Chatting about data

Lorenzo Martinico

MInf Project (Part 2) Report
Master of Informatics
School of Informatics
University of Edinburgh

2019

Abstract

Conversation privacy is not one of the primary concerns in the development of chatbots, which require the use of powerful Natural Language Processing engines to function. Applications of the technology in healthcare require confidentiality of patient information. We propose the first decentralised chatbot protocol, designed to protect message contents and user identities from a powerful global adversary that controls the NLP server. Our design moves the central data processing to the chatbot client and hides the message sender through anonymous routing and the removal of linguistic features. A variety of attacks and their mitigations are discussed. Initial benchmark results are reported on language transformations.

Acknowledgements

There are too many people to thank for helping me get to this point. Limiting myself to the creation of this report, my thanks go to:

- Lorenzo, Yanna and Cee, for double checking my maths and making sure, one way or another, that I got through the last few weeks;
- Brennan, for some thoughtful last minute feedback, and for putting up with my impossible hours;
- my parents, for the constant encouragement and support throughout my academic career, and beyond; and to my father in particular for patiently going over the formatting of more than 200 citations;

  Special thanks go the whole faculty and staff of Informatics and Edinburgh University, for enabling a stimulating atmosphere for learning and research, and in particular:

- Kami Vaniea and Rico Senrich, for the insightful lectures that have shaped much of the content of this project;
- Adam Lopez, Tariq Elahi, and Myrto Arapinis, for their availability to chat, and for steering me in the right direction;
- Markulf Kohlweiss, for reassuring me that it all made sense;
- and of course, my supervisor, Stuart Anderson, for letting me take this project in a new direction that I found exciting, and then helping me figure out what it should actually look like. I can not thank you enough for devoting so much of your time to my project, your always invaluable feedback, and constant encouragement when I got stuck.

Table of Contents

Chapter 1

Introduction

The first phase of the *Chatting about data* project [108] involved the development of chatbot systems for assisting users achieving their health and fitness goal. During the development of *Healthbot*, a chatbot for food logging, it became obvious that the data inserted into the system through conversation would not be adequately secured for a health-related application. In this second part, we set out to develop a secure chatbot platform. We now summarise the work carried out in the first part of the project, followed by an outline of this report.

1.1 Previous work

Our work in the first part of the project resulted in the creation of *Healthbot*, a nutritional assistant that uses a chat interface to collect a food diary, using the conversational and multimedia features of modern chat applications to augment the collection process. In particular, our implementation was the first prototype to allow image capturing and textual input in a nutritional chatbot setting, and further facilitated data entry by asking users to estimate their portion sizes across predefined categories. By hosting the chatbot on the *Facebook Messenger* platform, Healthbot was given a privileged placement, as one of the contacts in a user's address book. This allowed the bot to be easily retrievable for quick inputs, and the display of timely notification in the form of reminders to log more food, or tips for healthy eating. The rest of the chatbot infrastructure relied on the use of various cloud computing services, such as *Google Dialogflow* to power Natural Language Processing tasks.

The initial version of Healthbot implemented just a few simple handcrafted nutritional rules, and was hindered by various bugs in the NLP ruleset. However, despite some usability issues, we conducted an experimental study showing that the chat interface showed promise in sustaining engagement in users, a big problem for existing food logging applications.

Despite this positive result, another more worrying discovery emerged from the study: there is little to no expectation of privacy for chatbot users. As many modern web

applications, a typical chatbot will have a complex architecture, relying on various externally controlled services to get some domain specific information or store and manage a database. But while these services typically disclose a small amount of information, a chatbot can leak an incredible quantity of facts about a user. Medical chatbot users have been known [219] to confide more information to a chatbot than what they would to a human therapist. As a consequence, both the chatbot platform provider and the NLP server, who will be sent the entirety of the conversation between patient and chatbot, will be able to read the patient's most private thoughts. This is not only invasive, given that these service providers often (like in Healthbot's case) correspond to major ad tech companies; it is also possibly illegal, unless appropriate precautions are taken, since medical data is protected by many legislative bodies.

1.2 Summary of Part 2

Confidentiality in cloud computing application is an active area of research. Solutions like differential privacy have been adopted by industry, and others like Fully Homomorphic Encryption or Secure Multiparty Computations are being the growing focus of cryptographers. But despite much progress in the security of instant messaging, the privacy of chatbots has received little attention. In this work we propose and analyse a theoretical private chat protocol for a Healthbot-like chatbot, while offering suggestions for a concrete implementation. Additionally, we conduct some benchmark experiments on the efficacy of some components in our design.

We propose two variations of a generalisable decentralised design, where processing of user data is shifted from the cloud chatbot provider to a locally hosted chatbot application. We provide either the option of storing data and conducting aggregated computation on a trusted server, using state of the art medical data handling protocols; or an alternative, more speculative, offline-first model, where all users of the chatbot cooperate to improve the efficacy of each other's solution, through federated machine learning. In both instances, users' diets are analysed both on their individually collected data, and on wider population trends. To decrease the complexity of the chatbot implementation, Natural Language handling is offloaded to a remote server. Since we can not encrypt the contents of the message for the server to give us useful information, in order to preserve confidentiality, each conversation is anonymised using a protocol built on top of the Tor and Crowds anonymity networks, replicating a message from different users with artificial time delays. We perform natural language transformations, like word substitutions, to obfuscate potentially sensitive information, such as the actual contents of the user's diet. Through our design, we eliminate many of the potential sources of information leakage discovered in Healthbot's architecture, provide the user with more control of their own data, and reduce the capabilities of a malicious Natural Language service provider, at the cost of design complexity and efficiency. Having outlined our theoretical design and discussed possible attacks from an adversary and mitigations, we conduct some experiments to establish how feasible our protocol is, using current state-of-the-art language models and available data sources.

The report is structured as follows: in the next chapter, we discuss previous works

on the issues of privacy and how it relates to medical privacy, as well as additional background information for concepts and tool we will use in later chapters. Chapter 3 begins with a threat analysis of the Healthbot infrastructure, and follows with a detailed description of our proposed architecture and how it fixes the flaws of the previous model. Chapter 4 describes the routing protocol and its formal analysis, and Chapter 5 describes potential attacks at the language level to defeat anonymity, and linguistic transformations needed to counteract them. Chapter 6 outlines our attempts in validating some of our protocol components experimentally, and closing remarks are offered in the final Chapter 7.

Chapter 2

Background

We discuss background information on our core problem, increasing privacy in medical chatbots, as well as previous literature retaining to concepts we use for our proposed solution.

2.1 A quick cryptography primer

Classically, cryptography has been known as the science of hiding information: classic cryptosystems, most of which were based on principles of symmetric cryptography (both the sender and the receiver of the message share the same key) were primarily concerned on ensuring the property of confidentiality through encryption. With the development of modern cryptography, parallel to that of digital computers and internet communications, many further uses have emerged, from integrity (making sure that data is not modified in transit or at rest), to authentication (getting assurances on whom it is that you are communicating with), from secret sharing (breaking up a secret into different shares such that it requires multiple share holders to unlock the secret), to the recently increasing in popularity electronic cash and online voting.

Many recent advances derive from the concept of public key cryptography, first formulated in 1883 as Kerckhoffs's principle [92], but with the first practical formulation in the 1970s [47, 162]. The difference between symmetric and public-key cryptosystems consists in the capabilities of each key. In symmetric cryptography, the two participants in a protocol need to share a single secret key. This causes a fundamental problem in how keys can be distributed securely, which public-key cryptography addresses by decoupling the keys into a public and private key. In the domain of encryption, the public key, which can only encrypt a message, is published widely. This allows anyone to encrypt a message using the public key, and send it to the entity that published the key. The encrypted message can then only be decrypted using the private key, which should always be kept secure. Similarly, for a digital signing algorithm, the private key is only allowed to sign a document, and the public key can only verify the authenticity of the signature. By combining encryption and signing, the message satisfies both properties of confidentiality and authentication.

Public key infrastructure is crucial to the functioning of the modern web, driving fundamental protocols such as SSL/TLS for establishing secure internet connection (by authenticating the server), or DKIM in email authentication. While in most cases users are not exposed to the mechanisms of these cryptographic protocols, some tools exist for users to interact with them, the most popular being Pretty Good Privacy (PGP), and its implementation GNU Privacy Guard (GPG) [11]. Notoriously difficult to use [211], PGP allows user to sign and encrypt email messages to other PGP users. Because signed email clients attach the public key with the message, everyone can verify that emails are signed by the owner of the key; but only PGP users can successfully sign an email. Similarly, while everyone can encrypt an email to someone's public key, only PGP users can receive a PGP-encrypted email. If signing an email attests to the sender possessing a private key, it does nothing to verify the user's actual identity. For this purpose, a complex public key infrastructure (PKI) has been set up, involving decentralised key servers that user can submit their keys to, and a Web of Trust built upon the premise of PGP users signing the public keys of people they have verified actually control their matching private key. Additionally, PGP allows users to manage multiple identities, and revoke old keys they lose access to or stop trusting, by generating subkeys from the original master key. These subkeys are associated to a primary master key, which can revoke them from the Web of Trust by issuing a revocation certificate. Users are encouraged to keep the master key securely on an offline medium, and only use the subkeys on actively used devices for a short period of time.

Much of today's cryptography relies on the Computational hardness assumption. Many cryptographic algorithms in use today are designed to reduce to a hard computationally complex problem, such as integer factorisation. These algorithms are deemed secure because they reduce to a problem for which there are no known efficient (polynomial time) solutions. Such an algorithm can break when faced with a theoretical adversary with unlimited or nearly unlimited computing power - or, more realistically, by a future adversary with a significant increase in computation from what was thought possible when the algorithm was designed [168]. While today's cryptographic algorithms are considered to be safe from any currently existing adversary, the threat of powerful quantum computers that could break them looms close, and has required cryptographers to come up with quantum-safe algorithms [24], which fall in the category of Information-theoretical cryptography. These are ciphers that can only be broken if the adversary has enough information about them [176]. The most notable, and perhaps simplest encryption scheme, the One time pad, was invented in the 1800s by banker Frank Miller [20]. To encrypt a message, a logical XOR between the message's text and a randomly generated key of the same length for encryption is performed, and a second XOR between the produced ciphertext and the same key is used for decryption. The security of this system relies on the random number generated being truly random, and the key only being used to encrypt a single message. While the security of the one time pad is mathematically perfect, it is practically unusable, both for the difficulty of actually generating a random key (cryptographically secure random number generator require high sources of entropy [56]), and because of the need to distribute a key of the same length of the message, which needs to be discarded after the first message.

A special case of information theoretical security is perfect secrecy. This is achieved

by encryption schemes such that, given a ciphertext, no information can be extracted about the original plaintext. A weaker assumption, relying on computational hardness, is semantic security, for which the probability of extracting any information from the plaintext is negligible for any polynomial time attackers. Cryptographic proofs sometimes employ *random oracles*. Oracles are theoretical black box functions that respond to any query with a unique string drawn from a uniform probability distribution. They can be controlled by either the attacker or the challenger (an algorithm trying to verify that the attack has been successful), as an upper bound in cryptographic proofs: if a protocol is secure against a random oracle, it will also be secure in a setting where this theoretical device does not exist.

To protect computationally secure algorithms from future adversaries compromising a long term encryption key, some cryptographic algorithms are designed such that any message sent before the key was broken can not be decrypted. This is achieved through the non-deterministic creation of individual session keys for each single session. Capturing a single session key will also not give any information about any other session keys, or the master key [3].

Attackers of cryptographic protocols can be broadly described in two categories: passive and active [66]. A **passive** adversary can listen into all traffic in the system (in which case, it is also a global adversary) or only to messages going in and out of one participant (a local adversary), but is unable or unwilling to modify it. This attacker aims to break confidentiality by obtaining some secret information. An **active** adversary is more powerful, as they can change the behaviour of the protocol by acting as a man-in-the-middle who can intercept and modify communications, block connections from reaching one party, deleting data and edit packet headers. These attackers can impact confidentiality, integrity, and authentication.

2.1.1 Cryptographically secure computation

More recent applications of cryptography have enabled the development of algorithms that allow conducting computation on encrypted data. Fully Homomorphic Encryption is a semantically secure, asymmetric encryption scheme, with the added capacity of running any arbitrary circuit (and thus any computable function) on the encrypted ciphertext without the need to decrypt it and producing a result that is also encrypted with the senders' public key. Its first practical formulation was created by [68], and it has since been followed by different techniques [109]. Despite the many recent attempts to improve efficiency over the original, the performance costs of FHE are still significant compared to running circuits on plain text data. [203] proved that a fully homomorphic scheme is only feasible for single-client computation, where data travels uni-directionally from client to server, and a simple response is sent from the server to the client; it does not generalise to the case of multi-client server applications where the encryption scheme incorporates access control on what input data each party has access to.

This problem was addressed, before the advent of homomorphic encryption, by the more general concept of Secure Multiparty computation (or Secure Function Evalua-

tion), first formulated by [216] in terms of computing a function with multiple inputs, where different parties know one input and can communicate between each other without directly revealing their input to other participants, or for the results to leak to external observers. In the two-party setting this is achieved by a garbled circuit protocol, where an encrypted circuit is computed by one party and sent alongside its encrypted input so that the other party can feed in his own input and produce an encrypted output that can then be collaboratively decrypted. Alternative schemes to garbled circuits also use secret sharing, where each party possesses a share of a larger secret, and no information can be revealed until multiple shares are combined. Through mutual communications, intermediate operations can be run so that different parties can obtain various intermediate results, until the final solution is computed collaboratively. Like operating on Fully Homomorphic ciphertexts, leading software implementations of MPC incur in significant performance penalties [15], and there are still several open problems in the field [165].

2.2 Anonymity protocols

The concept of anonymity can be defined as the property of a message (a communication between several human parties) not to be associated with its author. Pseudonymity is when the author of a message hides behind a fake name, often fixed [139]. Starting in the 1970s, a vast literature on making computer usage anonymous has been developing [193]. Since early in the history of the internet, social practices of anonymity or pseudonymity were common, and are still used across vast areas of the internet [49], although that is mostly not true for social media [207]. Most users however are usually traceable through the internet, from logging of IP addresses to tracing email headers and placing unique identifiers in cookies. Although adoption of the TLS / SSL protocol and serving pages encrypted over HTTPS has reached 79% of webpages this year [99], before that an Internet Service Provider and anyone connected to a wireless network could read all information being sent by a user (for some websites still a risk, due to downgrade attacks [138]).

The solution of routing all traffic through a proxy, a third-party server, will make traffic appear to originate from the proxy server and is useful for avoidance of simple traffic tracing (most people today use Virtual private networks as a proxy [78]). However, a malicious proxy server can conduct both passive and active attacks, as they are fully in control of all traffic received.

[44] first addressed the issue of traffic analysis (understanding who communicates with whom in a network) by establishing the concept of Chaum Mixes, based on public key cryptography. A mix is mediated by a server, which people can send messages to (like email). Messages are encrypted to the receiver's public key, to which the address of the receiver is appended, and then encrypted with the mix's public key. A mix can thus receive a batch of emails, which they will hold onto for a period of time. After that, they will remove the first layer of encryption using their private key, and forward the message to the destination server, so that they might decrypt it using their own private key. The sender might also include a return address, along with a new public key, and

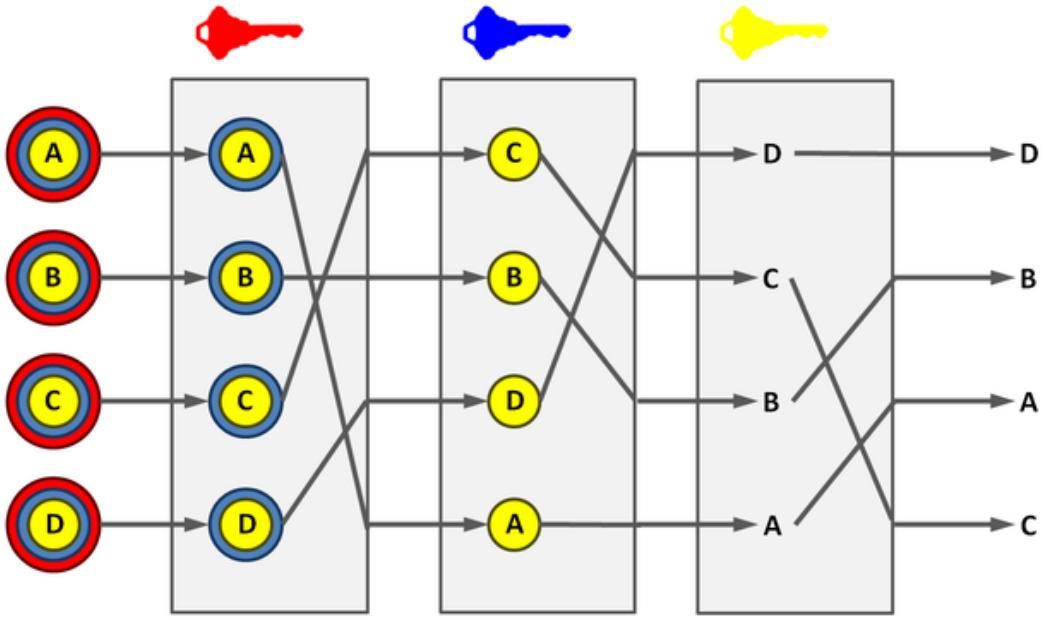

Figure 2.1: A Chaum Mix cascade. The last layer of encryption (to the final server) is not explicitly displayed. Source Primepq at English Wikipedia, “Red de mezcla” ⓒ

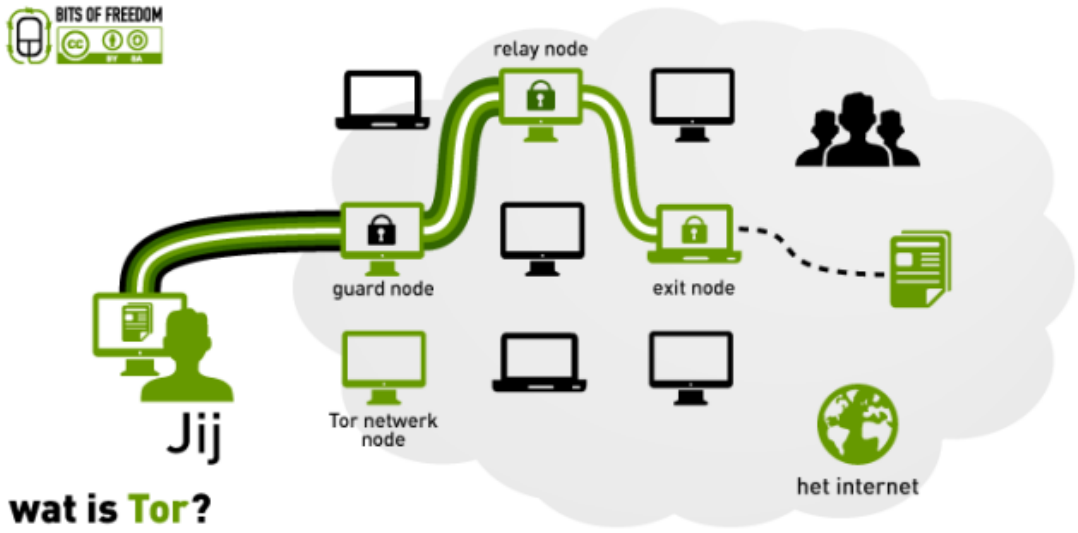

Figure 2.2: The route of a document request through the Tor network. Each *layer* in the connection correspond to a layer of encryption. Source Bits of Freedom ⓒ

encrypt it to the mix's public key, so that the receiver might send a response encrypted under the new key, but only the mix will know where to deliver it. To force anonymity on a malicious mix, a cascade (a series of mixes) might be used, requiring only one honest mix to be present. The sender would have to encrypt the message multiple times, using the public key of each mix in the sequence. When passing each batch of messages along, mixes would sign their delivery to the next mix in the cascade, so that if a message was dropped from its history the culprit would be identifiable. A cascade of Mixes is illustrated in Figure 2.1.

