

# Private Information Retrieval

Ida Hönigmann

7. April 2020

## Zusammenfassung

### 1 Idee

In diesem Projekt wird ein System entwickelt bei dem Clients Nachrichten von verschiedenen Servern abfragen können. Die Besonderheit liegt in der Sicherheit, die dem Client gewährleistet wird: keiner der Server weiß welche Nachricht der Client lesen möchte.

Eine Möglichkeit diese Sicherheit umzusetzen ist es den Client alle Nachrichten abfragen zu lassen. Dieses Projekt wählt allerdings einen anderen Ansatz, der eine geringere Menge an Daten, die übertragen werden müssen ermöglicht. Der Nachteil ist, dass ein Client jeweils Anfragen an zwei Server stellen muss. Daher müssen zumindest zwei Server existieren, um Nachrichten empfangen zu können.

### 2 Anwendungsfälle

Dieses System ist für alle Systeme der Form von Servern, die Clients Nachrichten anbieten anwendbar.

Denkbar wäre zum Beispiel eine solche Lösung bei Daten zu Krankheiten. Wenn ein Benutzer nach einer Krankheit sucht, von der er weiß oder befürchtet diese zu haben, möchte er eventuell nicht, dass der Betreiber des Servers Kenntnis davon hat. Das gilt jedoch auch für Personen, die die Krankheit nicht haben und Informationen darüber erhalten wollen. Es könnte angenommen werden, dass sie die Krankheit besitzen, da sie ja danach gesucht haben.

Ein zweiter Anwendungsfall wäre ein Online-Nachrichtendienst. Oft wird nicht dediziert das Verhalten der Nutzer analysiert, sondern nur Logdaten gesammelt um Analysen über die Auslastung der Systeme zu erstellen oder falls ein Problem auftritt dieses nachverfolgen zu können. Jedoch kann es vorkommen, dass eine andere Organisation diese Logdaten anfordert um Informationen über einen bestimmten Benutzer oder eine Benutzergruppe erlangen zu können. Dazu muss die Organisation, die die Daten anfordert in irgendeiner Form über mehr

a	b	a xor b
0	0	0
0	1	1
1	0	1
1	1	0

Tabelle 1: Wahrheitstafel der xor-Operation

Macht verfügen. Dies ist zum Beispiel bei Regierungen, auch anderer Länder, der Fall.

In diesen beiden und vielen anderen Fällen kann eine Implementierung eines *private information retrieval* die Benutzer schützen.

### 3 Funktionsweise

Die in diesem Projekt implementierte Version von *private information retrieval* nutzt die Eigenschaften der Exklusiv-Oder-Operation (xor).

#### 3.1 Exklusiv-Oder-Operation (xor)

Die xor Operation erhält, gleich der "und"- (Englisch "and") und der "oder"- (Englisch "or") Operation zwei Signale als Eingang und liefert ein Signal als Ausgang. Dabei kann man das Verhalten von xor als "entweder oder" beschreiben. Die Wahrheitstafel in Tabelle 1 definiert das Verhalten von xor genauer.

Die Eigenschaft, die dieses Projekt benötigt ist folgende: Jede Folge von Bytes  $a$  mit einer anderen Folge von Bytes  $b$  nach Anwendung der xor-Operation ergibt eine Folge von Bytes  $c$  für die gilt, dass  $c \text{ xor } a = b$  und  $c \text{ xor } b = a$ .

#### 3.2 Inhalt der gesendeten Nachrichten

Ein Client baut zwei Verbindungen zu unterschiedlichen Servern auf. Statt nur die gewünschte Nachricht anzufordern, verlangt er das Produkt von xor-Operationen vieler Nachrichten.

Welche Nachrichten der Client anfordert, die Reihenfolge der Nachrichten, sowie die Anzahl sind irrelevant, solange beide Server die fast die gleichen Nachrichten verwenden. Der einzige Unterschied darf darin liegen, dass in einer der beiden Anforderungen die gewünschte Nachricht enthalten ist, und in der anderen Anforderung nicht.

Beide Server schicken dann das geforderte Produkt der Nachrichten zurück.

#### 3.3 Kalkulation der gewünschten Nachricht

Der Client muss nun ebenfalls die xor-Operation auf die beiden erlangten Antworten der Server anwenden. Das Ergebnis ist die gewünschte Nachricht.

## 4 Möglichkeiten einer Untergrabung

## 5 Implementation

### 5.1 Command Line Interface

Das Command Line Interface wurde mit CLI11 umgesetzt.

Um Informationen über die Verwendung der beiden Programme *client* und *server* zu erhalten kann die Option `--help` (oder als Kurzform `-h`) spezifiziert werden.

Sonstige Optionen, die bei *client* spezifiziert werden können sind:

- `--list / -l`: gibt eine Liste aller Nachrichten Titel und deren Index aus
- `--index / -i`: gibt die Nachricht des Index aus
- `--port1 / -p`: Port des ersten Server an dem die Nachrichten angefordert werden sollen
- `--server1 / -s`: IP Adresse des ersten Server an dem die Nachrichten angefordert werden sollen
- `--port2 / -q`: Port des zweiten Server an dem die Nachrichten angefordert werden sollen
- `--server2 / -t`: IP Adresse des zweiten Server an dem die Nachrichten angefordert werden sollen
- `--verbose / -v`: gibt zusätzliche Debug Meldungen aus

Die Optionen bei *server* sind folgende:

- `--port / -p`: Port, an dem die Verbindung aufgebaut werden soll
- `--verbose / -v`: gibt Debug Meldungen aus
- `--loop / -l`: lässt den Server in einer Schleife laufen

### 5.2 Aufbau und Speicher der Daten

Da sich dieses Projekt auf die sichere Kommunikation fokussiert werden die Daten einfachheitshalber in einem Textfile gespeichert. Die Datei *data.txt* wird als Datenspeicher verwendet. Der Aufbau der Daten ist abwechselnd der Titel der Nachricht und dann die eigentliche Nachricht. Da in dem Text der Daten prinzipiell alle Zeichen vorkommen können wird als Trennzeichen ein Zeilenumbruch (`\n`) verwendet.

Das Datenfile wird ausgelesen nachdem der Server gestartet wurde.

- 5.3 Byte-Operationen
- 5.4 Zufällige Nachrichten anfordern
- 5.5 Kommunikation