Fachbereich Elektro- und Informationstechnik
Fachgebiet Kommunikationsnetze
Prof. Dr. Björn Scheuermann

TECHNISCHE UNIVERSITÄT DARMSTADT

Kommunikationsnetze I – 2024

Übungsblatt 4

Aufgabe 1 *Vergleich am 24. Mai 2024*

Ihnen steht eine kryptographisch sichere Hashfunktion H zur Verfügung, welche n Bit lange Hashwerte erzeugt. Wir wollen annehmen, dass diese Hashfunktion keine Schwachstellen besitzt, die beim Berechnen der Umkehrfunktion helfen könnten. Sie versuchen nun einen Datensatz zu erraten, der zu einem bereits existierenden Hashwert h passt – also eine Eingabe a für die Hashfunktion zu finden, die den gegebenen Hashwert $h = H(a)$ als Ausgabe erzeugt. Unter den gegebenen Annahmen bleibt Ihnen hier nur, so lange verschiedene Eingaben auszuprobieren, bis Sie Glück haben.

Natürlich könnten Sie mit sehr viel Glück gleich beim ersten Mal den richtigen – oder mit Pech beliebig oft den falschen Hash erzielen. In dieser Aufgabe sollen Sie berechnen, wie oft Sie raten und den Hashwert errechnen müssen, damit die Wahrscheinlichkeit, eine passende Eingabe zu finden, über 50% liegt.

Ohne weitere Begründung können Sie annehmen, dass die Anzahl E der dafür notwendigen Versuche sich wie folgt verhält. Dabei ist m die Anzahl der möglichen unterschiedlichen Hashwerte, die H ausgeben kann.

$$E(m) \approx \ln(2) \cdot m - \frac{\ln(2)}{2} \approx 0{,}69 \cdot m - 0{,}35$$

(a) Wie groß ist die Anzahl der notwendigen Versuche E abhängig von der Länge n (in Bit) der Hashwerte?

(b) Wie viele Versuche sind dies für eine 64 Bit große Hashfunktion? Wie viele Stellen hat diese Zahl der Versuche im Dezimalsystem? Wenn Sie mit einem Rechner eine Milliarde Hashwerte pro Sekunde berechnen können, wie lange dauert es, diese Zahl von Versuchen zu unternehmen?

(c) Was bedeutet das Ergebnis für die Abhängigkeit des Aufwands von der Länge der Hashwerte? Wie viel mehr oder weniger Versuche benötigen Sie, wenn die Ausgaben der Hashfunktion ein Bit länger werden?

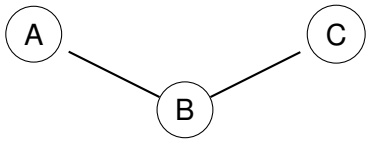
Aufgabe 2 *Vergleich am 24. Mai 2024*

Nehmen Sie an, Sie öffnen eine Website und Ihr Internetbrowser sendet mehrere HTTP-Requests über eine persistente Verbindung mit HTTP/1.1 an einen entsprechenden Webserver. In diesem Szenario ist es zwingend erforderlich (und deswegen auch im Standard vorgeschrieben), dass der Webserver in seinen Antworten jeweils einen `Content-Length`-Header angibt. Weshalb ist das notwendig?

Um zu wissen, wann ein request vorei ist und wo der nächste request anfängt

Bonusaufgabe 4 (6P) *Abgabe bis 24. Mai 2024, Vergleich am 24. Mai 2024*

Nehmen Sie die folgende Netzwerktopologie mit drei Hosts A, B, C und zwei Links an:



Der Link A–B hat eine Datenrate von $8\,\mathrm{Gbit/s}$ und eine symmetrische Einweglatenz von $10\,\mathrm{ms}$. Der Link B–C hat eine Datenrate von $16\,\mathrm{Gbit/s}$ und eine symmetrische Einweglatenz von $40\,\mathrm{ms}$.

(a) (2P) Wie lange dauert es vom Start der Übertragung bis zum Ende des Empfangs, ein 1000 MegaByte großes Paket von A zu B zu übertragen? Wie lange dauert es, dasselbe Paket von B zu C zu übertragen? Welcher Link hat die Übertragung schneller abgeschlossen?

(b) (1P) Wiederholen Sie die vorige Teilaufgabe für ein 10 MegaByte großes Paket.

(c) (1P) Berechnen Sie die (Einweg-)BDPs der beiden Links.

(d) (2P) Im Kontext der Untersuchung interaktiver Protokolle beschreibt das BDP einer Ende-zu-Ende-Verbindung üblicherweise diejenige Datenmenge, die mit der auf der Gesamtstrecke maximal möglichen Datenrate abgesendet werden kann, bevor eine erste Rückmeldung vom Empfänger beim Sender eintreffen kann. Die Effekte endlicher Paketgrößen werden dabei in der Regel vernachlässigt.

Berechnen Sie das BDP der Netzwerkstrecke für ein Protokoll, bei dem Daten von A über B nach C übertragen und vom Empfänger mit ACK-Nachrichten bestätigt werden.