While mixes delay the distribution of messages through batching, Onion Routing [69] (illustrated in Figure 2.2) adapts the concept to work in real time. By eschewing the mixing process, Onion Routing speeds up the message passing process, at the cost of becoming vulnerable to global attackers [189].

The most popular Onion Routing network is Tor [54], with more than 7000 nodes and 2 million daily active users today [195]. Tor has been widely used by dissidents, journalists and whistleblowers to avoid detection from hostile governments, as well as criminals [140]. By hosting untraceable *hidden services* on the network rather than on the regular internet, a parallel community of websites peddling illicit goods has developed (commonly known as *The Dark Web*). Besides web browsing, a whole range of applications rely on Tor for their backend, from secure whistleblowing tools to file sharing solutions, as well as censorship circumvention services, and deep integration in browsers, operating systems and messaging services [196].

An alternative to mixes and onion routing is the *Crowds* protocol [160]. Crowds does not try to preserve absolute anonymity, but rather envisions anonymity to be a continuum where a party can hide behind plausible deniability. On this scale, a party could be *beyond suspicion* if they have the same probability of anyone else of sending the message; *probably innocent*, if the likelihood of a specific agent sending the message is the same as that of not sending it; and *possibly innocent*, if there is a non-trivial probability of the sender being someone else. To protect sender and receiver anonymity, the Crowd protocol distributes a message randomly through a chain of Crowd peers (which might include the sender itself), such that for each node in the chain the message is forwarded to another peer with probability p_f, and sent to the destination otherwise. This guarantees beyond suspicious sender anonymity to the destination server, and probable innocence to a number of colluding peers. The protocol is however limited in its defence against a local attacker, which can observe that a request originating from a peer does not correspond to an incoming request from another source. Additionally, there is no defence against active attackers: all traffic is encrypted through pairwise symmetric keys between each peer, making it possible for a malicious one to modify the contents of the message, or to selectively put up a denial of service attack.

It is uncertain how resilient anonymity protocols can be. [214] and [180] formally set limit to the length of time by which a node can remain anonymous. Predecessor attacks are used to deanonymise over time a message sender by counting how often messages arrive from their destination; Sybil attacks can be performed by an adversary adding a large number of nodes to a protocol, to increase the probability of a message being routed through one they control.

2.3 Privacy

The Cambridge Dictionary [53] defines privacy as both "Someone's right to keep their personal matters and relationships secret" and "The state of being alone". While the first definition might be considered more relevant when talking about digital privacy, the second is too. The two definitions represent two aspects of modern digital privacy: the kind of information we give about ourselves to the rest of the world, and what information from the rest of the world we let into our awareness.

Accordingly, privacy is not a binary choice between broadcasting all information publicly or locking everything up in a secret vault no one can access, which corresponds to many formulations of the legal rights in privacy legislation [42], with choice usually modelled as providing an opt-out option. [133] models privacy choices in terms of information flows. People have expectations of how data moves (in terms of what actors and information are involved in a transaction, and the constraints of how data is moved). Expectations of how this information is handled offline in different contextual situations correspond to expectations on how it should be handled online as well. These often diverge from reality. Users of online services are often not aware of these flows of information due to the obscurity of privacy policies. Attempts to create machine-readable standards for privacy preferences have either failed to gain sufficiently widespread adoption [172], or are oversimplistic [91]: there appears to be a trade-off (the *transparency paradox* [25]) in how much information is given about data usage, and how well laypeople are able and willing to put the time in to understand [115], making true *informed consent* impossible. [134] argues that information flow must be regulated to match between offline and online contexts, rather than being guided by economic principles.

2.3.1 Privacy legislation

Since [210], privacy has been set up to be a trade-off between public good, as it pertains to the spread of information of public interest and free speech, and individual rights, as an evolution from the basic right of physical protection into that of preventing harm from anguish that might be caused by the intrusion into one's private life, the disclosure of embarrassing information, unwanted publicity or the misappropriation of one's name and likeness. [185] further differentiates privacy into 16 kinds of privacy problems, subdivided under the categories of *Information Collection*, *Information Processing*, *Information Dissemination*, and *Invasion*. These can create harms in society not just by hurting an individual directly, but by causing wider effects on society, such as chilling effects over free speech and association, loss of trust in public entities and businesses, power imbalances on the side of who has access to information, and loss of sovereignty to the individual. Thus, the harm to an individual's privacy can not be considered as a 0-sum game, as they affect society as a whole. The argument against the common rebuttal to privacy, "I have got nothing to hide", is that privacy is not just about hiding things, and merely framing in that context disenfranchises portion of the populations that benefit from other aspects of privacy. There is also a discrep-

ancy between what might be considered as *public interest* (matters that affect a large number of individuals, or a single person but with the potential of affecting a large portion of the public), and the *interests of the public* (things that affect *some* proportions of the public). Invasion of privacy into situations that fall into the latter (such as tabloid coverage of a celebrity) are often incorrectly justified as the former [125]. Furthermore, [107] argues that it is necessary, for society to progress and evolve, that law enforcement not be perfect, so that citizens might skirt the law and open the door to its reformation through accepted social practices.

Despite the lack of precision in the definition of the term *privacy*, the last century has seen some version of the concept being ratified in law. From the Universal Declaration of Human rights [2], which states that

> No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honour and reputation. Everyone has the right to the protection of the law against such interference or attacks.

Many countries around the world have adopted legislation that protects the privacy of subjects, from medical patients to information communication [95]. Among the most comprehensive laws is the recent European Union General Data Protection Regulation (GDPR), which provides [159] individuals the rights of

- being informed of data collection
- accessing the information collected from them
- rectifying any incorrect information
- requesting the erasure of any information collected
- restricting the processing of information for any specific purpose
- accessing their data in a machine-readable format (data portability)
- objecting to the processing of their data
- requesting information about how their data was used in regard to automatic information decision and profiling

Even though all these legislations now protect user privacy, governments across the world still maintain a vast amount of power in how much information they can gather about individuals. In 2013, whistleblower Edward Snowden published [98] classified documents from the National Security Agency in the USA. For the first time, some of the immense surveillance capabilities of the agency were revealed to the public, along with its cooperation with foreign national security bodies, such as the UK's GCHQ, to illegitimately conduct covert surveillance on its own citizens. Furthermore, large US tech companies, whose services are provided to billions of users across the world, were shown to have been part of a data sharing program with the US government [71].

Besides collaborating in government lead surveillance, many companies on the internet rely on profiling web surfers to sell them better ads and services. In a process dubbed as Dataveillance, profiles are built on users' demographics and interests, based

on their online activity. These involve not only information that is directly submitted to an online service (and not necessarily the one that stores the data afterwards), but also what other behaviour implicitly signals to a finely tuned Machine Learning model [101]. This kind of collection falls within the logics of Dataism, the belief that it is possible to extract hidden insights from the mass collection of data and metadata through "data mining", a principle espoused by big tech, government organisation and academia. This idea relies on an underlying assumption that tech platforms collect "real data", as it exists, despite the fact that each collected data point does not correspond to any "true fact", but it is mediated by the interaction with the collection tools, as a mean to achieve some predetermined purpose [202].

Currently at the centre of attention for its predatory behaviour on user privacy is Facebook , who sold access to its data to a UK based data analytics firm whose goals where manipulating elections through targeted advertisement campaigns [39]. This is only one of the many companies that collect hundreds of data points on web users during their daily activities [46], albeit perhaps some of the most personal. While users might not object to a little information about themselves being recorded by various entities, or in an anonymised form, very small datasets can be combined with other sources to reveal specific information that had not been intended for disclosure, or deanonymise an individual from the database (data triangulation) [188, 130].

2.3.2 Mathematical formulations of privacy

To prevent data triangulation and guarantee the anonymity of data subjects, it becomes necessary to apply some modifications to a database. One property used to ensure this is k-anonymity [187]. Given a set of attributes that might lead to the identification of a single user in the database, quasi-identifiers, k-anonymity requires the number of database entries that share equal value for these to be k. k-anonymity can be enforced either through generalisation (changing the field of a database to include values that are less specific e.g changing a Date of birth field into Year of birth), or suppression (removal of specific fields), which are combined to minimise the number of modifications. This approach is not perfect, however, since data in other sensitive columns might reveal through correlation the value of some of the removed quasi-identifiers (background knowledge attack), or if all sensitive fields within the k-anonymous set have the same value, being present in the k group discloses the sensitive characteristic (homogeneity attack). To address this, the property of ℓ-diversity [104] states that every block of quasi-identifiers contained in a table must count exclusively ℓ well-represented values for each sensitive attribute. Several definitions of "well-represented" are formulated, including each value being unique, having a small enough entropy of all values across the group, and by minimising the number of times the most frequent sensitive value appears when compared to less frequent values (recursive (c, ℓ)-diversity).

Another standard privacy protection technique is Differential Privacy [58], which guarantees a data subject that the use of their data will not affect them adversely. Differential Privacy achieves this by not responding to all queries truthfully: the addition of Laplacian noise gives participants an element of plausible deniability, because some

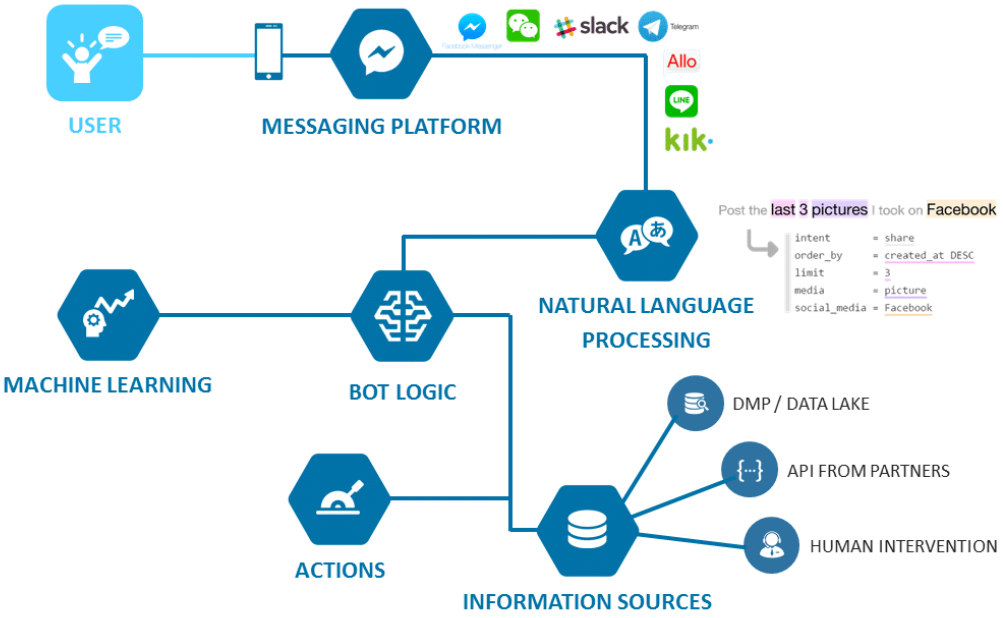

Figure 2.3: Architecture of a typical modern chatbot application

proportion of data will not be authentic. Differential Privacy algorithms are carefully parametrised so that it is known what proportion of information is being given. Inevitably, the trade-off between how much accurate information is being provided and the *privacy budget* means that, from an information theoretical perspective, we can not use differential privacy to establish completely accurate data analytics while guaranteeing strong privacy, just like generalisation and suppression in k-anonymity coarse grains and removes some amount of information from any model of the data.

While the described techniques can be used as to defend privacy, they are also metrics to objectively quantify how private one individual is. For a more comprehensive taxonomy on privacy metric, we refer to [206].

2.3.3 Privacy in chatbots

While the first prototypes of chatbots originate in the 1960s, widespread commercial solutions are a more recent invention. As such, the ecosystem has not yet reached the sufficient maturity where security and privacy are not considered but an afterthought.

We discussed a typical chatbot implementation in [108], as illustrated in figure 2.3. Between the different components, the chatbot provider can often be the more dangerous actor, as it has a full views on all activities of the user and chatbot developer, as well as potentially having gathered additional background information on the user, as described in [16], which lists what kind of information some popular Chatbot providers hold on the users of their platforms. Because the challenges in developing a custom-built chatbot stack are much greater than for traditional apps or websites, these providers are poised to obtain dominant market positions and hence obtain more data.

While massive data collection will accelerate the capabilities of these platforms, the fact that most of these chatbot providers are controlled by the same commercial entities in charge of large advertising companies is concerning from a privacy perspective, as their business interest are directly aligned with the maximisation of user data collection [76]. [10] argues that user perceive chatbots as not being private, similarly to how when calling a customer service hotline they are notified that they are called will be monitored and recorded for improving quality. However, our previous findings [108] are that users have mixed expectations on privacy based on their understanding of the technology, and assumptions on how chatbots actually work are varied. Additionally, participants in [219] confided in their chatbot agents because they felt they weren't being judged in their conversations. Similarly, [131] identifies how a computer program that seems to reveal sensitive information about itself (or its made-up persona) can trigger a process of reciprocal self-disclosure, making it easy for a well-designed bot to learn more about its user. The design of platforms surrounding the chatbot ecosystems will also influence how much information the users will be (unwittingly) led to reveal: as [10] points out, similarly to how mobile operating system opaquely request lifelong access to different kind of sensors and data, in the future as chatbots become to add more features, they might also require more access to user data; and while this could well be integrated into the conversation flow [77], it seems likely that chatbot provider would not make the process extremely clear, because they benefit from more sensor data. In a period where social media is plagued by political bots pushing for extreme political agendas, another concern is that users can clearly identify when they are engaged in conversation with a chatbot as opposed to a real human being; thus informed consent on whether users can be contacted by a bot and clear signals on when it is happening need to be a priority. Facebook is leading the pack with lack of transparency in when a user can be contacted by bots, by trialling a *Customer matching* programme where business can use phone numbers to automatically put in touch clients with their bots [64]; however, they are susceptible to popular outrage on the topic, and following the Cambridge Analytica media scandals, the entire chatbot platform was momentarily suspended due to security concerns [110]. More recently, from the experiences of the Babylon Hands on GP app (a virtual platform that combines a symptom recogniser chatbot with video chats with a clinician) it has become evident that uncontrolled adoption of chatbots can cause unintended effects in the physical word as well. Adopted by an NHS England clinic in Fulham, London, and now approved for deployment in Birmingham, usage of this app required patients to transfer their GP practice to the Fulham clinic. Over just two years this caused a massive migration of patients across the city, with the size of registrations increasing tenfold, redirecting funding from adjacent clinics, amid the concerns that sign-ups were closed to patients with more serious clinical conditions [94], and that the advice given by the bot are often wrong [136, 36].

[75] highlights what the tech industry has seen as unreconcilable differences in chat: privacy and intelligence. The history of secure instant messaging starts from the development of Off-the-record messaging (OTR), a perfect forward secrecy and repudiability add-on to other chat protocols, in 2004 [30], presented as an improvement over PGP, but still limited to one-to-one communications. In 2014, [61] presented a call to action by highlighting the security of different chat applications. Since then, most major centralised chat applications now advertise the capability of End-to-End

encryption, having reportedly adopted the Signal protocol developed by Open Whisper Systems for their open source Signal messaging application, or homebrew solution (like the Telegram and Wire applications). Implementations vary widely, from Whatsapp turning encryption on by default for its more than a billion users, to other large players like Telegram, Facebook Messenger, Google Allo and Microsoft Skype providing end-to-end encrypted channels as a private conversation feature when selected by the user. In this private conversation mode, use of chatbots is disabled; the only commercial chatbot platforms that provide end-to-end encrypted chatbots are Wire, and federated messaging protocol Matrix, but there has not been a large-scale, fully implemented bot platform built for either protocol yet. Apple iMessage provides by default end-to-end encrypted messages, but their Apps platform Business chat (not a traditional conversational interface but more of a small application embedded in a chat window) is stored in clear text on Apple servers, and even adds a customer service platform to the number of agents who receive the content of the message; the chatbot users are however pseudonymous to the chatbot developer and the customer service platform, being served a unique identifier from Apple.

Several chat services have been run on the Tor anonymity protocol (see section 2.2), but none have incorporated the use of chatbots. But while the original Tor Messenger [183] and the Ricochet protocol [34] have been discontinued, development for other messaging systems such as Cwtch [100] and Briar [164] are still active. Briar, in particular, is promising, as a decentralised and censorship resistant peer to peer protocol, which synchronises over Wi-Fi using Tor and Bluetooth. Briar already includes capabilities beyond messaging, such as forums, blogs and RSS feeds, and the project goals are to eventually become a platform for secure, distributed applications (which could include chatbots).

2.3.4 Privacy in medical data

For millennia, confidentiality of patient's information has been part and parcel of the medical profession. Even the Hippocratic Oath, a guideline for fifth century BC doctors that is still adopted by many countries today, enshrines the obligation of medical professionals to keep as *holy secret* whatever information they might gain about their patients.

[181] explains how, since it is necessary for a healthcare provider to collect as much Personal Health Information as possible to provide the patient with high quality of care, privacy is a fundamental economic asset for them. The establishment of fair and transparent privacy practices is fundamental in building trust with patients, which facilitates the flow of information. This entails a number of fundamental requirements a medical institution needs to ensure patient privacy: an appropriate infrastructure to keep patient information private, knowledge of law and practices, know how within the staff to take appropriate actions, and leadership, which are all enabled by training, policies and communication, and require periodic assessment of hospital operations on privacy.

Medical history is a particularly sensitive topic. A patient medical history leaking

might have concrete negative impact on their life (discrimination in hiring or within one's social circles) and cause distress and embarrassment when made aware of, but there are also more deontological harms caused by the mere loss of control that comes with leakage of information, even if it ends up not being used in other harmful ways [150].

The treatment of medical data, in some legislation like the USA, is regulated more strictly than other personal data. The Health Insurance Portability and Accountability Act (HIPAA) is a strict regulation that protects all Patient Health Information (PHI) through anonymisation and limiting data usage to be approved by Institutional Review Boards. Similarly, many laws exist [63, 65] to address discrimination based on health or genomic data. The effects of these laws are, however, limited, as they do not protect patients from data triangulation, are not consistently enforced [208]. Most regulations such as HIPAA only cover data collected from a specific set of institutions, thus excluding other relevant medical-adjacent information, such as internet search results for medication, or data tracked by a smart wearable device, but also credit history, apps used and location information (a *shadow health record* [151]).

