-Remailer
 - Funktionsweise
 - Analyse
- 4 Nym-Server
- 5 Mixminion-Remailer

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
- 4 Nym-Server
- 5 Mixminion-Remailer

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

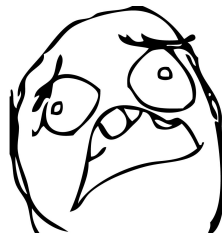


Table of Contents

- 1 Motivation
- 2 Cypherpunk-Remailer
 - Funktionsweise
 - Sicherheitsanalyse
 - Zuverlässigkeitsanalyse
- 3 Mixmaster-Remailer
 - Funktionsweise
 - Analyse
- 4 Nym-Server
- 5 Mixminion-Remailer

- Baut auf Cypherpunk-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:



Table of Contents

- 1 Motivation
- 2 Cypherpunk-Remailer
 - Funktionsweise
 - Sicherheitsanalyse
 - Zuverlässigkeitsanalyse
- 3 Mixmaster-Remailer
 - Funktionsweise
 - Analyse
- 4 Nym-Server
- 5 Mixminion-Remailer

Nym-Server - Motivation

- bisher - Anonymisierung
 - Identität ist verborgen
 - Bob kennt nur letzten Remailer
 - antworten nicht möglich
- nun - Pseudonymisierung
 - Identität versteckt hinter Decknamen
 - Bob kennt Decknamen
 - antworten möglich

Typ-0 Remailer?

Definition (Quelle - techopedia.com)

A nym server is a pseudonym server that furnishes an untraceable email address. The purpose of this server is to allow users to have usernames (pseudonyms) and send and receive messages without revealing their true identities. Even the nym server operators cannot trace a user's email address.

Nym-Server - Motivation

- bisher - Anonymisierung
 - Identität ist verborgen
 - Bob kennt nur letzten Remailer
 - antworten nicht möglich
- nun - Pseudonymisierung
 - Identität versteckt hinter Decknamen
 - Bob kennt Decknamen
 - antworten möglich

Typ-0 Remailer?

Definition (Quelle - techopedia.com)

A nym server is a pseudonym server that furnishes an untraceable email address. The purpose of this server is to allow users to have usernames (pseudonyms) and send and receive messages without revealing their true identities. Even the nym server operators cannot trace a user's email address.

Nym-Server - Motivation

- bisher - Anonymisierung
 - Identität ist verborgen
 - Bob kennt nur letzten Remailer
 - antworten nicht möglich
- nun - Pseudonymisierung
 - Identität versteckt hinter Decknamen
 - Bob kennt Decknamen
 - antworten möglich

Typ-0 Remailer?

Definition (Quelle - techopedia.com)

A nym server is a pseudonym server that furnishes an untraceable email address. The purpose of this server is to allow users to have usernames (pseudonyms) and send and receive messages without revealing their true identities. Even the nym server operators cannot trace a user's email address.

Nym-Server - Motivation

- bisher - Anonymisierung
 - Identität ist verborgen
 - Bob kennt nur letzten Remailer
 - antworten nicht möglich
- nun - Pseudonymisierung
 - Identität versteckt hinter Decknamen
 - Bob kennt Decknamen
 - antworten möglich

Typ-0 Remailer?

Definition (Quelle - techopedia.com)

A nym server is a pseudonym server that furnishes an untraceable email address. The purpose of this server is to allow users to have usernames (pseudonyms) and send and receive messages without revealing their true identities. Even the nym server operators cannot trace a user's email address.

Ziele

- Schicken und Empfangen über Pseudonym
- Betreiber transparent für Betreiber

Idee: Umsetzung über Cypherpunk-Remailer

Ziele

- Schicken und Empfangen über Pseudonym
- Betreiber transparent für Betreiber

Idee: Umsetzung über Cypherpunk-Remailer

Bestandteile eines Nym

- öffentlicher Schlüssel
- Reply-Block
- Pseudonym

Reply-Block

- Enthält Pfad über Cypherpunk-Remailer Netzwerk zum Empfänger hinter dem Nym.
- schichtenweise verschlüsselt

Bestandteile eines Nym

- öffentlicher Schlüssel
- Reply-Block
- Pseudonym

Reply-Block

- Enthält Pfad über Cypherpunk-Remailer Netzwerk zum Empfänger hinter dem Nym.
- schichtenweise verschlüsselt

Nym-Server - Ablauf (Erstellung)

- Bob möchte Pseudonym
 - sucht sich Menge an Cypherpunk-Remailern heraus
 - erstellt Reply-Block
 - denkt sich Pseudonym aus
 - stellt öffentlichen Schlüssel bereit
- schickt Nym an Nym-Server
- Achtung: Muss anonymisiert gesendet werden!
- (ggf. Validierung des Nyms vom Nym-Server)

