Aus dem Institut für Technische Informatik der Universität zu Lübeck
Direktor: Prof. Dr. rer. nat. Rüdiger Reischuk

Privately computing the intersection of two SNP sets

Berechnung des Schnitts zweier SNP Mengen unter Erhalt der Privatsphäre

Praktikumsbericht
im Rahmen des Studienganges Medizinische Informatik
der Universität zu Lübeck

vorgelegt von
Niklas Jobst

ausgegeben und betreut von
Prof. Dr. rer. nat. Rüdiger Reischuk

mit Unterstützung von
Florian Thaeter

Lübeck, den November 16, 2017

Abstract

Ziel dieses Praktikums war es zu erörtern, wie zwei Parteien die Ähnlichkeit ihrer DNA berechnen können, ohne, dass dabei eine der Parteien Informationen über den genetischen Code der jeweils anderen erlangt.

Die Grundlagen für diese Berechnungen basieren auf bereits existierenden Methoden, mit welchen der Schnitt zweier Mengen unter Sicherung der Privatsphäre berechnet werden kann.

Im Zuge dieses Praktikums habe ich drei dieser Methoden mit Bezug zum gegebenen Anwendungsfall implementiert und deren Effizienz miteinander vergleichen:

- R.Egert et al. : Privately Computing Set-Union and Set-Intersection Cardinality via Bloom Filters, LNCS volume 9144, 2015
- A.Davidson et al. : An Efficient Toolkit for Computing Private Set Operations, LNCS volume 10343, 2017
- S. K.Debnath et al. : Secure and Efficient Private Set Intersection Cardinality Using Bloom Filter, LNCS volume 9290, 2015

Contents

1 Einleitung

1.1 Ähnlichkeit der DNA

In diesem Projekt wird die DNA der beiden Parteien als Mengen betrachtet. Aufgrund der Tatsache, dass der Großteil der DNA bei allen Menschen identisch ist, nutzte ich genetische Marker, welche die DNA unterscheiden. Der Schnitt dieser beiden Marker dient dann als Maß der Ähnlichkeit der jeweiligen DNAs.

1.2 Genetische Marker

Unter genetischen Markern werden Bestimmte klar definierte Sequenzen und Positionen im genetischen Code können dazu genutzt werden Personen zu identifizieren.

1.2.1 Personalisierte Medizin

In der personalisierte Medizin werden individuelle Eigenschaften von Personen berücksichtigt die

1.2.2 SNPs

1.2.3 INDELs

1.3 Anwendung

1.3.1 Personalisierte Medizin

In der personalisierte Medizin werden individuelle Eigenschaften von Personen berücksichtigt, insbesondere genetische In der Personalisierten Medizin sind Therapien bestimmte genetische Profile gekoppelt. Um festzustellen, ob eine Therapie für einen Patienten zulässig ist, muss daher zunächst sein genetischer Code mit dem für diese Therapie notwendigem verglichen werden. Derzeit werden diese Vergleiche ohne die entsprechenden Datensicherheits-Vorkehrungen vorgenommen. Ziel dieses Praktikums war es durch Anwendung der genannten Methoden die Sicherung der Privatsphäre bei der Durchführung eines solchen Vergleichs zu erhöhen.

2 Methoden

2.1 Bloom Filter

Alle diese Methoden basieren auf sogenannten Bloomfiltern. Hierbei handelt es sich um eine Technik um festzustellen, ob bestimmte Daten in einem Datensatz vorhanden sind oder nicht. Sie bestehen aus einem mit Nullen vorinitialisiertem m Bit langen Array und k Hashfunktionen, welche auf die Positionen des Arrays abbilden.

Zur Initialisierung werden auf jedes Element des Datensatzes alle k Hashfunktionen angewendet. Die zur Ausgabe der Hashfunktionen korrespondierenden Bits im Array werden darauf hin auf Eins gesetzt.

Soll für ein Datenelement geprüft werden, ob dieses Teil des Datensatzes ist, werden alle Hashfunktionen auf dieses angewendet.
Nur wenn alle Positionen im Array an den korrespondierenden Punkten der Ausgabe dem Wert Eins entsprechen wird angenommen das sich das Element im Datensatz befindet.

Diese Überprüfung ist nicht resistent gegenüber

2.2 Kryptosysteme

2.2.1 Homomorphie

Homomorphie bezeichnet eine Eigenschaft von Kryptosystemen. Ein Kryptosystem ist genau dann homomorph gegenüber einer mathematischen Operation, wenn Berechnungen im Ciphertext mit dieser Operation denen im KLartext entsprechen.