Reducing the amount of data collection or sharing also has its harmful effect, by limiting the effectiveness of longitudinal studies, or public access to existing data from researchers in unrelated areas of the medical sciences [150]. Excessive freedom to self-censor medical history can cause harm to the patient and those around them [70]. The harsh penalty provisioned by these regulations are also another cause, beyond the current technical limitations, in the difficulty of transferring Electronic Health Records between providers of care [218], or even in matching individual health records with the correct patient [37], issues that cause an alarmingly high number of deaths every year.

Various options have been proposed for enabling wider portability in the United States [106]. A widely deployed solution adopted by many vendors internationally [147] to enable facility of data sharing is the Fast Healthcare Interoperability Resources (FHIR) standard [21], which includes a REST API for data sharing and interface elements for visualisation. The SMART open specifications [105] integrate with FHIR to enable developers of apps who need to use medical data, using open standard for authentication (OpenID connect [166]) and access control (OAuth2 [74]).

Chapter 3

Architecture

The Healthbot chatbot system [108] was one of the first multimedia chat-based nutritional interface, incorporating textual input with image recognition. To achieve the goals of the chatbot, numerous moving pieces were involved, and the prototype produced as outcome to the project did not take all security measure which are necessary when dealing with this kind of complexity. Healthbot is hosted on the Facebook Messenger platform, which provides a client application, forwards its messages to a webhook, and collects the responses from this cloud function to provide a reply to the user.
The webhook was a Google Dialogflow function, a natural language processing agent that takes user input, categorises it into intents, and handles each case with purpose-built code to generate a reply. Some intents that deal with more complex functionality are associated with another webhook, running on a free Heroku server instance from a Node.js application. Most of these will process some kind of data, either saving or retrieving information from a free MongoDB cloud instance, or querying image recognition or nutritional information APIs.

3.1 Threat Modelling of Healthbot

We conduct an assessment of potential threats to the Healthbot architecture through the use of Data Flow Diagrams (DFD) [6]. A DFD provides a visual representation of what data resources a system possesses, and where the data can be sent, along with an indication of the level of trust that can be attributed to it, on a simplified scale of High (green, or the +), medium (orange, or #), and low (red, or -). The taxonomy of DFDs includes:

- Processes, system code that performs a task or handles some data (represented as circles). These are the core components of the system, thus inherently trusted
- Data stores, a repository of data used for storage and retrieval (represented as text sandwiched by horizontal lines)

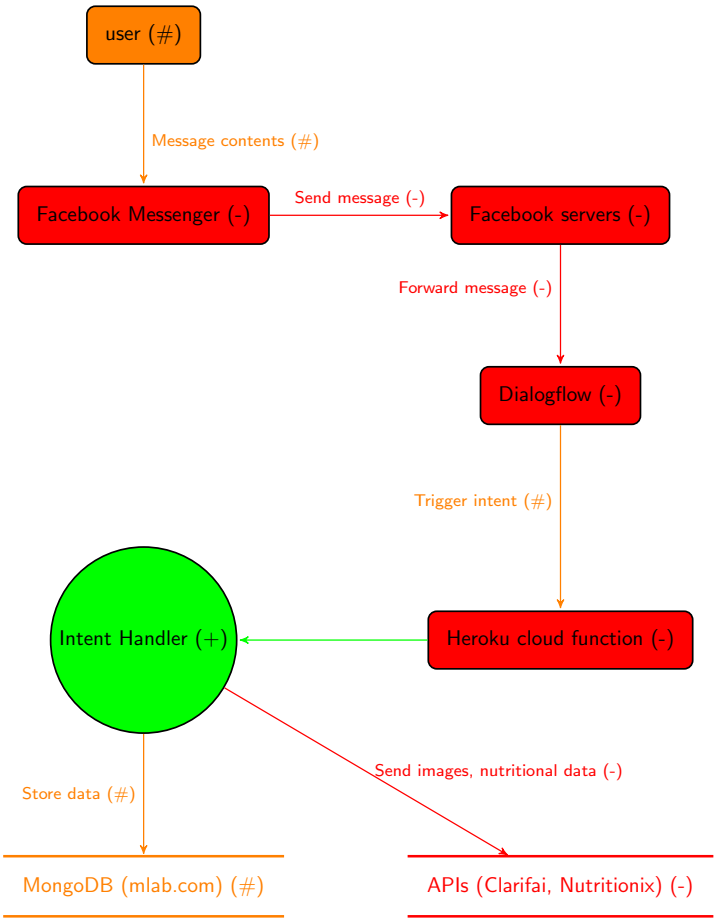

Figure 3.1: DFD diagram for Healthbot

- External Interactors, which act on data produced by the system (either as a sender or a receiver) while not being part of its codebase (rectangles in the diagram)
- Data flows between the various objects (represented as arrows pointing towards the data's destination).

Figure 3.1 shows the DFD for the Healthbot architecture. For each element of the graph, potential risks can be evaluated using the STRIDE methodology, to detect Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. In a semi-closed system such as an off-the-shelf chatbot built using commercial tools such as ours, most of these threats have been addressed by the vendor. However, the addition of external tools requires a further evaluation.

Information flowing between these several parties violates users' privacy (Information Disclosure) expectations in several ways. While there are no threats of a local attacker sniffing the content of communication, as all Facebook clients enforce transport-layer encryption, conversations with the chatbot are not end-to-end encrypted, making the entire plain-text of the chat available to Facebook. Similarly, all transcripts are also uploaded to Google servers for NLP. Since both companies' business models are based off personalised marketing, it is quite likely that this information will be associated with each users' dossier and used to improve their ad targeting. While it might be in-

nocuous and even beneficial to receive advertisement about food the users like, larger trends in the users' nutritional history could be analysed and sold to third parties, like insurance companies.
Also privy to the users' conversation history is the bot developer; while this might seem rather obvious, it might not be for the user, who expects conversation with an automated agent to remain confidential.
Other elements in the infrastructure also provide further weak points. The entire patient history is stored, unencrypted, in a cloud database (provided by mlab.com). Any access to this would reveal the entire nutritional history of every patient. Similarly, unauthorised access to the Heroku instance running our webhook could result in a malicious attacker siphoning off information as it arrives, or provide fake, harmful results. The danger of hosting such critical infrastructure on cloud platforms is double: not only can developer credentials be compromised, but the safety of the information is reliant on the security practices and honesty of the provider company. On the other hand, if the cloud providers are trusted, offloading infrastructure management to an experienced company will actually help reduce the developers' workload and actually improve overall security by limiting individual security tasks to a minimum.
Private information can also leak from the data provided to third-party APIs for nutritional values and image recognition. The risk factor here is definitely smaller than for the other components described, because all information is sent with the developer's API key, rather than per-user identifiers, so the providers cannot trace whose conversation requested information about which food item; however, images sent where not stripped of their metadata, which, by including information like location and camera manufacturer, tend to be quite crucial in identifying the provenance of an image.

At all steps of this process, information is stored unencrypted at rest, leaving it vulnerable to tampering. While malicious modification in the user information might be extremely damaging to the chatbot user, when receiving a recommendation, the incentives for any of the service providers to purposefully modify the data seem quite small compared to the reputation risks in the face of discovery, thus this class of attack seems less likely.

Denial of service attacks could cause to be quite damaging for this service, with potential loss of logged meals resulting in extreme distrust of the system, and thus inusitation. While the larger service provider will have appropriate defensive capabilities to fend off this kind of attack, a benefit of designing for scalability, the smaller API and storage services hosted on smaller providers are potentially at risk.

3.2 A distributed chatbot architecture

Despite the discussed threats by an adversarial component in the infrastructure described above, the use of cloud-based infrastructure gives us some distinct non-security advantages, by allowing to offload much of the computation to a server and its independently curated models. A local version of some of the tools used, such as the NLP and image recognition services, would provide less accuracy for increased computational and maintenance costs. The advantage of modern chatbot frameworks is that the

developer does not need to be an expert in Natural Language Processing to develop one (although expertise can help in making sure it is good).

We therefore propose a novel chatbot architecture, designed to maximise user privacy. The architecture removes the need for a separate cloud chatbot provider in favour of a client-centric model. The client, which is the program a user interacts with, will not be just a thin terminal like the Facebook messenger app, but will do some processing on the data. Queries will then be sent to a Natural Language Processor for intent recognition, entity extraction, any further parsing. The NLP server will send the intent and additional information extracted from parsing the sentence to the client, which will contain handler functions to trigger any further necessary behaviour. This might entail contacting other cloud services, or doing some local processing. We propose two alternative models, one centralised and one distributed, to handle data analysis functions.

3.2.1 Client

Having dispensed of our chatbot service provider, we need to adopt a new chatbot client. Our main requirement is that we are able to perform operations, such as encryption and input message filtering, on the text or images entered by the user. Additional features that are beneficial to chatbot adoptions are usability on mobile, and integration with other Instant Messaging services used. While there are several Desktop applications [174, 4] that satisfy the requirements, nothing fulfils that role on mobile. However, short of rewriting a new chat application (or modifying the internal of an existing one), we can take advantage of the chatbot features of the Matrix protocol.

Matrix [5] is a decentralised federated communication protocol introduced in 2014, which has been used for instant messaging, VoIP calling and Internet of Things automation. It aims to become an all-inclusive chat solution, by providing a bridge to many other chat protocols, and has seen widespread adoption across various communities, most recently having been forked by the French Government [111]. The Matrix architecture involves a federated group of homeservers, which communicate with clients through events, and can host rooms. Rooms hosted on one server can be joined by clients who are registered on different servers, and allow any number of users to communicate (a one-to-one communication between two users takes place in a private room where they are the only participants).

There exist several implementations of the Matrix specification for both servers and clients. Chatbots can be modelled as clients through existing APIs. They can join any room and be programmed to respond to incoming messages. We propose to model our private chatbot using this framework, by establishing a private room for each user participating in the protocol.

While the message data is not strictly being processed by the client itself, it can be made equivalent by running the chatbot process locally, through a separate service on the computer or phone, or in a trusted remote environment. The communication will still be handled on a Matrix homeserver, but since Matrix implements end-to-end

encryption, only the sender client and the chatbot application will be able to process content. While the end-to-end encryption capabilities of Matrix do not allow the homeserver to read the content of our chatbot messages, it will have access to metadata about the communication, such as the time we sent a message or received a reply. Once the chatbot program receives a message, they will conduct some simple preprocessing to filter it out of sensitive content. This might involve stripping some metadata from an image, which can reveal a lot of information about the user [72, 217], before sending it to an online version, or perform some linguistic substitutions before the message is sent to the NLP server.

3.2.2 NLP server

Since we are modelling the NLP server as adversarial, our goal is making sure that it is not able to reconstruct the message we are sending. To achieve this, we both try to anonymise the identity of the sender, and remove semantically significant content from the messages. This can be handled by expert-specific systems (which we describe later in the chapter). For the purposes of the NLP server, we are mostly interested in intent recognition and question answering, which do not require the server to have any domain-specific knowledge. Thus, we use the client to replace significant words with plausible substitutions that will not alter the meaning of the sentence significantly, but remove details specific to the sender. As a concrete example, going back to the purpose of Healthbot, as a diet tracking chatbot a user might send a message stating they ate some carrot soup. For the purposes of the NLP server, if the words "carrot soup" are replaced with the word "broccoli", there will be no difference for the intent matching task, since both terms refer to food. The client can thus select words that are considered sensitive, replace them before the message is sent, and keep trace of each substitution locally; when they receive a response from the server, they will be able to substitute back in the original words based on the substitutions. We describe linguistic transformation in more detail in Chapter 5.

For the complete functionality of our chatbot, we will not be able to use an off-the-shelf NLP server. If we were to implement the protocol, we would need to negotiate with an existing provider to integrate our additions. Unlike most current solutions, we will not expect to receive a complete textual response. Rather, intent handling rules will be defined between the client and server, so that once there is a match, the server can send a control message back to the client, which will prompt it to send back a message to the user. These control information might be parametrised, as appropriate, to take into account the contents of the user message. Additionally, if the server requires further response for clarification, they will send an appropriate control message to the client, which might either prompt the user to respond, or send the information itself if it had already been collected.

Our partner server will also need to support decryption of incoming messages. While an ideal defence against the server disclosing information would be using technologies such as Fully Homomorphic Encryption or Secure Multiparty computation, to allow the server to process messages without being given the plaintext, these crypto-

graphic schemes are still very slow, and thus unsuitable for a real-time problem such as chat. Thus, we will need the server to decrypt our messages on reception. We designed a transport protocol, based on Onion Routing and Crowds, to anonymously send our messages through replication between several users' chatbot clients (described in Chapter 4). Messages coming into the protocol will be concatenated with ephemeral public keys, generated by the client, and padding to make the message reach a standard size. This packet will be encrypted using the server's public key, and sent off through the routing protocol. Ephemeral keys are normally maintained for a single message; however, if the server tries to receive a response to the original query, this will also be attached to the same key. Since the malicious server would try to indefinitely extend the survival of a key for better profiling, it is the client that is in charge of determining how long to maintain a conversation using the same key, based on the matching intent. Once a new key is used, the identity of the client to the server is renewed. If a server requires some background information on the user once the key is reset, they will be able to request it through control messages, to which the client might reply with real or by substituting with equivalent information. We now describe the two options the chatbot might use for backend data holding and its "intelligence".

3.2.3 Backend server

The easiest option for a backend is having a centralised chatbot application server, just like in Healthbot. This application server would be contacted by the chatbot client and send encrypted information, based on the parsing and entity extraction conducted by the NLP server. The purposes of the backend server, besides long term information storage, are the analysis of the collected data on a single individual, and large scale analytics conducted on the collective dataset from all users through a central aggregator service.

Because the chatbot needs to track and analyse historical data entry for an individual user, it would not be possible to adopt the mitigations discussed for the NLP server. Therefore, it is necessary to model the server as a trusted, or at least partially trusted entity. For chatbots dealing with medical data, analysis of other aspects of a patient's medical history might also offer some advantages for better inferences. It seems worth taking advantage of an existing Electronic Health Record. As discussed, one of the more popular EHR standards is FHIR, and its extension SMART. FHIR allows the collection of a variety of information about patients from different sources. The FHIR standard is not opinionated about security in itself, but encourages implementation to safeguard patient data. It provides resource labels at different granularities to enable fine-grained access control and determine the importance of integrity, to be implemented by the developer. SMART build on this functionality to allow access to data hosted on the EHR to external applications. It manages users authentication using the OpenID federated identity standard, and authorisation through the OAuth2 public standard. Based on the OAuth2 classification, apps running on mobile phones are considered *public* rather than *confidential* (as they can not be provisioned with a secret at installation time that will be secure against attackers on the same platform), and *standalone* (access to resources start from a separate app from outside the EHR

interface).

Our chatbot provides some data to the EHR under the assumption of trustworthiness, in that the FHIR server will not leak or tamper with provided health data, and will only share it with authorised applications. The OpenID Heart working group [161] has set to develop a standard that will give greater control to the user on what services are able to access their data, across different institutions and applications. However, if we choose to trust the EHR, that does not mean the EHR trusts us. OAuth2 defines the need for a separate authorisation server, which mediates access requests to any FHIR resources. When launching a standalone application on SMART, it will include a request for what context - the resources the application is trying to access - are needed by the app. Authorisation might require user interaction, which could be integrated in the chatbot by displaying a webview as part of the conversation. This will be recorded by the authorisation server, so that future requests to resources limited by access control might be fulfilled.

Despite these assurances, a chatbot developer might choose to model the EHR as adversarial. Rather than sending patient data as collected for storage, they will obfuscate the data using Differential Privacy. While this will reduce the quality of predictions, because some level of specificity of the data is lost, it will prevent leakage from being particularly dangerous, because the distribution of each user data will be altered to be statistically similar (within the parameters set for differential privacy). Alternatively, the full data might be uploaded for individual user statistics, and scope-limited for user-only access, and a modified differential-private version made accessible to the aggregator service to compute on large scale data analytics for all users.

3.2.4 Federated backend

The previously described backend model, while convenient to implement, puts a significant amount of trust in the centralised FHIR server. Although users will probably trust their medical institution to take care of their data and respect their privacy (an expectation that is not always reasonable [73]), the combined health records of many different users will provide a valuable target for hackers [41]. To limit the attack surface and remain in control of where their health data is stored, the chatbot client should be able to keep the data collected locally. In recent years, with mobile device specifications becoming increasingly powerful and techniques like transfer learning [198] and model distillation [80] increasing in prominence, it has become possible to quickly run advanced machine learning models on device in relatively short times [83]. But if this allows the client to compute data on the users' own information, we now lack the centralised server's ability to draw upon other users' data to compute population scale statistics.

A first step in this direction has been achieved through Federated Learning. First proposed in [119], Federated Learning proposes a Gradient Descent based model in which training data is kept on edge devices, such as mobile phones. While each device trains and maintains a local model, to which it applies local weight batch updates, a central model still collects all user weight updates, performs a weighted sum to apply to a

global model, and redistributes it to all users. Various optimisations on the learning updates can be applied to handle efficiency, crucial when the quantities of data analysed are large or the training algorithms are computationally intensive [96].

Federated learning would address the weakness in our previous model of relying on a trusted EHR service. By having a centralised server which never receives any user data, the risk of a malicious health provider or authorisation errors would be minimised. There are however still privacy concerns about the use of these technologies, as it is still possible to reconstruct some of the input data from the weight changes to the model [120]. It would be possible for a malicious EHR to intercept the weight update to each user and "reverse engineer" the training data that must have generated them. Furthermore, it has been demonstrated that federated learning is much more vulnerable to model poisoning (the insertion of malicious training data to force the model to produce incorrect results) by a single [26] or multiple colluding agents [14], since the server can not validate that the training data is not malicious. The addition of differential privacy to the model [118] still leaves the model vulnerable to such an attack, although the threshold of compromised nodes necessary to permanently backdoor a model increases from 1% to 5% .

[19] proposes an alternative solution in a fully decentralised, peer to peer learning model for classification. Each peer in the protocol broadcasts their local model, scrambled using differential privacy, to a few neighbours, and train a randomised block coordinate descent algorithm to converge linearly with the number of broadcasted updates [215]. Further efficiency improvements can be achieved by switching to a leader / follower pattern where a set leader nodes, who have better performance, guide the remaining follower nodes to accelerate convergence [45]. Because of differential privacy, communications are hidden from any attacker, preventing information leak even in the face of all peers in the network colluding. This solution slots in well within our architecture, as we already make use of peer chatbot clients to send information to the NLP server. Since the field of peer to peer federated learning is still quite novel, it is uncertain whether it might be possible to scale this technique up to more complex machine learning tasks. However, considering that deep learning models suffer when it comes to explainability [167], a quality highly prised in data analysis of healthcare data [8], it might not be a necessary requirement for a successful healthcare chatbots.