- Alice
 - kennt Bobs Pseudonym
 - möchte Bob eine Nachricht zukommen lassen
 - schickt Nachricht an Bobs Pseudonym an den Nym-Server
- Nym-Server
 - findet den Nym
 - verschlüsselt Nachricht mit öffentlichem Schlüssel
 - schickt Nachricht mit Reply-Block an ersten Remailer im Block
- Nachricht von Alice wird über Remailernetzwerk an Bob geschickt

- Alice
 - kennt Bobs Pseudonym
 - möchte Bob eine Nachricht zukommen lassen
 - schickt Nachricht an Bobs Pseudonym an den Nym-Server
- Nym-Server
 - findet den Nym
 - verschlüsselt Nachricht mit öffentlichem Schlüssel
 - schickt Nachricht mit Reply-Block an ersten Remailer im Block
- Nachricht von Alice wird über Remailernetzwerk an Bob geschickt

Sicherheit?

Table of Contents

- 1 Motivation
- 2 Cypherpunk-Remailer
 - Funktionsweise
 - Sicherheitsanalyse
 - Zuverlässigkeitsanalyse
- 3 Mixmaster-Remailer
 - Funktionsweise
 - Analyse
- 4 Nym-Server
- 5 Mixminion-Remailer

- Aktualität
- Nym-Server unsicher
- Zuverlässigkeit

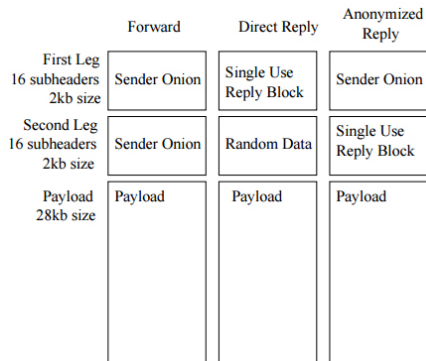
Was bleibt gleich?

- Netzwerkstruktur
- Anonymisierung
- Traffictransparenz
- schichtenweise Verschlüsselung (Header)

Was ist neu?

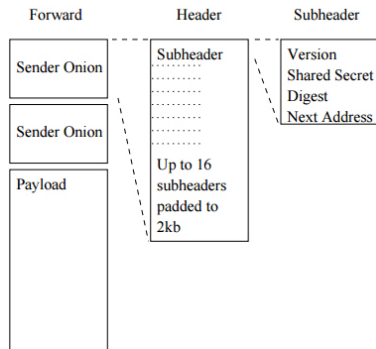
- eigenes Protokoll
- verschlüsselte Verbindung (TLS)
- automatisches Remailerverzeichnis
- Antworten auf anonymisierte Nachrichten

Mixminion - Protokoll



- drei Typen
- Ununterscheidbarkeit
- zweigeteilter Header

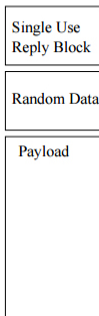
Mixminion - normale Nachricht



Subheader

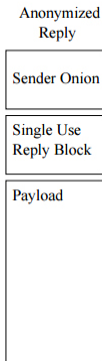
- Master-Secret für TLS
- Adresse des nächsten Remailer (verschlüsselt)
- Prüfsumme zur Überprüfung restlichen Headers

Direct Reply



SURB

- Adresse von Alice (verschlüsselt)
- Pfad durch das Netzwerk (verschlüsselt)
- nur einmal verwendbar
- von Bob nicht entschlüsselbar



- Pfad von Bob durch Netzwerk
- SURB von Alice
- Ziel: Auch Bob bleibt anonym.

Verzeichnisserver

- aktuell verwendete Schlüssel des Remailers
 - aktuellen Zustand
 - die Existenz eines Remailers
-
- redundante Gruppe von Servern
 - regelmäßige Kommunikation
 - untereinander (Verifikation)
 - Remailer (Synchronisation)

- Alice besorgt sich aktuelle Informationen über Verzeichnissever (Schlüssel, Adresse von Mix)
- Aufbereitung der Nachricht gemäß Protokoll
- Alice schickt Nachricht an ersten Remailer
- Jeder Remailer:
 - Prüft Integrität des Headers (Prüfsumme)
 - Speichert Prüfsumme (Replay-Angriff) bis Schlüsseltausch
 - TLS-Verbindung zum nächsten Remailer (verifiziert und verschlüsselt)
 - überträgt Nachricht
- **swap-Operation**
- danach weiter bis Bob Nachricht erhält

Sicherheit?



Sicherheit?