2.2.2 Elgamal

1. Bei Elgamal handelt es sich um ein im Jahr 1985 vom Kryptologen Taher Elgamal entwickeltes Public-Key-Verschlüsselungsverfahren. Elgamal ist eine Erweiterung des Diffie-Hellmann Schlüsselaustausches.

Schlüsselerzeugung

Zunächst wählt der Client eine endliche zyklische Gruppe Z der Ordnung q mit einem Generator g.

- Secrect key: Der Client wählt eine zufällige Zahl $a < q$ mit dem $GGT(a, q) = 1$. Dies ist der Secret key
- Public Key: Der public key ist dann $P = g^{\mathrm{a}}$

Verschlüsselung

Sei $m \in Z_{\mathrm{q}}$ die zu versendende Nachricht. Dann wählt der Server eine zufällige Zahl $r < q$ mit dem $GGT(r, q) = 1$. Nun berechnet sich $c_1 = g^{\mathrm{r}}$ sowie $c_2 = P^{\mathrm{r}} * m$. Der Ciphertext besteht so aus $C = (c_1, c_2)$.

Entschlüsselung

Zur Entschlüsselung wird $\Sigma = c_1^{-q} * c_2 brechnet$.

Homomorphie

Elgamal ist homomorph gegenüber der Multiplikation

$$E(m_1 * m_2) = (E(m_1) * E(m_2))$$

Sicherheit

Zuverlässigkeit

2.2.3 Pailier

Das Schlüsselpaar wird folgendermaßen generiert:

2.2.4 Goldwasser-micali

2.3 Implementierte Algorithmen

2.3.1 Algorithmus 1 - Elgamal

Der hier beschriebene Algorithmus wurde zunächst im ... veröffentlicht. Es wurden Algorithmen für unterschiedliche Konstellationen postuliert. An dieser Stelle habe ich den zwei Parteien Fall genutzt.

Der Client erstellt zu Beginn einen Boomfilter seiner Daten. Dabei wird jedes Datenelement einzeln Zur Verschlüsselung wählt der Client zunächst public und secret key nach elgamal. Daraufhin wird jedes Bit des Bloomfilter Arrays einzeln verschlüsselt. Hierzu werden die zu sendende Nachrichten so gewählt das $m = g^B F[i]$ Dies führt dazu, dass m an Stellen an welchen der Bloomfiltern einen 1 besitzt einer 1 entspricht und g wenn er dort eine 0 besitzt. Dann berechnet er seine diesen

$$(R_i, S_i) = (g^{\mathrm{r_i}}, pk^{\mathrm{r_i}} * g^{\text{1-BF}_1[\mathrm{i}]})$$

$$S_{\mathrm{i}} = pk^{r_i} \ * \ \begin{cases} g^0 = 1 \ bei \ BF_1[i] = 1 \\ g^1 = g \ bei \ BF_1[i] = 0 \end{cases}$$

Aufmultiplikation von R_i bzw S_i an jenen Stellen, an welchen $BF_2 = 0$ ist. Rerandomisierung mit g^s bzw. pk^s Diese Berechnungen sind ohne Datenverlust aufgrund der Homomorphie Eigenschaft von Elgamal möglich

$$V = (g^{\mathrm{s}} * \Pi_{i:BF_2[i]=0} R_i)$$

$$W = (pk^{\mathrm{s}} * \Pi_{i:BF_2[i]=0} S_i)$$

$R_i,\ S_i$ aus vorherigem Schritt in V,W einsetzen.

$$V = (g^{s+r_{i_1}+r_{i_2}+ \ \cdots \ +r_{i_k}})$$

$$W = \begin{cases} pk^{s+r_{i_1}+r_{i_2}+ \ \cdots \ +r_{i_l}} * 1 \ falls \ BF_1 = 1, BF_2 = 0 \\ pk^{s+r_{i_1}+r_{i_2}+ \ \cdots \ +r_{i_m}} * g^x \ falls \ BF_1 = BF_2 = 0 \end{cases}$$

$$W = (pk^{s+r_{i_1}+r_{i_2}+ \ \cdots \ +r_{i_k}} * g^x)$$

2.3.2 Algorithmus 2 - Paillier

2.3.3 Algorithmus 3 - Goldwasser-Micali