Further analysis will also be needed on how decentralisation affects model poisoning, since each peer will directly communicate with a fraction of total participants. A potential solution to verify peer's honesty would be randomly requesting, every so often, a Zero-knowledge proofs [29] that a weight update sent as part of the learning process is actually based on honest training results. Removal of a centralised server also shifts the duty of backing up to the user. For our purposes, each chatbot client need to keep a full backup of the conversation, per-user data model, and encryption keys. This might be automated by storing data within the Matrix room, but the user will still have to deal with key management.

3.3 Updated Threat model

Our novel architecture iterates on the Healthbot design to prevent much of its *Information disclosure* risks. As in the previous model, the user is not fully trusted because of the risk of naively revealing too much information. After the client receives the message or image, it is possible to properly cleanup sensitive information through the linguistic transformation and metadata removal processes, making the final encrypted message trusted.

Having limited the capabilities of the server to identify users or the contents of their communications, the most likely threat of information disclosure is the backend. Specific care needs to be taken in hosting the service on an EHR provider which has been audited to properly implement the FHIR standard, and take appropriate security measures to defend their data, for instance from *Elevation of Privilege* by other apps' users. Likewise, the OAuth server needs to be properly secured to adequately handle malicious requests from other applications hosted on the EHR, and the chatbot app developer needs to set up sufficiently strict access control labels depending on the sensitivity of the data.

As we stop relying on the infrastructure provided by large tech companies, our system becomes more brittle to *Denial of Service*. Our architecture relies on the availability of the Matrix Homeserver, the EHR and authentication servers, the NLP provider, and having enough Tor nodes and other chatbot users open to communication. While all of these actors have reputational incentives in maintaining good up-time, an attacker compromising any of them could stop the protocol in its track. Additionally, the nature of our routing protocol (described in more detail in Chapter 4) entails increasing the traffic through the Tor network and the NLP server by several times the number of messages sent. If the chatbot protocol ended up being adapted at large enough scale, perhaps through a Sybil attack, it could disrupt communications severely, slowing down responsiveness of the application. Both of these issues would be problematic, and potentially cause users to stop using the service, since responsiveness is a highly valued quality in chatbots [135]. On the other hand, the decentralised nature of the protocol ensures that our application can not be suspended on a whim by a service provider we rely on.

The chatbot architecture also addresses, to some extent, risks of *tampering*. Although communication in Healthbot was encrypted and authenticated during transport, we have to trust that the Facebook Messenger cryptographic schemes are secure, with no way of verifying it [171], and after the message leaves our server, we have no control on how the other providers handle our data, and whether they are able to modify it. By encrypting and signing all communications we control from the client itself, even through several layers when employing onion routing, we can ensure that any information we send will not be modified before it can reach its destination. We should again be concerned about application running on the FHIR server, as they might serve the centralised learning model, or update the client, with adversarial machine learning inputs, as discussed in section 3.2.4.

In our threat analysis we assume that the chatbot developer is acting in the users' inter-

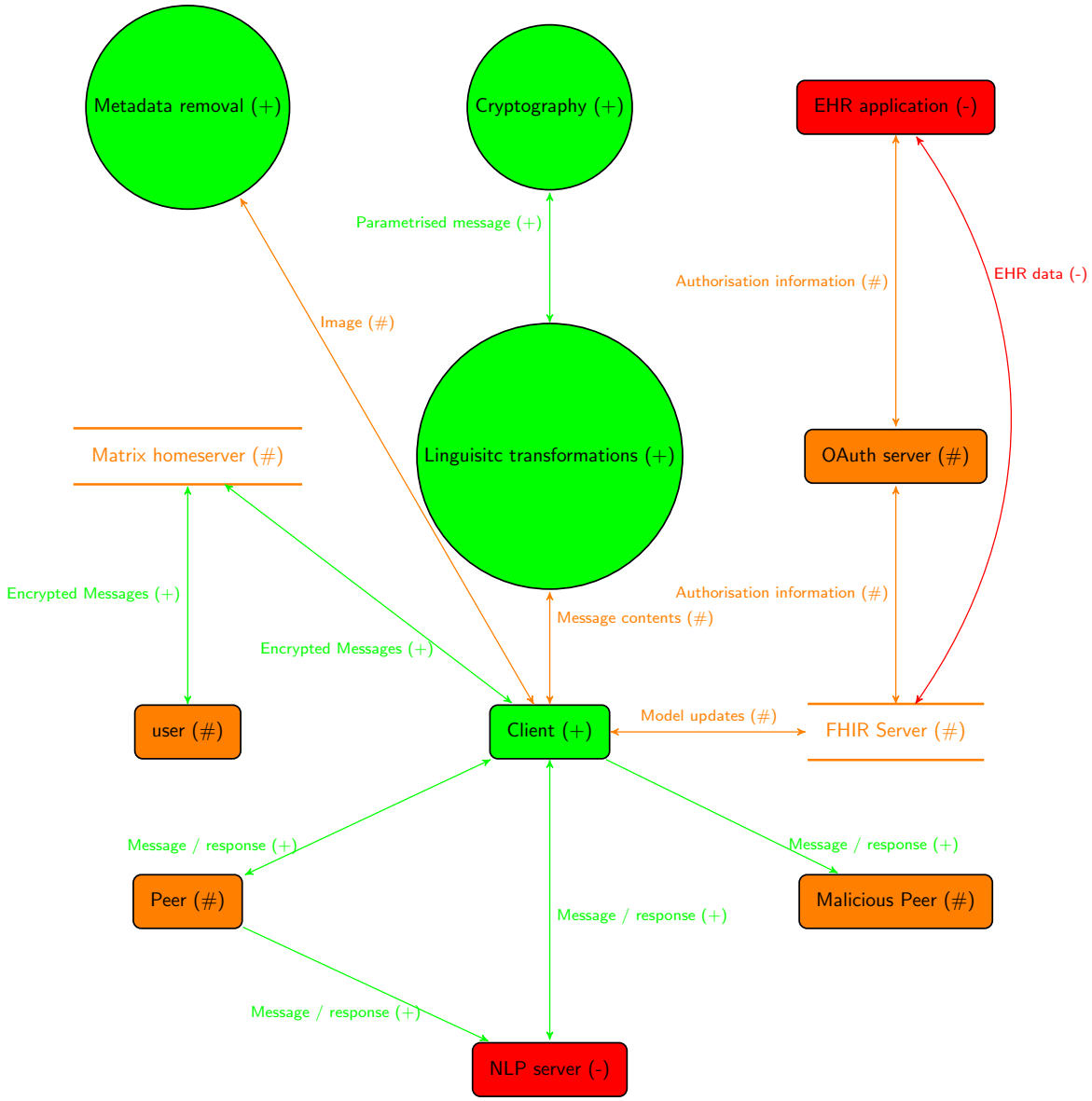

Figure 3.2: DFD diagram for the anonymous chatbot protocol. While this version shows a centralised backend, the fully decentralised protocol has a similar structure, but with the removal of EHR application, OAuth Server and FHIR server entities, and model parameters being sent to the peers

est. This does of course not have to be true, but our requirement of the client software running on a user controlled device allows any malicious activity to be detected by auditing the running software. To favour this, the chatbot architecture should force conversation scripts to be available to the user in a plain text format.

Chapter 4

Routing Protocol

This chapter discusses the routing algorithm for the anonymous chatbot protocol described in the previous chapter, and its properties, including a threat analysis in terms of user anonymity. Anonymity is the best way to guarantee that the message is private from the Natural Language Processing server, because messages need to be necessarily sent unencrypted, so the content being legible can only harm the sender if the adversary can tell who they are. Since the content of the anonymous messages could still disclose the identity of the sender, either through explicit information contained in the text, or implicitly through signalling, we discuss in the next chapter some mitigations our architecture might employ.

4.1 Formalising Anonymity

[145] outlines informal definitions for various concepts in anonymous networks. For our purposes we are interested in Sender Anonymity, and Sender Unlinkability. Other properties, such as relationship unlinkability or receiver anonymity do not apply to our protocol, since we have a unique well-known receiver which all clients will be talking to. These properties were formalised by [12] in their AnoA framework for anonymous communication. In AnoA, inputs to the protocol are modelled as a table, each row representing a list of successive messages sent from a Sender to a Receiver, along with some auxiliary information. For a probabilistic adversary bound in polynomial time (PTT) who can choose input tables that only differ in one row whose only distinguishing feature is a different sender (the challenge row), a protocol achieves indistinguishability-based Computational Differential Privacy if the adversary fails to distinguish the two tables. This is a stronger assumption than in [124], where there are no restrictions on the Challenge Row, because if the message is distinct for the two senders, it might already provide some identifying information. Given a challenger oracle $CH(\mathcal{P}, \alpha, b)$ for protocol $\mathcal{P}$, adjacency function α and that adversary $\mathcal{A}$ can call on to select a response 0 or 1 to choose between two input tables, $(\varepsilon, \delta) - \alpha$-IND-CDP is defined as

$$Pr[b = 0 : b \leftarrow \mathcal{A}^{\mathcal{P},\alpha,0}] \leq e^{\varepsilon} \cdot Pr[b = 0 : b \leftarrow \mathcal{A}^{\mathcal{P},\alpha,1}] + \delta$$

The adjacency function takes two tables and returns either a error token or a modified version of the tables, and varies based on the anonymity property analysed. The challenger oracle, upon receiving the message tables D_0, D_1, will run $(D'_0, D'_1) \leftarrow \alpha(D_0, D_1)$ and execute $\mathcal{P}$ on D_b, forwarding all the messages it outputs to $\mathcal{A}$.

In this model, the adjacency function for sender anonymity returns an error for all cases where the sender of the challenge rows are not the same, and the original tables otherwise. For sender unlinkability, the adjacency function will randomly select one of two senders for either challenge row. In Tor, a partially global passive adversary is able to break the two properties only for the distinguishing event $\mathcal{D}_\alpha$ where they control the Tor circuit's entry node, for the sender's message and for both messages in the challenge row respectively [190]. For a network of n nodes where k are compromised by the adversary, if the probability of any node being selected as the start of a circuit is uniform, the probability of the distinguishing event for sender anonymity of the entry node being compromised is $Pr[\mathcal{D}_{SA}] = 1 - \frac{\binom{n-1}{k}}{\binom{n}{k}} = \frac{k}{n}$. For sender unlinkability the distinguishing event involves two Tor nodes being broken, so $Pr[\mathcal{D}_{SU}] = Pr[\mathcal{D}_{SA}] \cdot Pr[\mathcal{D}_{SA}] = (\frac{k}{n})^2$.

Our strong adversary can issue the same message multiple times, changing circuit for each message until the entry node is one of the k corrupted ones. To prevent this family of attack, Tor circuits use entry guards, a set of nodes chosen for their bandwidth availability to serve as entry points for all circuits a user establishes [59]. A list of entry guards is kept by the client for a long duration of time, so that in that period, if no malicious node is chosen as a guard, there will be no risk of being compromised. However, if a node chosen as entry node is corrupted, there will be a high probability to pick a circuit with a corrupted entry point for the duration of that entry guard cycle. If the set of entry guard nodes has size m, the probability of the distinguishing event for sender anonymity becomes $1 - \frac{\binom{n-m}{k}}{\binom{n}{k}}$. If entry guards are replaced after a maximum of l sessions, and the user initiates d session, the probability gets bounded by $1 - \frac{\binom{n-\lceil \frac{d}{l} \rceil m}{k}}{\binom{n}{k}}$.

4.2 A more anonymous chat routing protocol

Unlike the AnoA adversary as modelled in the previous section, chatbot users in our protocol do not have the benefit of undecipherability when going against the Natural Processing server as an adversary. It is possible than that a message may contain identifying information about whom the sender is. While some more egregious data could be scrubbed on the client-side using automated filters, over time the server might build up enough information about what kind of messages it receives to identify who a user is from contextual information.

Our proposed solution adapts the Crowds anonymity protocol [160], replicating each message from several nodes and sending its copies in a shuffled order. Unlike Crowds, messages will not be just forwarded to a different peer, but will duplicated as well, so that the adversary will receive multiple copies of the same message. We also improve

over Crowds by transporting each peer to peer communication through Onion Routing, which prevents, in most cases, local peers from knowing who they talked to, and active attacks through the integrity properties of Onion Routing. The addition of artificial delays to transport times protects the protocol against timing attacks. We consider a scenario of a powerful global adversary. In reality the Natural Language Processing server would most likely be a local, passive adversary, but we consider the case where it has been compromised by another, more powerful active adversary with a larger view of the network. We assume that the adversary is merely curious and its responses to the chatbot will not be malicious; if the server was not trying to fulfil its primary purpose, a sophisticated client application might detect it, and compromise the reputation of its service. Additionally, the adversary might be colluding with both c chat clients in the protocol and k Tor nodes that are not participating in the chat protocol but are merely part of our transport layer. Later, we will also address the possibility of more powerful adversaries.

Our solution relies on four parameters, δ, δ', ρ, ρ'. We illustrate its behaviour with an example, describing a conversation (an exchange of messages) between a user Alice and her chatbot.

4.2.1 Description

User Alice wants to send a message to the chatbot. She has a long term master key she can use to generate ephemeral public-private key pairs; she generates one such and appends the public key to the message. Alice then encrypts her message+key pair using the server's public key. After waiting for some time in the range 0–δ, it sends the message to the NLP server using a newly established Tor circuit. Concurrently, it also chooses to send the message to another chat client, an event we call χ (equivalent to Crowds' p_f), with probability ρ. The client receives the message, which it won't be able to read (since it's encrypted with the server's private key), stores it, and forwards it through a new Tor circuit to the server, after a delay between 0 and δ milliseconds. It might also choose to forward it to yet another client with probability ρ. Having received the forwarded message, this different client will also forward it, after a delay of 0 and δ milliseconds, to a new client, and also send it to the server. The message will keep getting forwarded between chat clients who also send it to the server, until event $\neg\chi$ happens.

The server, having received all these messages, will run for each one intent parsing, generate a parametrised response, encrypt it to the attached ephemeral public key, store the triple (plain text, public key, response object), and send it back through the same Tor circuit from which it was sent. Any peer who receives the response, other than Alice with her private key, will not be able to decrypt it. If the server requires a follow-up response, Alice's client can send another message using the same public key, if it deems the request warrants one; otherwise, her next message will use a new ephemeral key. Every message Alice sends runs through a new circuit, and the order of peers each client sends messages to is chosen randomly. For every message received by a peer, after waiting an arbitrary amount of time smaller than δ', which is chosen to be

much larger than δ, it sends the message to the server a second time, with probability ρ', typically smaller than ρ, through a different Tor circuit than the first time. By this point the server might have already sent a response, so it will retrieve the stored triple and send it along this new route, since it can't know if this is in fact the original sender.

4.2.2 Analysis

In the described protocol, the NLP server can only identify the source of an individual message by breaking sender anonymity, by corrupting both the entry and exit nodes in a Tor circuitr. Alternatively, in the case where the only traffic sent from the client's IP address goes to the chatbot, it might be possible to drop all traffic passing through their node, to force the establishment of a new circuit (Selective Denial of Service attack) [9]. Additionally, it must break sender unlinkability, to be able to distinguish whether the sender had been replicating a message sent from a different peer, or if it was the original sender, giving the user a degree of plausible deniability (even if the first node used by Alice is corrupted, she can just lie saying she received the message from another user). The random probability in distributing the packet to a peer means that a colluding peer will not know at what stage in the protocol they were to receive the message, and adding a delay before sending the message to the server ensures that the originator message is not always the first to be received. Additionally, neither the server nor any of the peers can be provided with a full list of all clients active at any one point, because of the decentralised nature of the protocol. Clients will learn about the location of other participants to forward their message to by using a gossip protocol [27].

A rational adversary would stop the propagation of a message from being forwarded by one of the clients they controlled. Assuming that is the case, for a Tor network with n nodes and g peers, an adversary that corrupts k nodes and c peers will have probability of a single peer receiving a message from another peer $P(\chi, \neg A) = \rho \cdot (1 - \frac{c}{g})$, where A is the event where the sending node is controlled by the adversary and chooses not to forward the message. Given the independence of χ and A, as a negative binomial distribution, the probability of a message being forwarded b times is thus

$$P(\chi, \neg A)^b (P(\neg\chi) + P(\chi, A)) = (\rho \cdot (1 - \frac{c}{g}))^b (1 - \rho + \rho\frac{c}{g})$$

This count will additionally grow by i for binomial distribution

$$\binom{b}{i} (\rho' \cdot 1 - \frac{c}{g})^i (1 - \rho' + \rho'\frac{c}{g})^{b-i}$$

For an adversary to learn the identity of the original senders, it is necessary that $b = 1$ and $i < 2$, and that for all such connections both the entry and exit nodes are corrupt. Thus, the probability of detection is

$$((\frac{k}{n})^2 (\rho \cdot (1 - \frac{c}{g}))(1 - \rho + \rho\frac{c}{g})) \cdot (((\frac{k}{n})^2 (\rho' \cdot 1 - \frac{c}{g})) + (1 - \rho' + \rho'\frac{c}{g}))$$

4.2.3 Issues

Given its odds as described above, the adversary can increase its probability of detection by initiating a Sybil attack, by adding new nodes to the Tor network or creating fake chat client programs. Since only centralised systems can perfectly defend from Sybil attacks [57], Tor and our protocol are both vulnerable. However, a large scale Sybil attack is expensive, since running a decoy Tor node has severe computational constraints: all nodes need to be running at full CPU and network utilisation, it takes several days before they are flagged as stable, and they need to be placed in separate $\backslash 16$ network ranges to be included within the same circuit. The *sybilhunter* program [213] uses features such as network churn, server uptime, fingerprinting on the onion service relay address, and nearest neighbourhood classification, to detect suspicious activity that might indicate such an attack. To defend from the addition of attacker-controlled chatbot clients, it could be possible to add a form of computational constraint to our protocol, whereby the addition of a new peer would require some cost, trivial for one user but expensive at scale. This could be a network or computational cost, such as solving a cryptographic problem similar to Bitcoin's proof of work [128], or a monetary one. We could also attempt to authenticate chatbot users as being humans, by presenting them with a challenge similar to CAPTCHA [205] at signup. This should be administered periodically, as to prevent the attack from subverting already authenticated clients.

Our protocol favours anonymity at the cost of redundancy: each message can be repeated multiple times, and for each message replication a new Tor circuit needs to be established. From a network perspective, the cost could probably be shouldered, since the latest reports [195] put the current network capacity of the Tor network at three times its current traffic. Even if the addition of this new chat application became highly used, causing the amount of traffic of the Tor network to increase significantly (which would require a very high number of users, since most communication would be text, small in size compared to images or media), and even if the level of Tor nodes remained constant (which would go against the current growth trends [9]), there are many simple measures developed, as part of the many existing network protocol that we could adopt, to reduce congestion, an issue that the Tor protocol does not prevent by itself [122].

The bigger bottleneck will then be the NLP server. If the chatbot obtains enough users, replicating the message several times would make the server busier than it expects. In this case, it might be necessary for clients to send a small probe to the server to check if the message queue is too busy, before sending a message that might get dropped. Modern cloud computing architectures use a server-less model, where they rely on a large cloud service vendor to provide as much infrastructure as they require to scale the product; a NLP provider using this kind of architecture should then be able to handle an increase in network traffic, from a technical perspective. From the economics side however, it is not certain that there are incentives to provide the service, since the cost of a higher capacity cloud function would increase and the amount of interesting (for-profit) data mining is already smaller than in a typical chatbot. As with many existing privacy-first companies, this might be addressed by providing the service provider of-

fering a more limited free tier, and more expensive paid options for large vendors, in the hope of falling in the niche market area. Alternatively, if this protocol or something similar was standardised by an international standard setting body, there could be a legislative effort to force vendors to provide their services in compliance with the protocol, especially when dealing with sensitive medical data.

In addition to adversaries who control Tor relays, a class of attack might be led by network adversaries, Autonomous Systems (ASes) or Internet eXchange Providers (IXPs). [179] shows that these adversaries have a significant chance to compromise a Tor user. While there are some proposals that might limit their impact [84], our protocol still requires an attacker to also control a significant portion of chat client nodes, since deanonymising the user still provides plausible deniability that their message was not the first being sent.

For a global adversary, who considers the Tor network as a black box from which it observes all inputs and outputs, if the only network traffic going through clients is messages related to the protocol, it becomes trivial to identify the original message sender, since they will be the first to send a request to the network, either for sending the message to the NLP server or to forward it to another node, without having received an incoming message in recent times. That kind of behaviour could be attributed to a delayed message sending, but the probability of that happening is much lower. If the network is very active, the same client might be selected as a message forwarder by other users even while producing its own message, therefore hiding its original outgoing traffic between that of another client. As the network scales, it will be necessary for a certain number of messages per user to be sent near-simultaneously, to appropriately hide the traffic. As chatbots can suffer from engagement dips due to small attention spans of users, it might also be necessary to enlist other networked functionalities of the chatbot to go through the Tor network. For instance, it might be ideal to request model updates from other clients some point before a message is sent (when the user opens the client or starts typing a long message). The Peer-to-peer communication created through the various nodes, combined with the time delays added to the message, might be able to hide from a timing analysis the source of the message. This will also make sender anonymity useless for the partially global adversary we have analysed so far, since the data coming in from the peer could be addressed to any service and not necessarily the NLP server; it will thus be necessary to break relationship anonymity as well.

Chapter 5

Natural language alterations

We have described how our theoretical chatbot protocol would be able to send messages anonymously to an adversarial Natural Language Processing server, without being identified on a protocol level. Any measures we have taken would be easily defeated if the adversary could guess our identity from the message itself. In this chapter, after some background on the relevant topics in the Natural Language Processing literature, we describe both what kind of processing an adversarial server might conduct, and mitigations we could apply as part of our chatbot.

5.1 Background

The development of chatbots is rooted in the history of Natural Language Processing (NLP). For a background on the field and recent developments, we refer the reader to our first report [108].

Many solutions of NLP problems require the use of Language Model, a probability distribution over all sequences of words in a sentence. Traditional language models have been based on *n-grams*, counts of all sequences of n words appearing in the language corpus [48]. More recently, neural network based models have been trained as probabilistic classifiers [22]. A particular class of Neural Networks that has proven particularly apt for the task is Recurrent Neural Networks (RNN), and their Long short-term memory (LSTM) implementation in particular [89]. These networks efficiently process sequences by keeping a record of training data they have already processed and feeding it as an input to successive phases of training, alongside the new words in the sequence.

Advances in transfer learning in the last year promise to push the boundaries of the state of the art for language modelling. We begin by describing how recent models can be used for pretraining, and follow with specific relevant tasks.

5.1.1 Pretraining Language models

Since the development of the Word2vec [123] model and the GloVe algorithm [143], word embeddings (vector representation of word features) have been used to perform "shallow" learning on Neural models. Only in 2018, several concurrent developments came to fruition. ELMo [144] is a bidirectional LSTM language model which was used contextualise word embeddings based on the sentence they were in. ULMFiT [82] adopted fine-tuning on a task specific corpus as an intermediate step between pretraining and the actual task. The OpenAI Transformer (GPT [154] and the infamous [153] GPT-2 [155]) adapted the decoder from a popular machine translation architecture [153], which performed better than LSTMs in handling long term dependencies, to learn language modelling tasks, performed through specific input transformations.

Finally, BERT [52] closed the gap between these models by adding bidirectional encoding to the transformer architecture. The BERT pretrained models (the base model comparable to the OpenAI transformer in size, and the 3 times bigger $BERT_{LARGE}$) are trained by artificially replacing 15% of words in the input with a mask or an alternative word (an adaption of the Cloze task described in section 5.1.3), to avoid training to be conditioned by the rest of the sentence. It can be optionally fine-tuned to a specific task. BERT-based models were found to perform well in most language tasks, breaking the state-of-the-art results for tasks in question understanding and answering, classification, language inference, sentiment analysis, acceptability, semantic similarity, paraphrasing, and recognising entailment.

5.1.2 Named Entity Recognition

The Named Entity Recognition (NER) task in NLP requires automatically tagging the components of a sentence in unstructured text as belonging to a certain class.

Classically, NER is seen as a sequential prediction model, and a popular solution has been the use of Conditional Random Fields (CRFs) [97], a further improvement over the precedent Hidden Markov Models [152]. CRFs define a set of feature functions, based on linguistic properties of the sentence, and assign a score to each word by taking the weighted sum of each function. Feature functions take into account the position within a sequence and the current label of the current and previous word. Gradient descent can be used to learn the optimal weights for each function, and from there an optimal labelling of the sentence. [158] improve over the classic algorithms by refining the sentence tagging to include multi-token chunks, and integrating contextual features aggregation (large windows of tokens surrounding a word). Their initial prediction on the corpus was then fed back into the system for another pass, which took into account the history of predictions up to that point.

More recently, a variety of Neural architectures, mostly based on LSTMs, have pushed the performance on this task [103]. But even the best results, using ELMO, are outperformed by a simple model trained on BERT [88].

5.1.3 Word replacement

The task of lexical substitution is that of selecting a word that can act as a replacement for another one, without altering the meaning of the sentence, originally defined as an intermediate step for Word Sense disambiguation [112]. A simple lexical substitution method might involve the use of a thesaurus, such as WordNet, to substitute the target word with a synonym or a hypernym. However, this might not always result in a successful substitution, and likewise specific sentences might justify the use of a word that is not normally associated with the target, based on the construction of the thesaurus [191]. Participants of the SemEval 2007 challenge [113] all used systems based on n-gram distribution counts from large corpora [114], which achieved best performances of no better than 20%. Later, [121] used skip-grams (an extension of n-gram models for non-consecutive words) to create a contextual embedding for word2vec, which was used as a substitutability metric to assess replacements.

A parallel task is the Cloze test, (first described in [192]), an exercise for replacing a blank space in a sentence with a fitting word or expression, as a measure of readability of a sentence in humans. It has been suggested human solve the Cloze task by applying something akin to an n-gram probability model [177]. The Cloze test is not a common task for NLP researchers to attempt, although Cloze-style datasets are often used as the basis for metrics of sentence or story comprehension [173]. While the task might be solved by applying a language model, there is an additional subtlety in that the word that needs to be guessed is not at the end, but in the middle of a sentence, making the use of bi-directional language model appropriate [175].

Usage of the Cloze test was adopted in BERT, under the name of masked language model, to prevent the bidirectional language transformer from overfitting on the words of the training sequences. In this setting, 12% of all training inputs were replaced with the *[MASKED]* token, while 3% were substituted with a simple word. The original BERT training was run over 10 million epochs and achieved an accuracy of 84%. Although evaluations of BERT as a language model are sparse, it is found to score well under diversity as a Markov Random Field model, if not in quality [209].

5.2 Attacker model

The NLP server is a very powerful adversary, in that it stores numerous messages from many users, potentially across many applications. The more the service is used, the more information it can gather. Having available state-of-the-art natural language tools, it will be able to conduct large scale data analysis on the collected data.

Modelling attacks from an adversary depend on what its goals are. While our protocol is trying to defend the sender's anonymity, our ultimate goal is not revealing information about them to preserve privacy. If the adversary wants to build a complete history of all data a user sent to the model, they will try to use author identification to determine the provenance of each message. A first step in doing that might be author attribution to limit the pool of possible senders. This task might also constitute

the attack for an active adversary, who wants to send personalised ads to the chatbot user, and is willing to modify its responses to include marketing messages, explicitly or subliminally, based on the user's demographics and personality attributes.

Identifying users from chatbot messages is not an easy task. Online documents are generally short and poorly written [32] there is no agreement among researchers regarding which features yield the best results; in fact, more features might be a waste of resources because they provide little useful information.

One obvious way for the attacker to gain information is the user explicitly stating it. Since the NLP server is trying to learn intents, it will be able to recognise whether a slot contains deanonymising information, or it can be used to reconstruct some. Users sending sensitive information under the impression that they will be anonymous are not guaranteed to stay so, considering how the server might have access to another resource to correlate the information entered. For instance, a server seeking to identify the dietary records of users might fingerprint them by linking their anonymous food logging messages with their credit card activity or their supermarket loyalty cards.

5.2.1 Stylometry

Another risk facing anonymous chatbot users is stylometry, the practice of quantitatively analysing the style of a text by describing its linguistic properties. While no single scholar can be attributed with the development of stylometry [81], the first notable case was Wincenty Lutosławski's chronology of Plato's dialogues [141]. Later, computer stylometry can be traced to [126]'s work on the Federalist Papers. The main tasks within stylometry are authorship attribution, authorship verification, authorship profiling, stylochronometry (the study of changes in style over time), and adversarial stylometry [132]

While stylometric techniques mentioned can be used to improve security, such as [33]'s proposal to use authorship verification as a continuous authentication method, there are valid concerns that they might lead to the deanonymisation of private messages, or automatic profiling. The FBI even suggests [50] classifying writing style as a biometric characteristic, and stylometry has been used in US and UK courts as evidence [43, 23]

Recent attention has been given to the application of stylometry to the study of online content for authorship verification [163], attribution and characterisation [90], and instant messaging in particular. [137] use " syntactic and structural layout traits, patterns of vocabulary usage, unusual language usage, and stylistic features" to uniquely identify (with over 60% accuracy) the sender of a message. Later, [156] used unigrams for authorship detections, and achieved almost 90% accuracy by stacking multiple SMS together, effectively increasing the document size. However, they also noted how increasing the number of authors in the corpus lowers the percentage of correctly identifying a message. This is a plausible result, considering that the probability of a determined unigram count being high increases with the number of authors, and will apply to most kind of stylometric features [132], unless the classifiers is specifically designed to handle larger datasets [129]. Within the task of authorship characterisation

in SMS, [201] achieved best performances (85% for gender, 64% for age, and 53% on the joint distribution) using a combination of word unigrams and character bigram, trigram and 4-grams to detect age and gender. Taking into account word n-grams of greater length actually reduced performance.

5.3 Mitigations

As a defence from fingerprinting based on sensitive information, the chatbot client will need to detect the information and replace it with a syntactically equivalent sentence, as to preserve the functionality on the servers' NLP tasks.

The simplest approach would be detecting the sensitive words through NER, and replacing them with a standard "null" word in the same category (or several, which are periodically cycled through for substitutions). This would not fool our sophisticated adversary, since even a simple language model could detect words not fitting appropriately in the right context. It is difficult to say how well the adversary could retrieve the original substituted word from context; a technique to assess this is provided in [67], but it requires a human generated training corpus. It seems plausible that, using the vast arrays of computational power to it available, the server might be able to accomplish this task with reasonable certainty. Therefore, it will be necessary to equip the chatbot client with its own language model to generate plausible substitutions to go undetected by the adversary, while keeping within the constraints of running on a device with limited power. Once all words in a message are detected and a plausible substitution is generated, the pair of original word to replacement would be added to a local dictionary data structure. Then, the message would be sent to the server, which would parse the message and return an appropriate response data object, including the parsed intent, its parameters, and any additional commands to execute. The parameters received will include reference to the replaced words; these can then be replaced with the original word and used to construct a reply for the user.

Preventing the adversary from extracting information from the chatbot can be addressed at the client level, but also requires careful design decisions from the chatbot application developer. When writing the script for chatbot intents, some considerations need to be made to not include any conversations thread where the user might include unpredictable private information. Even if the chatbot client will try to automatically remove domain specific sensitive information, because the processing power and language modelling tools available to the chatbot will always be inferior to the adversary's, it will never be possible to perfectly remove all sensitive information.

The other approach is to protect the sender anonymity through adversarial stylometry. We are still constrained in terms of what resources are available to the client for this task. Approaches to author obfuscation have been developed to use manual, computer assisted and automatic methods. [31] found that human subjects perform successful authorship obfuscation by "dumbing down" their language (shortening sentences, using words with fewer syllables etc). While this approach works well for longer texts, it would not be as effective for instant messaging, because they generally present sim-

ilar characteristics already. On the other hand, [79] found that teenagers' expressivity when writing is context dependant, and is generally less explicit when using instant messaging application, suggesting that users could be trained to effectively self-censor their own stylistic attributes. There are risks, however, that this kind of artificial restrictions on the utterances might impact how honestly the user will interact with the chatbot agent, which could be harmful for some e-health applications. [116] develop an open source toolchain for research in adversarial stylometric tracking and defence, *Anonymouth*, which shows words to remove and words to add while the text is being composed. Similarly, the *Nondescript* web app guides user in removing stylometrically significant information from their documents [51]. It is unclear how well such a system would apply to usage in a chat setting. Even if recommendations were neatly integrated in the interface to the chat program, users would be to some extent impaired in their ability to conduct a conversation if sufficiently slowed down. Thus, a seamless experience would only be a guaranteed by an automated system. [157] first proposed the usage of two-way machine translation to take advantage of information loss in the translation models to provide a plausible sentence paraphrase without loss of meaning. The feature was then added in recent versions of Anonymouth [117]. Automatic phrase transformations through synonym replacement was also shown to have some success [93], with further performance increases through Transformer-based style transfer, at the cost of preserving the sentences' meanings [62]. However, [60] found that a learned representation of linguistic tasks would still encode demographic information about the authors, even when trying to use machine learning techniques such as adversarial training to limit it.

Measuring the effectiveness of these techniques is challenging, especially since the adversary might try to hide its capabilities from us. Most Machine Translation methods rely on a corpus of existing translations to provide an evaluation. If we choose to model our task as such (translating from text with a particular style to text with lack of defining style, or using a different style), we can evaluate any proposed solutions in terms of the *Extended-Brennan-Greenstadt adversarial corpus* [7], which contains text samples and an equivalent obfuscated or imitation text. Other evaluation methods have been proposed, such as [184]'s evaluation of different offensive and defensive language modelling strategies in terms of how cryptographic protocols are evaluated, through an adversarial game where one player is trying to generate plausible text, and the other tries to detect forgeries using linguistic phenomena that will only occur in the real data. [148] formulates standard metrics for author obfuscation: a produced text should be safe (hard for an adversary to reconstruct its original style), sound (the meaning from the original text should be preserved) and sensible (it should be obvious that the text has not been replaced artificially). A combination of these metrics will be needed to evaluate the appropriateness of any method employed in our protocol.

Chapter 6

Evaluation

In this section, we attempt to validate the effectiveness of our system by conducting tests on some of its components.

It should be noted that in order to analyse a complex system, it is not sufficient to test its parts separately, but their interaction needs to be assessed as well. This is particularly difficult when human interaction is involved, as human agents are unpredictable. Some behaviour can be captured through experimenting with human participants, but at a limited scale, so the development of an expert system to simulate the user behaviour might be required [212]. Additionally, security applications need to model the adversary's capabilities. While adversarial modelling on a network and cryptographic level is well understood [55], and techniques have been developed to produce automatic tests of security properties [170], the field of modelling adversaries from the Natural Language perspective is still relatively novel. Because development of secure protocols is error prone, stronger assurances can be had through formal verification, for which several formalisations have been developed [28, 35, 40].

6.1 Routing

While we have developed analytically the probability of the adversary breaking the anonymity of a chatbot user in terms of the network size and variables ρ and δ, the choice of parameters has not been discussed. In particular, we need to find what is an acceptable value of ρ such that the probability of breaking anonymity is acceptably low, while the number of messages received by the server is sustainable. Additionally, the delay δ, while it does not play directly in the probability of breaking a connection, needs to be large enough to prevent timing attacks; but it can not be too large, since to produce an effective chatbot, a system with human-like performance in replying to messages is necessary, hence total round trip latency can only be limited to a few seconds [127]. We therefore have to test the performance of our protocol for a variation of parameters.

There are ethical issues when trying to analyse a live anonymity system such as Tor

[102], besides the technical challenge of distributing an instrumented version of the Tor client to be run by all nodes. Although it is possible to verify design choices using a theoretical model, these will inevitably be limited in scope [85, 13] and require further practical experimentation. The Tor project has been developing Chutney, a method to automatically configure and launch a private test Tor network of arbitrary size [194]. Chutney is relatively easy to set up, requiring only the download of the Tor source code. We were quickly able to generate and run some test networks, and customise their topology. However, the API for the tool is still severely limited, and controls to specify the quantity of network traffic for each node to send are still very coarse grained.

Multiple research-oriented distributed overlay networks have been used in the past for Tor research (DETERLab, EmuLab, PlanetLab), but they have severe limitations in term of applications that can be run and resource availability [178]. To test the protocol at scale without significant hardware requirements, experiment with different variables, and obtain reproducible results, it is appropriate to reproduce the behaviour of the network algorithmically, and observe the results. There are two approaches, simulation and emulation: simulations simplify the behaviour of a protocol while trying to obtain similar results, and emulation attempts to reproduce all functionality in real time using virtual nodes, at significant performance costs. Emulation programs include ExperimenTor [18] and SNEAC [182], while COGS [59] and Tor Path Simulator [87] (limited in scope to modelling path selection in routing) have been used in previous studies. None of these tools seem to still be in active development, except for the SNEAC-based NetMirage emulator [200], which is still in beta, and the Shadow simulation tool.

Shadow [86] is a discrete-event network simulator tool which provides several dynamically loaded plugins to run experiments on various network protocols, and whose Tor interface (previously named Scallion) runs actual Tor code to accurately simulate its performance. There are several Shadow distributions available for use. Despite this, our attempts to successfully run the simulator were severely frustrated. We were unable to successfully deploy the Docker and AWS images, due to outdated package distributions. Compiling from sources required the installation of GCC debug systems and an older CMake version, the combination of which proved to be very hard to obtain on any systems we had access to. We were only able to deploy the software on a Vagrant virtual machine, but resources available on the host system were not sufficient to run any useful simulations. To obtain a good semblance of fidelity, a successful Shadow simulation would require a far larger amount of resources than what was available to us.

It is quite concerning that running any interesting experiment on Tor would take the amount of effort that we went through, and more. Being such a critical service for a large population which requires anonymity to protect their very lives, it should be expected for researchers to be able to easily deploy and test networks in a safe environment.

6.2 Natural language alterations

We performed some experiments to evaluate the effectiveness of the Natural Language obfuscation tools presented in chapter 5.

6.2.1 Word substitution

We first attempted to implement a simple tool to filter food words from a message, and replace with a chosen standard *null word*. This required, as a first step, a tool to detect which words in a sentence are food. Since the problem of modelling food words has not been extensively explored, our original approach for Healthbot involved using a hard-coded word list, combined with a catchall approach based on the intent scripts and sentence syntax. The first solution produced too many false negatives, and the latter too many false positives. For the word replacement tool, we attempted to solve the problem using Named Entity Recognition. Initial results were provided by [204], which uses the CRF plugin for scikit-learn [142], trained on a dataset of restaurant reviews from the Yelp website, which have been manually tagged to identify food words. Training features for the CRF include the word, its ending, part of speech tag, and neighbouring vocabulary; and the limited-memory BFGS optimisation algorithm [38] was used for training. To address the fact that Yelp review would only contain prepared, restaurant food within its dataset, in our first experiment, we retrained the model to take into account home cooked food, using cookbooks from [1], a collection of machine-readable historical cookbooks. For our training, we used the naive assumption that the text within the *ingredient* tags could be considered as food.

Based on this NER model, we built a simple filtering tool that would run the CRF model to tag each word in a message sentence as Food or Other. Every occurrence of a Food-tagged word (or a sequence of them) would then be replaced by a standard null-word. We tested the effectiveness of this model using randomly chosen food world *broccoli*, by filtering the experiment data captured during the testing of Healthbot [108].

From a dataset of 2895 messages, our model inserted the null word broccoli into 1160 messages, for a total of 2275 substitutions. Since the Healthbot dataset is untagged, it is difficult to evaluate its effectiveness. As an indicative test, we selected a small sample of 36 messages containing 392 words overall. For this sample set, the F_1 score, a standard metric in NLP tasks that balances the precision and recall of a system (in our case, the proportion of replaced words that were actually food in the original, as opposed to the proportion of actual food words that were successfully replaced) to be

$$2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} = 2 \cdot 0.42 \cdot 0.70.42 + 0.7 = 0.525$$

.

Examining the resulting sentences, we can observe that the model has some limited understanding of words related to food, with most ingredients being correctly replaced, as well as words that are not directly foods but could be considered as “close” to

food in a Word2vec model get. However, reading some sentence, it becomes painfully obvious that they are not natural language (for instance, the model produced sentence "For lunch, I had broccoli and broccoli with broccoli"). A weakness of this CRF implementation was the tagging scheme in the training data; rather than using the classic BIO scheme (Beginning for the first word in an entity, Inside for any further words, Outside for words that are not), it simplified to a simple Inside/Outside tag. Thus, composite words, which our cookbook dataset abounded with, were all used as food words in training, when instead some of them only assumed the meaning of food when combined with other words. For instance, a frequent word that appeared in the ingredients tag was cup, as a measurement tool. But since our CRF tagged it as Inside, sentences like "I had a cup", which referred to measuring the quantity of some food previously logged, were incorrectly translated as "I had a broccoli". Further, weaknesses in the tagging caused a sentence like "I just ate some Indian food" to be translated to "I just ate some Indian broccoli". While here the word food is correctly detected as a food word, the word Indian, which was not usually tagged as Inside, is ignored. This doesn't satisfy our purposes, because the adversary will be easily able to retrieve what the original assertion was.

Luckily, it is very easy to modify the tagged dataset to include the full BIO training scheme. Having done so, we produced a second language replacement tool, by training a NER classifier on top of BERT. We finetuned the BERT large uncased model to a dataset of 1M entries from Yelp and the cookbooks, using code adapted from [197]. After 5 epochs, validation accuracy reached 98%, so we stopped training; this produced an F_1 score of 0.90. We then used this model to replace food words in our Healthbot dataset. Out of our 2831 messages, 1041 had at least a word replaced, for a total of 2180, a slightly more conservative quantity than the CRF model. Again, we selected a random sample of 36 messages, which contained 412 total words, and computed the F_1 score to be

$$2 \cdot \frac{0.4 \cdot 0.63}{0.4 + 0.63} = 0.491$$

Despite the high accuracy the model achieves on the validation set, the performance of the model is almost equivalent, if not worse. Of course, we could have just been unlucky in picking the 36 messages to evaluate, so further more sophisticated evaluations will be needed. A cursory pass through a larger quantity of substituted messages seem to produce better results, so perhaps the F_1 score is not a good metric for our task.

We then attempted to produce better word substitutions based on context. In order to do this, we used the language modelling capabilities of BERT to produce suitable replacements on the masked out words we removed through NER. Our first attempt to do this used the BERT large uncased model, without fine-tuning. We run the classifier for 1000 epochs, producing a 12% rate of matching the correct word. Because of high traffic on the GPU cluster at this time, we had to training this model on a CPU machine, which took more than a week; therefore, we had to limit our next model using fine-tuning to 100 epochs of training. This resulted in a 5% match rate. Having a low match is actually good for our hiding protocol, because we don't want the language replacement to reproduce the original word. However, the produced replacements severely

altered the syntax of the sentence, with many food words being replaced by punctuation. We attribute these results to our low quality dataset. Further work in food classification and replacement will need to source better datasets, or to apply more sophisticated preprocessing techniques.

6.2.2 Stylometric analysis

We also wanted to test how identifiable Healthbot users were. To do so, we performed some experiments on our dataset using a popular stylometry detection tool, *JStylo* [115].

JStylo provides a graphical user interface to conduct a variety of experiments on a multiple author dataset, with many options to choose configurations and features. We did not expect our small corpus to produce excellent results, but we wanted to get a lower bound on how easy it would be to identify authorship within our small set of 11 chatbot users.

We conducted multiple experiments based on the techniques described in [201] and [156], running k-fold validation on our entire dataset using the AdaBoost [169] and Support Vector Machines (SVM) [186] learning algorithms. As features, we chose word frequency, top character bi-gram and tri-gram (four-grams not being available in JStylo). We also attempted using other learning algorithms, such as Neural Networks, but the JStylo implementation failed to produce a result in a timely manner.

Our best accuracy, 71.49%, was provided by running AdaBoost using both word and character n-gram frequency on single message documents, and stacking messages in documents of 10 each resulting in a very close second for word only characteristics (as in [156]) with 71.32%. SVMs performance was also very close, with values ranging from 71.03% to 68.38%. However, for this learning algorithm, character n-gram seemed to have no effect, with results between model being very close despite the addition of that feature.

Our results should not be viewed as comprehensive benchmark, as we only explored a narrow set of algorithms and their parameters, features and document sizes. The relatively high results produced with low effort should be enough to argue that users interacting with a regular chatbot are highly identifiable, and taking the precaution of using stylometric obfuscation is necessary to ensure anonymity. Further studies should take advantage of the powerful nature of the Jstylo framework for stylometric studies. However, further fine-tuning is needed for the study of short documents like instant messages.

Having verified that JStylo had enough success detecting authorship in our message corpus, we wanted to see if its companion tool Anonymouth would provide some interesting suggestions onto what kind of modifications to run on the messages. Unfortunately, we were unable to run the program, which was built using an older version of Java, and it has not been actively maintained in the last few years [17]. That the once premiere author obfuscation tool is now usable is also a serious issue, although there is

no indication if users in the wild ever relied on it as a defensive technique, or if it was only used as a research tool.

Chapter 7

Conclusion

In this work, we provide the first, to our knowledge, attempt at designing a secure chatbot, where user data is kept confidential from the various actors participating in the protocol. We assessed, through Data Flow Diagrams, the risk factors in our previous chatbot design, and developed a distributed architecture that is resilient to several attacks, putting the chatbot client at the centre of data transfer operations. We discussed the advantages and drawbacks of using centralised or decentralised architectures for long term data storage and as a platform to run data analysis. The security of our protocol against a global passive adversary, which can control both the NLP server and other chatbot clients in our peer-to-peer routing protocol, relies on keeping users anonymous. Our protocol achieved this by routing messages through various clients, a technique introduced in the Crowds protocol. Our usage of Tor as a transport layer to provide confidentiality and integrity addresses the main flaw of Crowds. We also propose the addition of artificial network delays to counteract timing attacks in Tor. We compute the probability of a successful attack by this adversary, in terms of the size of the network, the size of colluding Tor nodes and chatbot clients, and the probability of propagating a message. Additionally, we discuss the threat posed by adversaries with various capabilities, and how our protocol addresses them. We describe how the analysis of natural language text is a risk for protocols using anonymity through the fingerprinting of sensitive information, and stylometric attack, and review how currently available language technologies can be used to modify the text for anonymity. Since the majority of our time was spent on producing a strong theoretical design, its evaluation was less developed, in part due to technical problems deriving from the unavailability of tooling. Still, we conducted some experiments in order to find how recent neural architectures could improve the performance of the tasks we were interested in, and found that there are still limitations to their effectiveness, which we speculate derives from the low quality of available datasets.

As part of our work, we have provided some suggestions for practical solutions that could be used today to implement our design. We believe that the suggested technical solutions are today mature enough to build a prototype as we described, at least using a centralised backend. We can not guarantee, however, that such a prototype will be safe for usage. Further analysis of the routing protocol and natural language processing

capabilities will be needed to guarantee acceptable levels of security. Furthermore, our design is much more computationally intensive than the current paradigm of thin clients; efficiency will therefore have to be another primary concern for any attempts at an implementation. This also involves severe increases in compute power for the server infrastructure. While we discussed some proposals on how to incentivise the adoption of our protocol, further work on producing convincing economic motivations will be needed.

The field of security and privacy is an exciting one to work in, as more people start to understand what the large scale collection of data allows companies and governments to infer about individuals, and privacy violations become more reported in the media. As commercial products try to catch up with changing customer exceptions, the landscape of the field is in constant change. After their initial 2014 scorecard [61], the Electronic Frontier Foundation released an updated version in 2018 [146], outlining how the field of secure messaging is complicated to navigate, and no solution is perfect for everyone's need. Even if more companies are adopting end-to-end encryption, there is more to build a secure messaging platform, from backup encryption to allowing anonymous aliases. Their checklist for building a secure messaging application is a good overview on features that are necessary to protect users, some of which we adopted in our protocol.

More recently, Facebook, which controls the two largest chat platforms and the biggest bot marketplace in the West, announced that they will be rolling out end to end encryption to users across all their apps, which will be unified under a single communication protocol [220]. If their plan were to carry through, it is uncertain what will happen to their bot offering. We hope that chatbots will not be removed from the platform, or become a second class citizen, left with less security assurances compared to regular chat. While we applaud the effort or tearing down the artificial walls of chat ecosystems, and rolling out encryption to as wide a population as possible, we are sceptical of the company's good intentions. We have little hope that Facebook will use this opportunity to embrace open standards [149], and it seems likely that the aim of this operation is to increase the already massive collection of metadata the company has accumulated [199]. As we have discussed, this should be a worrying sign, as metadata and the combination of different data sources allow accurate inference in the face of other security guarantees. Therefore, whatever the eventual outcomes of this project, we recommend the reader to be cautious about what instant messaging platform, as it relates to chatbot in particular, they choose to entrust their information with.

Bibliography

[1] Feeding America: The Historic American Cookbook Dataset. URL: https://www.lib.msu.edu/feedingamericadata/.

[2] Universal Declaration of Human Rights. December 1948. Art. 12. Accessed 17 February 2015. URL: http://www.unhchr.ch/udhr/lang/eng.pdf.

[3] IEEE Standard Specifications for Public-Key Cryptography. *IEEE Std 1363-2000*, pages 1–228, August 2000. URL: http://dx.doi.org/10.1109/IEEESTD.2000.92292.

[4] Pidgin, 2018. URL: http://pidgin.im.

[5] Matrix, 2019. URL: matrix.org.

[6] Marwan Abi-Antoun, Daniel Wang, and Peter Torr. Checking threat modeling data flow diagrams for implementation conformance and security. In R. E. Kurt Stirewalt, Alexander Egyed, and Bernd Fischer, editors, *22nd IEEE/ACM International Conference on Automated Software Engineering (ASE 2007), November 5-9, 2007, Atlanta, Georgia, USA*, pages 393–396. ACM, 2007. URL: https://doi.org/10.1145/1321631.1321692, doi:10.1145/1321631.1321692.

[7] S. Afroz, M. Brennan, and R. Greenstadt. Detecting Hoaxes, Frauds, and Deception in Writing Style Online. In *2012 IEEE Symposium on Security and Privacy*, pages 461–475, May 2012. doi:10.1109/SP.2012.34.

[8] Muhammad Aurangzeb Ahmad, Carly Eckert, and Ankur Teredesai. Interpretable Machine Learning in Healthcare. In *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 559–560. ACM, 2018.

[9] Mashael AlSabah and Ian Goldberg. Performance and Security Improvements for Tor: A Survey. 2015. https://eprint.iacr.org/2015/235.

[10] Ame Elliot. Chatbots, UX, and Privacy, May 2016. URL: https://simplysecure.org/blog/chatbots-ux-privacy.

[11] J. Michael Ashley. The GNU Privacy Handbook. The Free Software Foundation, 1999.

[12] Michael Backes, Aniket Kate, Praveen Manoharan, Sebastian Meiser, and Esfandiar Mohammadi. AnoA: A Framework For Analyzing Anonymous Communication Protocols. In *Proceedings of the of the 26th IEEE Computer Security Foundations Symposium (CSF)*, pages 163–178. IEEE, 2013.

[13] Michael Backes, Aniket Kate, Sebastian Meiser, and Esfandiar Mohammadi. (Nothing else) MATor(s): Monitoring the Anonymity of Tor's Path Selection. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security - CCS '14*, pages 513–524, Scottsdale, Arizona, USA, 2014. ACM Press. URL: http://dl.acm.org/citation.cfm?doid=2660267.2660371, doi:10.1145/2660267.2660371.

[14] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How To Backdoor Federated Learning. 2018. arXiv:1807.00459.

[15] Raad Bahmani, Manuel Barbosa, Ferdinand Brasser, Bernardo Portela, Ahmad-Reza Sadeghi, Guillaume Scerri, and Bogdan Warinschi. Secure Multiparty Computation from SGX. In Aggelos Kiayias, editor, *Financial Cryptography and Data Security*, volume 10322, pages 477–497. Springer International Publishing, Cham, 2017. URL: http://link.springer.com/10.1007/978-3-319-70972-7_27, doi:10.1007/978-3-319-70972-7_27.

[16] Barbara Ondrisek. Privacy and Data Security of Chatbots. URL: https://medium.com/@electrobabe/privacy-and-data-security-of-chatbots-6ab87773aadc.

[17] Marc Barrowclift. Compile without Eclipse, November 2013. URL: https://github.com/psal/anonymouth/issues/2#issuecomment-27826933.

[18] Kevin Bauer, Damon McCoy, Micah Sherr, and Dirk Grunwald. ExperimenTor: A Testbed for Safe and Realistic Tor Experimentation. In *4th USENIX Workshop on Cyber Security Experimentation and Test*, page 8, August 2011. URL: http://www.cs.uwaterloo.ca/~k4bauer/papers/bauer-cset11.pdf.

[19] Aurélien Bellet, Rachid Guerraoui, Mahsa Taziki, and Marc Tommasi. Personalized and Private Peer-to-Peer Machine Learning. 2017. arXiv:1705.08435.

[20] Steven M. Bellovin. Frank Miller: Inventor of the One-Time Pad. *Cryptologia*, 35(3):203–222, July 2011. URL: https://doi.org/10.1080/01611194.2011.583711, doi:10.1080/01611194.2011.583711.

[21] Duane Bender and Kamran Sartipi. HL7 FHIR: An Agile and RESTful approach to healthcare information exchange. In *Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems*, pages 326–331. IEEE, 2013.

[22] Y. Bengio. Neural net language models. *Scholarpedia*, 3(1):3881, 2008. revision #140963. doi:10.4249/scholarpedia.3881.

[23] Benjamin Zimmer. Language Log: Forensic linguistics, the Unabomber, and the etymological fallacy, January 2006. URL: http://itre.cis.upenn.edu/~myl/languagelog/archives/002762.html.

[24] Daniel J. Bernstein. Introduction to post-quantum cryptography. In Daniel J. Bernstein, Johannes Buchmann, and Erik Dahmen, editors, *Post-Quantum Cryptography*, pages 1–14. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. URL: https://doi.org/10.1007/978-3-540-88702-7_1.

[25] Ethan S Bernstein. The transparency paradox: A role for privacy in organizational learning and operational control. *Administrative Science Quarterly*, 57(2):181–216, 2012.

[26] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. Analyzing Federated Learning through an Adversarial Lens. *arXiv:1811.12470 [cs, stat]*, November 2018. URL: http://arxiv.org/abs/1811.12470, arXiv:1811.12470.

[27] Ken Birman. The promise, and limitations, of gossip protocols. *ACM SIGOPS Operating Systems Review*, 41(5):8–13, 2007.

[28] Bruno Blanchet. Modeling and Verifying Security Protocols with the Applied Pi Calculus and ProVerif. *Foundations and Trends® in Privacy and Security*, 1(1-2):1–135, 2016. URL: http://www.nowpublishers.com/article/Details/SEC-004, doi:10.1561/3300000004.

[29] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive Zero-knowledge and Its Applications. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, pages 103–112. ACM, 1988. URL: http://doi.acm.org/10.1145/62212.62222, doi:10.1145/62212.62222.

[30] Nikita Borisov, Ian Goldberg, and Eric Brewer. Off-the-record communication, or, why not to use PGP. In *Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society - WPES '04*, page 77, Washington DC, USA, 2004. ACM Press. URL: http://portal.acm.org/citation.cfm?doid=1029179.1029200, doi:10.1145/1029179.1029200.

[31] Michael Robert Brennan and Rachel Greenstadt. Practical attacks against authorship recognition techniques. In *Twenty-First IAAI Conference*, 2009.

[32] Marcelo Luiz Brocardo, Issa Traore, Sherif Saad, and Isaac Woungang. Authorship verification for short messages using stylometry. In *2013 International Conference on Computer, Information and Telecommunication Systems (CITS)*, pages 1–6. IEEE, 2013.

[33] Marcelo Luiz Brocardo, Issa Traore, and Isaac Woungang. Continuous Authentication Using Writing Style. In Mohammad S. Obaidat, Issa Traore, and Isaac Woungang, editors, *Biometric-Based Physical and Cybersecurity Systems*, pages 211–232. Springer International Publishing, Cham, 2019. URL: https://doi.org/10.1007/978-3-319-98734-7_8, doi:10.1007/978-3-319-98734-7_8.

[34] Brooks, John. Ricochet, 2017. URL: https://ricochet.im.

[35] N Brownlee. Formal Systems (Europe) Ltd. Failures-Divergence Refinement. FDR2 User Manual. Available at http://www. formal. demon. co. uk/fdr2manual/index. html. In *Blount MetraTech Corp. Accounting Attributes and Record Formats http://www. Ietf. Org/Rfc/Rfc2924. Txt*. Citeseer, 2000.

[36] Burgess, Matt. Can you really trust the medical apps on your phone? *Wired UK*, January 2017. URL: https://www.wired.co.uk/article/health-apps-test-ada-yourmd-babylon-accuracy.

[37] Butcher, Lola. Why Is it So Hard to Match Patients With Their Medical Records? *Undark*, March 2019. URL: https://undark.org/article/patient-matching-medical-records/.

[38] R. Byrd, P. Lu, J. Nocedal, and C. Zhu. A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995. URL: https://doi.org/10.1137/0916069, arXiv:https://doi.org/10.1137/0916069, doi:10.1137/0916069.

[39] Carole Cadwalladr and E Graham-Harrison. The Cambridge analytica files. *The Guardian*, 21:6–7, 2018.

[40] Ran Canetti. Universally Composable Security: A New Paradigm for Cryptographic Protocols. 2000. https://eprint.iacr.org/2000/067.

[41] Caroline Humer and Jim Finkle. Your medical record is worth more to hackers than your credit card. *Reuters*, September 2014. URL: https://www.reuters.com/article/us-cybersecurity-hospitals-idUSKCN0HJ21I20140924.

[42] Fred H Cate. The failure of fair information practice principles. 2006.

[43] Carole E Chaski. Who's at the keyboard? Authorship attribution in digital evidence investigations. *International journal of digital evidence*, 4(1):1–13, 2005.

[44] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2), February 1981.

[45] Hsin-Pai Cheng, Patrick Yu, Haojing Hu, Feng Yan, Shiyu Li, Hai Li, and Yiran Chen. LEASGD: An Efficient and Privacy-Preserving Decentralized Algorithm for Distributed Learning. 2018. arXiv:1811.11124.

[46] Wolfie Christl and Sarah Spiekermann. *Networks of Control. A Report on Corporate Surveillance, Digital Tracking, Big Data & Privacy*. facultas, 2016.

[47] Clifford C Cocks. A note on non-secret encryption. *CESG Memo*, 1973.

[48] Michael Collins. Language Modeling. Lecture Notes, 2013. URL: http://www.cs.columbia.edu/~mcollins/lm-spring2013.pdf.

[49] Collins, Malcom. The Ideology of Anonymity and Pseudonymity. October 2013. URL: https://www.huffingtonpost.com/malcolm-collins/online-anonymity_b_3695851.html.

[50] Destini Davis, Peter Higgins, Peter Kormarinski, Joseph Marques, Nicholas Orlans, and James Wayman. State of the art biometrics excellence roadmap. *MITRE Corporation: Bedford, MA, USA*, 1:4–14, 2008.

[51] Robin Camille Davis. Obfuscating Authorship: Results of a User Study on Nondescript, a Digital Privacy Tool. *CUNY Academic Works*, 2019. URL: https://academicworks.cuny.edu/jj_pubs/253/.

[52] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, abs/1810.04805, 2018.

[53] Cambridge Dictionary. Cambridge Advanced Learner's Dictionary & Thesaurus. *Pieejams: https://dictionary. cambridge. org/dictionary/english/interactive.(skatīts 15.04. 2018)*, 2017.

[54] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. Technical report, Naval Research Lab Washington DC, 2004.

[55] Quang Do, Ben Martini, and Kim-Kwang Raymond Choo. The Role of the Adversary Model in Applied Security Research. Technical Report 1189, 2018. URL: http://eprint.iacr.org/2018/1189.

[56] Yevgeniy Dodis and Joel H. Spencer. On the (non)Universality of the One-Time Pad. In *FOCS*, 2002.

[57] John R. Douceur. The Sybil Attack. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, IPTPS '01, pages 251–260, London, UK, 2002. Springer-Verlag. URL: http://dl.acm.org/citation.cfm?id=646334.687813.

[58] Cynthia Dwork and Aaron Roth. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3-4):211–407, 2013. URL: http://www.nowpublishers.com/articles/foundations-and-trends-in-theoretical-computer-science/TCS-042, doi:10.1561/0400000042.

[59] Tariq Elahi, Kevin Bauer, Mashael AlSabah, Roger Dingledine, and Ian Goldberg. Changing of the guards: A framework for understanding and improving entry guard selection in Tor. In *Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society - WPES '12*, page 43, Raleigh, North Carolina, USA, 2012. ACM Press. URL: http://dl.acm.org/citation.cfm?doid=2381966.2381973, doi:10.1145/2381966.2381973.

[60] Yanai Elazar and Yoav Goldberg. Adversarial Removal of Demographic Attributes from Text Data. *CoRR*, abs/1808.06640, 2018. URL: http://arxiv.org/abs/1808.06640, arXiv:1808.06640.

[61] Electronic Frontier Foundation. Secure Messaging Scorecard, November 2014. URL: https://www.eff.org/node/82654.

[62] Chris Emmery, Enrique Manjavacas, and Grzegorz Chrupala. Style Obfuscation by Invariance. *CoRR*, abs/1805.07143, 2018. URL: http://arxiv.org/abs/1805.07143, arXiv:1805.07143.

[63] Equality, Human Rights Commission, et al. Equality Act 2010 Employment Statutory Code of Practice. *London: The Stationery Office, 2011. ohaw. co/HuyvZO*, 2010.

[64] Facebook. Customer Matching. URL: https://developers.facebook.com/docs/messenger-platform/identity/customer-matching.

[65] Eric A. Feldman. The Genetic Information Nondiscrimination Act (GINA): Public Policy and Medical Practice in the Age of Personalized Medicine. *Journal of General Internal Medicine*, 27(6):743–746, June 2012. URL: https://doi.org/10.1007/s11606-012-1988-6, doi:10.1007/s11606-012-1988-6.

[66] Niels Ferguson, Bruce Schneier, and Tadayoshi Kohno. *Cryptography Engineering: Design Principles and Practical Applications*. Wiley Publishing, Inc., Indianapolis, Indiana, October 2015. URL: http://doi.wiley.com/10.1002/9781118722367, doi:10.1002/9781118722367.

[67] Michael Gamon, Anthony Aue, and Martine Smets. Sentence-level MT evaluation without reference translations: Beyond language modeling. In *Proceedings of EAMT*, pages 103–111, 2005.

[68] Craig Gentry. *A FULLY HOMOMORPHIC ENCRYPTION SCHEME*. PhD thesis, STANFORD UNIVERSITY, 2009.

[69] David Goldschlag, Michael Reed, and Paul Syverson. Onion routing for anonymous and private internet connections. *Communications of the ACM*, 42(2):39–40, 1999.

[70] Lawrence O Gostin and James G Hodge Jr. Personal privacy and common goods: A framework for balancing under the national health information privacy rule. *Minn. L. Rev.*, 86:1439, 2001.

[71] Glenn Greenwald and Ewen MacAskill. NSA Prism program taps into user data of Apple, Google and others. *The Guardian*, 7(6):1–43, 2013.

[72] Benjamin Greschbach, Gunnar Kreitz, and Sonja Buchegger. The devil is in the metadata - New privacy challenges in Decentralised Online Social Networks. In *2012 IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 333–339, Lugano, Switzerland, March 2012. IEEE. URL: http://ieeexplore.ieee.org/document/6197506/, doi:10.1109/PerComW.2012.6197506.

[73] Quinn Grundy, Kellia Chiu, Fabian Held, Andrea Continella, Lisa Bero, and Ralph Holz. Data sharing practices of medicines related apps and the mobile ecosystem: Traffic, content, and network analysis. *BMJ*, 364, 2019. URL: https://www.bmj.com/content/364/bmj.l920, arXiv:https://www.bmj.com/content/364/bmj.l920.full.pdf, doi:10.1136/bmj.l920.

[74] DE Hammer-Lahav and D Hardt. The oauth2. 0 authorization protocol. 2011. *Technical report, IETF Internet Draft*, 2011.

[75] Hamza Harkous. Encryption, AI, and the Myth of Incompatibility, June 2016. URL: https://chatbotsmagazine.com/encryption-ai-and-the-myth-of-incompatibility-9afca1ca115.

[76] Hamza Harkous. How Chatbots Will Redefine the Future of App Privacy, April 2016. URL: https://chatbotsmagazine.com/how-chatbots-will-redefine-the-future-of-app-privacy-eb68a7b5a329.

[77] Hamza Harkous and Kassem Fawaz. PriBots: Conversational Privacy with Chatbots. In *Twelfth Symposium on Usable Privacy and Security, SOUPS*, page 6, June 2016.

[78] Hide me. VPN vs Proxy. URL: https://hide.me/en/proxy/vpn-vs-proxy.

[79] Lisa Hilte, Reinhild Vandekerckhove, and Walter Daelemans. Expressive markers in online teenage talk. *Nederlandse Taalkunde*, 23(3):293–323, 2019.

[80] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[81] David I. Holmes. The Evolution of Stylometry in Humanities Scholarship. *Literary and Linguistic Computing*, 13(3):111–117, September 1998. URL: https://doi.org/10.1093/llc/13.3.111, arXiv:http://oup.prod.sis.lan/dsh/article-pdf/13/3/111/2752801/13-3-111.pdf, doi:10.1093/llc/13.3.111.

[82] Jeremy Howard and Sebastian Ruder. Universal Language Model Fine-tuning for Text Classification. 2018. arXiv:1801.06146.

[83] Andrey Ignatov, Radu Timofte, William Chou, Ke Wang, Max Wu, Tim Hartley, and Luc Van Gool. Ai benchmark: Running deep neural networks on android smartphones. In *European Conference on Computer Vision*, pages 288–314. Springer, 2018.

[84] Mohsen Imani, Armon Barton, and Matthew Wright. Guard Sets in Tor using AS Relationships. *Proceedings on Privacy Enhancing Technologies*, 2018(1), 2018.

[85] Rob Jansen, Kevin Bauer, Nicholas Hopper, and Roger Dingledine. Methodically Modeling the Tor Network. In *Proceedings of the USENIX Workshop on Cybersecurity Experimentation and Test (CSET 2012)*, page 9, August 2012.

[86] Rob Jansen and Nicholas Hooper. Shadow: Running Tor in a Box for Accurate and Efficient Experimentation:. Technical report, Defense Technical Information Center, Fort Belvoir, VA, September 2011. URL: http://www.dtic.mil/docs/citations/ADA559181, doi:10.21236/ADA559181.

[87] Aaron Johnson, Chris Wacek, Rob Jansen, Micah Sherr, and Paul Syverson. Users Get Routed: Traffic Correlation on Tor by Realistic Adversaries. In *Pro-*

ceedings of the 20th ACM Conference on Computer and Communications Security (CCS 2013). ACM, 2013.

[88] Kaiyinzhou. BERT-NER, December 2018. URL: https://github.com/kyzhouhzau/BERT-NER.

[89] Andrej Karpathy. The unreasonable effectiveness of recurrent neural networks. *Andrej Karpathy blog*, 21, 2015.

[90] Ravneet Kaur, Sarbjeet Singh, and Harish Kumar. Authorship Analysis of Online Social Media Content. In C. Rama Krishna, Maitreyee Dutta, and Rakesh Kumar, editors, *Proceedings of 2nd International Conference on Communication, Computing and Networking*, pages 539–549. Springer Singapore, 2019.

[91] Patrick Gage Kelley, Lucian Cesca, Joanna Bresee, and Lorrie Faith Cranor. Standardizing privacy notices: An online study of the nutrition label approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1573–1582. ACM, 2010.

[92] Auguste Kerckhoffs. La cryptographie militaire. *Journal des sciences militaires*, pages 5–38, 1883.

[93] Foaad Khosmood and Robert Levinson. Automatic synonym and phrase replacement show promise for style transformation. In *2010 Ninth International Conference on Machine Learning and Applications*, pages 958–961. IEEE, 2010.

[94] Kobie, Nicole and Burgess, Matt. The messy, cautionary tale of how Babylon disrupted the NHS. *Wired UK*, March 2019. URL: https://www.wired.co.uk/article/babylon-health-nhs.

[95] Theodore J. Kobus, III and Gonzalo S. Zeballos. International Compendium of Data Privacy Laws, 2015. URL: https://web.archive.org/web/20160521235643/http://www.bakerlaw.com/files/Uploads/Documents/Data%20Breach%20documents/International-Compendium-of-Data-Privacy-Laws.pdf.

[96] Jakub Konecný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated Learning: Strategies for Improving Communication Efficiency. *CoRR*, abs/1610.05492, 2016. URL: http://arxiv.org/abs/1610.05492, arXiv:1610.05492.

[97] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[98] Susan Landau. Making sense from Snowden: What's significant in the NSA surveillance revelations. *IEEE Security & Privacy*, 11(4):54–63, 2013.

[99] Let's Encrypt. Percentage of Web Pages Loaded by Firefox Using HTTPS. Technical report, March 2019. URL: https://letsencrypt.org/stats/.

[100] Sarah Jamie Lewis. Cwtch: Privacy Preserving Infrastructure for Asynchronous, Decentralized, Multi-Party and Metadata Resistant Applications. 2018. URL: https://cwtch.im/cwtch.pdf.

[101] Huaxin Li, Qingrong Chen, Haojin Zhu, Di Ma, Hong Wen, and Xuemin Sherman Shen. Privacy leakage via de-anonymization and aggregation in heterogeneous social networks. *IEEE Transactions on Dependable and Secure Computing*, 2017.

[102] Karsten Loesing, Steven J. Murdoch, and Roger Dingledine. A Case Study on Measuring Statistical Data in the Tor Anonymity Network. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Radu Sion, Reza Curtmola, Sven Dietrich, Aggelos Kiayias, Josep M. Miret, Kazue Sako, and Francesc Sebé, editors, *Financial Cryptography and Data Security*, volume 6054, pages 203–215. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. URL: http://link.springer.com/10.1007/978-3-642-14992-4_19, doi:10.1007/978-3-642-14992-4_19.

[103] Lopez, Patrice. A reproducibility study on neural NER, September 2018. URL: http://science-miner.com/a-reproducibility-study-on-neural-ner/.

[104] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. In *22nd International Conference on Data Engineering (ICDE'06)*, pages 24–24, April 2006. doi:10.1109/ICDE.2006.1.

[105] Joshua C Mandel, David A Kreda, Kenneth D Mandl, Isaac S Kohane, and Rachel B Ramoni. SMART on FHIR: A standards-based, interoperable apps platform for electronic health records. *Journal of the American Medical Informatics Association*, 23(5):899–908, 2016.

[106] Marchesini, Kathryn, J.D. and Noonan, Timothy, J.D. HIPAA & Health Information Portability: A Foundation for Interoperabilityf. *HealthITBuzz*, August 2018. URL: https://www.healthit.gov/buzz-blog/privacy-and-security-of-ehrs/hipaa-health-information-portability-a-foundation-for-interoperability.

[107] Marlinspike, Moxie. We Should All Have Something To Hide, June 2013. URL: https://moxie.org/blog/we-should-all-have-something-to-hide/.

[108] Lorenzo Martinico. *Chatting about Data - a Conversational Interface for Meal Tracking*. MInf Project (Part 1) Report, University of Edinburgh, Edinburgh, 2018.

[109] Paulo Martins, Leonel Sousa, and Artur Mariano. A Survey on Fully Homomorphic Encryption: An Engineering Perspective. *ACM Computing Surveys*, 50(6):1–33, December 2017. URL: http://dl.acm.org/citation.cfm?doid=3161158.3124441, doi:10.1145/3124441.

[110] Matt Schlicht. Critical Announcement: Facebook Messenger Is Not Allowing New Bots (Resolved), March 2018. URL: https://chatbotsmagazine.com/critical-announcement-facebook-messenger-is-not-allowing-new-bots-temporary-4a13d12ed76.

[111] Matthew Hodgson. Matrix and Riot confirmed as the basis for France's Secure Instant Messenger app, April 2018. URL: https://matrix.org/blog/2018/04/26/matrix-and-riot-confirmed-as-the-basis-for-frances-secure-instant-messenger-app/.

[112] Diana McCarthy. Lexical substitution as a task for wsd evaluation. In *Proceedings of the ACL-02 Workshop on Word Sense Disambiguation: Recent Successes and Future Directions-Volume 8*, pages 109–115. Association for Computational Linguistics, 2002.

[113] Diana McCarthy and Roberto Navigli. SemEval-2007 Task 10: English Lexical Substitution Task. In *SemEval@ACL*, 2007.

[114] Diana McCarthy and Roberto Navigli. The English lexical substitution task. *Language Resources and Evaluation*, 43(2):139–159, June 2009. URL: https://doi.org/10.1007/s10579-009-9084-1, doi:10.1007/s10579-009-9084-1.

[115] Aleecia M McDonald and Lorrie Faith Cranor. The cost of reading privacy policies. *ISJLP*, 4:543, 2008.

[116] Andrew WE McDonald, Sadia Afroz, Aylin Caliskan, Ariel Stolerman, and Rachel Greenstadt. Use fewer instances of the letter "i": Toward writing style anonymization. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 299–318. Springer, 2012.

[117] AW McDonald, Jeffrey Ulman, Marc Barrowclift, and Rachel Greenstadt. Anonymouth revamped: Getting closer to stylometric anonymity. In *PETools: Workshop on Privacy Enhancing Tools*, volume 20, 2013.

[118] Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning Differentially Private Recurrent Language Models. In *International Conference on Learning Representations (ICLR)*, 2018. URL: https://openreview.net/pdf?id=BJ0hF1Z0b.

[119] H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated Learning of Deep Networks using Model Averaging. *CoRR*, abs/1602.05629, 2016. URL: http://arxiv.org/abs/1602.05629, arXiv:1602.05629.

[120] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning Differentially Private Language Models Without Losing Accuracy. *CoRR*, abs/1710.06963, 2017.

[121] Oren Melamud, Omer Levy, and Ido Dagan. A simple word embedding model for lexical substitution. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 1–7, 2015.

[122] Mike Perry. Tor's Open Research Topics: 2018 Edition, August 2018. URL: https://blog.torproject.org/tors-open-research-topics-2018-edition.

[123] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. 2013. arXiv:1301.3781.

[124] Ilya Mironov, Omkant Pandey, Omer Reingold, and Salil Vadhan. Computational Differential Privacy. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009*, volume 5677, pages 126–142. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009. URL: http://link.springer.com/10.1007/978-3-642-03356-8_8, doi:10.1007/978-3-642-03356-8_8.

[125] David E Morrison and Michael Svennevig. The Public Interest, the Media and Privacy. Technical report, BBC, BSC, ICSTIS, ITC, IPPR, RA, London: Broadcasting Standards Council/ITC, March 2002. Broadcasting Standards Commission Independent Committee for the Supervision of Standards of Telephone Information Services Independent Television Commission Institute for Public Policy Research The Radio Authority.

[126] Frederick Mosteller and David Wallace. *Inference and Disputed Authorship: The Federalist*. Addison-Wesley, Reading, MA, 1964.

[127] Fiona Fui-Hoon Nah. A study on tolerable waiting time: How long are Web users willing to wait? *Behaviour & Information Technology*, 23(3):153–163, May 2004. URL: http://www.tandfonline.com/doi/abs/10.1080/01449290410001669914, doi:10.1080/01449290410001669914.

[128] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. URL: https://nakamotoinstitute.org/bitcoin/.

[129] Arvind Narayanan, Hristo Paskov, Neil Zhenqiang Gong, John Bethencourt, Emil Stefanov, Eui Chul Richard Shin, and Dawn Song. On the Feasibility of Internet-Scale Author Identification. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, SP '12, pages 300–314. IEEE Computer Society, 2012. URL: https://doi.org/10.1109/SP.2012.46, doi:10.1109/SP.2012.46.

[130] Arvind Narayanan and Vitaly Shmatikov. How To Break Anonymity of the Netflix Prize Dataset. *CoRR*, abs/cs/0610105, 2006. URL: http://arxiv.org/abs/cs/0610105, arXiv:cs/0610105.

[131] Clifford Nass and Youngme Moon. Machines and Mindlessness: Social Responses to Computers. *Journal of Social Issues*, 56(1):81–103, January 2000. URL: http://doi.wiley.com/10.1111/0022-4537.00153, doi:10.1111/0022-4537.00153.

[132] Tempestt Neal, Kalaivani Sundararajan, Aneez Fatima, Yiming Yan, Yingfei Xiang, and Damon Woodard. Surveying Stylometry Techniques and Applications. *ACM Comput. Surv.*, 50(6):86:1–86:36, November 2017. URL: http://doi.acm.org/10.1145/3132039, doi:10.1145/3132039.

[133] Helen Nissenbaum. *Privacy in Context: Technology, Policy, and the Integrity of Social Life*. Stanford University Press, 2009.

[134] Helen Nissenbaum. A contextual approach to privacy online. *Daedalus*, 140(4):32–48, 2011.

[135] Cecilie Bertinussen Nordheim. *Trust in Chatbots for Customer Service–Findings from a Questionnaire Study*. PhD thesis, 2018.

[136] Olson, Parmy. This Health Startup Won Big Government Deals—But Inside, Doctors Flagged Problems. *Forbes*, December 2018. URL: https://www.forbes.com/sites/parmyolson/2018/12/17/this-health-startup-won-big-government-dealsbut-inside-doctors-flagged-problems/#2074fe4aeabb.

[137] Angela Orebaugh and Jeremy Allnutt. Classification of Instant Messaging Communications for Forensics Analysis. *The International Journal of Forensic Computer Science*, pages 22–28, 2009. URL: http://www.ijofcs.org/abstract-v04n1-pp02.html, doi:10.5769/J200901002.

[138] Alberto Ornaghi and Marco Valleri. Man in the middle attacks Demos. *Blackhat [Online Document]*, 19, 2003.

[139] Jacob Palme and Mikael Berglund. Anonymity on the Internet. December 2004. URL: http://dsv.su.se/jpalme/society/anonymity.html.

[140] Patrick Howell O'Neill. Tor's ex-director: 'The criminal use of Tor has become overwhelming'. *Cyberscoop*, May 2017. URL: https://www.cyberscoop.com/tor-dark-web-andrew-lewman-securedrop/.

[141] Adam Pawłowski and Artur Pacewicz. Wincenty Lutosławski (1863–1954): Philosophe, helléniste ou fondateur sous-estimé de la stylométrie? *Historiographia Linguistica*, 31(2):423–447, 2004.

[142] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[143] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014. URL: http://www.aclweb.org/anthology/D14-1162.

[144] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep Contextualized Word Representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018. URL: http://dx.doi.org/10.18653/v1/N18-1202, doi:10.18653/v1/n18-1202.

[145] Andreas Pfitzmann and Marit Hansen. A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management. August 2010. v0.34. URL: http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf.

[146] Erica Portnoy, Nate Cardozo, and Gennie Gebhart. Secure Messaging? More Like A Secure Mess., March 2018. URL: https://www.eff.org/deeplinks/2018/03/secure-messaging-more-secure-mess.

[147] Steven Posnack and Barker, Wes. Heat Wave: The U.S. is Poised to Catch FHIR in 2019. *HealthITBuzz*, October 2018. URL: https://www.healthit.gov/buzz-blog/interoperability/heat-wave-the-u-s-is-poised-to-catch-fhir-in-2019.

[148] Martin Potthast, Matthias Hagen, and Benno Stein. Author Obfuscation: Attacking the State of the Art in Authorship Verification. In *CLEF (Working Notes)*, pages 716–749, 2016.

[149] Pranesh Prakash. Privacy laws cannot make Facebeook and Google accountable, January 2019. URL: https://www.hindustantimes.com/analysis/privacy-laws-cannot-make-facebeook-and-google-accountable/story-Yne6DwUoGb0eO9mRxaDTaL.html.

[150] W Nicholson Price and I Glenn Cohen. Privacy in the age of medical big data. *Nature medicine*, 25(1):37, 2019.

[151] W. Nicholson Price, Margot E. Kaminski, Timo Minssen, and Kayte Spector-Bagdady. Shadow health records meet new data privacy laws. *Science*, 363(6426):448–450, 2019. URL: http://science.sciencemag.org/content/363/6426/448, arXiv:http://science.sciencemag.org/content/363/6426/448.full.pdf, doi:10.1126/science.aav5133.

[152] Lawrence R Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[153] Alec Radford, Jeff Wu, Dario Amodei, Daniela Amodei, Jack Clark, Miles Brundage, and Ilya Sutskever. Better Language Models and Their Implications, February 2019. URL: https://openai.com/blog/better-language-models/.

[154] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving Language Understanding by Generative Pre-Training. 2018. URL: https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/languageunsupervised/languageunderstandingpaper.pdf.

[155] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. Technical report, OpenAI, 2019. URL: https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

[156] Roshan Ragel, Pramod Herath, and Upul Senanayake. Authorship detection of SMS messages using unigrams. *2013 IEEE 8th International Conference on Industrial and Information Systems*, December 2013. URL: http://dx.doi.org/10.1109/ICIInfS.2013.6732015, doi:10.1109/iciinfs.2013.6732015.

[157] Josyula R Rao, Pankaj Rohatgi, et al. Can pseudonymity really guarantee privacy? In *USENIX Security Symposium*, pages 85–96, 2000.

[158] Lev Ratinov and Dan Roth. Design Challenges and Misconceptions in Named Entity Recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, CoNLL '09, pages 147–155. Association for Computational Linguistics, 2009. URL: http://dl.acm.org/citation.cfm?id=1596374.1596399.

[159] General Data Protection Regulation. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46. *Official Journal of the European Union (OJ)*, 59(1-88):294, 2016.

[160] Michael Reiter and Aviel Rubin. Crowds: Anonymity for Web Transactions. *ACM Transactions on Information and System Security*, 1(1), June 1998.

[161] Richer, J. Ed. Health Relationship Trust Profile for OAuth 2.0. Technical report, OpenID Foundation, April 2017. URL: https://openid.net/specs/openid-heart-oauth2-1_0-2017-05-31.html.

[162] Ronald L Rivest, Adi Shamir, and Leonard M Adleman. Cryptographic communications system and method. September 1983. US Patent 4,405,829.

[163] A. Rocha, W. J. Scheirer, C. W. Forstall, T. Cavalcante, A. Theophilo, B. Shen, A. R. B. Carvalho, and E. Stamatatos. Authorship Attribution for Social Media Forensics. *IEEE Transactions on Information Forensics and Security*, 12(1):5–33, January 2017. doi:10.1109/TIFS.2016.2603960.

[164] Michael Rogers, Eleanor Saitta, Bernard Tyers, Dehm, Julian, and Torsten Grote. The Briar Project, 2018. URL: https://briarproject.org.

[165] Jared Saia and Mahdi Zamani. Recent Results in Scalable Multi-Party Computation. In Giuseppe F. Italiano, Tiziana Margaria-Steffen, Jaroslav Pokorný, Jean-Jacques Quisquater, and Roger Wattenhofer, editors, *SOFSEM 2015: Theory and Practice of Computer Science*, volume 8939, pages 24–44. Springer Berlin Heidelberg, Berlin, Heidelberg, 2015. URL: http://link.springer.com/10.1007/978-3-662-46078-8_3, doi:10.1007/978-3-662-46078-8_3.

[166] Nat Sakimura, John Bradley, Mike Jones, Breno de Medeiros, and Chuck Mortimore. OpenID Connect Core 1.0 incorporating errata set 1. *The OpenID Foundation, specification*, 2014.

[167] Wojciech Samek, Thomas Wiegand, and Klaus-Robert Müller. Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models. 2017. arXiv:1708.08296.

[168] Yu F Sasaki and Kazumaro Aoki. Finding Preimages in Full MD5 Faster Than Exhaustive Search. In *EUROCRYPT*, 2009.

[169] Robert E. Schapire. Explaining AdaBoost. In Bernhard Schölkopf, Zhiyuan Luo, and Vladimir Vovk, editors, *Empirical Inference*, pages 37–52. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. URL: http://link.springer.com/10.1007/978-3-642-41136-6_5, doi:10.1007/978-3-642-41136-6_5.

[170] Ina Schieferdecker, Juergen Grossmann, and Martin Schneider. Model-Based Security Testing. *Electronic Proceedings in Theoretical Computer Science*, 80:1–12, February 2012. URL: http://arxiv.org/abs/1202.6118, arXiv:1202.6118, doi:10.4204/EPTCS.80.1.

[171] Bruce Schneier. *Schneier on Security*. John Wiley & Sons, 2009.

[172] Ari Schwartz. Looking back at P3P: Lessons for the future. *Center for Democracy & Technology*, 2009.

[173] Roy Schwartz, Maarten Sap, Ioannis Konstas, Leila Zilles, Yejin Choi, and Noah A. Smith. Story Cloze Task : UW NLP System. pages 52–55, April 2017. 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics 2017, LSDSEM 2017 ; Conference date: 03-04-2017 Through 03-04-2017. doi:10.18653/v1/W17-0907.

[174] Sébastien Helleu. Weechat, 2019. URL: https://weechat.org.

[175] Dmitriy Serdyuk, Nan Rosemary Ke, Alessandro Sordoni, Adam Trischler, Chris Pal, and Yoshua Bengio. Twin Networks: Matching the Future for Sequence Generation. In *International Conference on Learning Representations*, 2018. URL: https://openreview.net/forum?id=BydLzGb0Z.

[176] C. E. Shannon. Communication theory of secrecy systems. *The Bell System Technical Journal*, 28(4):656–715, October 1949. doi:10.1002/j.1538-7305.1949.tb00928.x.

[177] Cyrus Shaoul, R. Harald Baayen, and Chris F. Westbury. N-gram probability effects in a cloze task. *The Mental Lexicon*, 9(3):437–472, 2014. URL: https://www.jbe-platform.com/content/journals/10.1075/ml.9.3.04sha, doi:https://doi.org/10.1075/ml.9.3.04sha.

[178] Fatemeh Shirazi, Matthias Goehring, and Claudia Diaz. Tor Experimentation Tools. In *2015 IEEE Security and Privacy Workshops*, pages 206–213, San Jose, CA, May 2015. IEEE. URL: https://ieeexplore.ieee.org/document/7163227/, doi:10.1109/SPW.2015.20.

[179] Fatemeh Shirazi, Milivoj Simeonovski, Muhammad Rizwan Asghar, Michael Backes, and Claudia Diaz. A Survey on Routing in Anonymous Communication Protocols. *arXiv:1608.05538 [cs]*, August 2016. URL: http://arxiv.org/abs/1608.05538, arXiv:1608.05538.

[180] V. Shmatikov. Probabilistic analysis of anonymity. In *Proceedings 15th IEEE Computer Security Foundations Workshop. CSFW-15*, pages 119–128, June 2002. doi:10.1109/CSFW.2002.1021811.

[181] Howard Simkevitz. Why Privacy Matters in Health Care Delivery: A Value Proposition. In *2009 World Congress on Privacy, Security, Trust and the Management of e-Business*, pages 193–201, Saint John, New Brunswick, Canada, August 2009. IEEE. URL: http://ieeexplore.ieee.org/document/5341698/, doi:10.1109/CONGRESS.2009.16.

[182] Singh, Sukhbir. *Large-Scale Emulation of Anonymous Communication Networks*. Master's Thesis, University of Waterloo, 2014. URL: http://hdl.handle.net/10012/8642.

[183] Singh, Sukhbir. Sunsetting Tor Messenger, April 2018. URL: https://blog.torproject.org/sunsetting-tor-messenger.

[184] Noah A. Smith. Adversarial Evaluation for Models of Natural Language. *CoRR*, abs/1207.0245, 2012. URL: http://arxiv.org/abs/1207.0245, arXiv:1207.0245.

[185] Daniel J Solove. I've got nothing to hide and other misunderstandings of privacy. *San Diego L. Rev.*, 44:745, 2007.

[186] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[187] Latanya Sweeney. Achieving k-Anonymity Privacy Protection Using Generalization and Suppression. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10:571–588, 2002.

[188] Latanya Sweeney. Only you, your doctor, and many others may know. *Technology Science*, 2015092903(9):29, 2015. URL: https://techscience.org/a/2015092903.

[189] Paul Syverson. Why I'm not an Entropist. In Bruce Christianson, James A. Malcolm, Vashek Matyáš, and Michael Roe, editors, *Proceedings of Security Protocols XVII: 17th International Workshop, April 2009, Revised Selected Papers*, pages 231–239. Springer-Verlag, LNCS 7028, 2013.

[190] Paul Syverson, Gene Tsudik, Michael Reed, and Carl Landwehr. Towards an Analysis of Onion Routing Security. In Hannes Federrath, editor, *Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability Berkeley, CA, USA, July 25–26, 2000 Proceedings*, pages 96–114. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001. URL: https://doi.org/10.1007/3-540-44702-4_6, doi:10.1007/3-540-44702-4_6.

[191] Tanti, Marc. Thoughts on lexical substitution: A contextual thesaurus, October 2014. URL: https://geekyisawesome.blogspot.com/2014/10/thoughts-on-lexical-substitution.html.

[192] Wilson L Taylor. "Cloze procedure": A new tool for measuring readability. *Journalism Bulletin*, 30(4):415–433, 1953.

[193] The Freehaven project. Anonymity Bibliography. URL: https://www.freehaven.net/anonbib/full/date.html.

[194] The Tor Project. Chutney. URL: https://gitweb.torproject.org/chutney.git/tree/README.

[195] The Tor Project. Tor Metrics - Traffic. Technical report, The Tor Project. URL: https://metrics.torproject.org/bandwidth-flags.html.

[196] The Tor Project. Strength in Numbers: An Entire Ecosystem Relies on Tor, December 2018. URL: https://blog.torproject.org/strength-numbers-entire-ecosystem-relies-tor.

[197] Tobias Sterbak. Named Entity Recognition with BERT, October 2018. URL: https://www.depends-on-the-definition.com/named-entity-recognition-with-bert/.

[198] Lisa Torrey and Jude Shavlik. Transfer learning. In *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, pages 242–264. IGI Global, 2010.

[199] Kevin Townsend. Privacy Fears Raised Over Facebook Messaging Apps Integration, January 2019. URL: https://www.securityweek.com/privacy-fears-raised-over-facebook-messaging-apps-integration.

[200] Unger, Nik and Goldberg, Ian. NetMirage. Qatar University. URL: https://crysp.uwaterloo.ca/software/netmirage/.

[201] Sharmila Devi V, S. Kannimuthu, Ravikumar G, and Anand Kumar M. KCE_DAlab@MAPonSMS-FIRE2018: Effective word and character-based features for Multilingual Author Profiling. In *FIRE (Working Notes)*, volume 2266 of *CEUR Workshop Proceedings*, pages 213–222. CEUR-WS.org, 2018.

[202] José Van Dijck. Datafication, dataism and dataveillance: Big Data between scientific paradigm and ideology. *Surveillance & Society*, 12(2):197–208, 2014.

[203] Marten Van Dijk and Ari Juels. On the Impossibility of Cryptography Alone for Privacy-preserving Cloud Computing. In *Proceedings of the 5th USENIX Conference on Hot Topics in Security*, HotSec'10, pages 1–8, Washinton, DC, 2010. USENIX Association. URL: http://dl.acm.org/citation.cfm?id=1924931.1924934.

[204] Vineet Abraham. Foodie Favorites, 2017. URL: https://github.com/vabraham/foodie_favorites.

[205] Luis Von Ahn, Manuel Blum, Nicholas J Hopper, and John Langford. CAPTCHA: Using hard AI problems for security. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 294–311. Springer, 2003.

[206] Isabel Wagner and David Eckhoff. Technical Privacy Metrics: A Systematic Survey. *CoRR*, abs/1512.00327, 2015. URL: http://arxiv.org/abs/1512.00327, arXiv:1512.00327.

[207] Wagner, James. Facebook Ironically Blocks Internet Pioneer for Using Long-Established Pseudonym "R.U. Sirius" on Facebook, October 2015. URL: https://nwn.blogs.com/nwn/2015/10/ru-sirius-facebook-ken-goffman.html.

[208] Waldman, Annie and Ornstein, Charles. Few Consequences For Health Privacy Law's Repeat Offenders. *ProPublica*, December 2015. URL: https://www.propublica.org/article/few-consequences-for-health-privacy-law-repeat-offenders.

[209] Alex Wang and Kyunghyun Cho. BERT has a Mouth, and It Must Speak: BERT as a Markov Random Field Language Model. *arXiv preprint arXiv:1902.04094*, 2019.

[210] Samuel D Warren and Louis D Brandeis. Right to privacy. *Harv. L. Rev.*, 4:193, 1890.

[211] Alma Whitten and J D Tygar. Why Johnny Can't Encrypt - A Usability Evaluation of PGP 5.0. In L Cranor and G Simson, editors, *Security and Usability: Designing Secure Systems That People Can Use*, pages 679–702. O'Reilly, 2005.

[212] Terry Williams. Simulating the man-in-the-loop. *OR Insight*, 9(4):17–21, October 1996. URL: https://doi.org/10.1057/ori.1996.20, doi:10.1057/ori.1996.20.

[213] Philipp Winter, Roya Ensafi, Karsten Loesing, and Nick Feamster. Identifying and characterizing Sybils in the Tor network. In *USENIX Security*. USENIX, 2016. URL: https://nymity.ch/sybilhunting/pdf/sybilhunting-sec16.pdf.

[214] Matthew K Wright, Micah Adler, Brian Neil Levine, and Clay Shields. An Analysis of the Degradation of Anonymous Protocols. In *NDSS*, volume 2, pages 39–50, 2002.

[215] Stephen J Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015.

[216] A. C. Yao. Protocols for secure computations. In *23rd Annual Symposium on Foundations of Computer Science (Sfcs 1982)*, pages 160–164, November 1982. doi:10.1109/SFCS.1982.38.

[217] Lin Yuan, Pavel Korshunov, and Touradj Ebrahimi. Privacy-preserving photo sharing based on a secure JPEG. In *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 185–190. IEEE, 2015.

[218] Yurkiewicz , Ilana. Paper Trails: Living and Dying With Fragmented Medical Records. *Undark*, September 2018. URL: https://undark.org/article/medical-records-fragmentation-health-care/.

[219] Jennifer Zamora. I'm Sorry, Dave, I'm Afraid I Can't Do That: Chatbot Perception and Expectations. In *Proceedings of the 5th International Conference on Human Agent Interaction - HAI '17*, pages 253–260, Bielefeld, Germany, 2017. ACM Press. URL: http://dl.acm.org/citation.cfm?doid=3125739.3125766, doi:10.1145/3125739.3125766.

[220] Mark Zuckerberg. A Privacy-Focused Vision for Social Networking, March 2019. URL: https://www.facebook.com/notes/mark-zuckerberg/a-privacy-focused-vision-for-social-networking/10156700570096634/.