THE UNIVERSITY OF
SYDNEY

INVESTIGATION OF THE SECURITY IMPACT OF INTERNET CENTRALIZATION

Author: Niousha Nazemi

Lead-Supervisor: Prof. Albert Y. Zomaya

Co-Supervisors: Prof. Ralph Holz and Dr. Omid Tavallaie

A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy
in the School of Computer Science at
The University of Sydney

April 2025

Statement of Originality

This is to certify that to the best of my knowledge, the content of this thesis is my own work. This thesis has not been submitted for any degree or other purposes.

I certify that the intellectual content of this thesis is the product of my own work and that all the assistance received in preparing this thesis and sources have been acknowledged.

Niousha Nazemi Signature: Date:

Abstract

This thesis investigates how Internet centralization impacts the security of online services, focusing on three problems: digital divides in Australian government services, DNS dependencies of government domains, and privacy in Federated Learning (FL), where each addresses distinct but interconnected aspects of centralization. First, we analyze the DNS dependencies of Australian government domains, identifying potential disparities in service availability between the general population and indigenous communities. By categorizing DNS providers into leading, non-leading, and government-hosted groups, we expose how the digital divide contributes to service unavailability for indigenous domains and increases their vulnerability to outages and attacks. We construct a dataset of Australian government domains to retrieve their DNS providers. Second, we map direct and indirect dependencies through dependency graphs and provide the IP geolocation of DNS providers. We then introduce attacker models by categorizing the attackers' resources and intentions to analyze the implications of DNS dependencies on the vulnerability of different domain groups. Lastly, we address privacy concerns in FL systems, where centralized model aggregation leads to model inversion attacks. We propose ACCESS-FL, a secure aggregation protocol with communication and computation costs as O(1). Experimental results on benchmark datasets (MNIST, FMNIST, and CIFAR) demonstrate that ACCESS-FL significantly reduces computation and communication overhead compared to state-of-the-art methods (Google's SecAgg and SecAgg+) while maintaining comparable model accuracy in honest-but-curious scenarios. This makes ACCESS-FL particularly suitable for large-scale, stable FL environments, such as healthcare systems. In conclusion, this thesis analyses the security consequences of centralization across DNS infrastructure and FL systems to enhance the availability and privacy of online services.

Dedicated to mom and dad: Shohreh and Darab.

Acknowledgements

Foremost, I would like to deeply thank my PhD supervisors, **Prof. Albert Y. Zomaya**, **Prof. Ralph Holz**, and **Dr. Omid Tavallaie** for their continuous support, guidance, and motivation throughout my PhD study. They provided insightful feedback, patiently reviewed my work multiple times, and helped me to step-by-step refine my research approach from identifying research questions to writing. I deeply express my appreciation for Prof. Albert's ongoing support, guidance, and motivation. Prof. Ralph, despite the challenges of remote collaboration, consistently provided his time and support. Dr. Omid established a collaborative team of fellow PhD students and generously dedicated time for each of us throughout the week to exclusively guide us.

Joining Prof. Albert's research group has been the fortune and honor of my life. I am grateful to have had the opportunity to collaborate with distinguished academics, Dr. Anna Maria Mandalari, Dr. Kanchana Thilakarathna, and Prof. Hamed Haddadi. Their insightful feedback on my publications has been invaluable. I would also like to thank fellow PhD students in Prof. Albert's research group.

Additionally, I wish to express my gratitude to Assoc. Prof. Xiu Wang and Dr. Clément Canonne for their insightful feedback through constructive Progress Evaluation Meetings, which played a significant role in shaping my research progress.

I am profoundly grateful to **The University of Sydney** for the financial support provided through the *Postgraduate Research Scholarship in Cyber Security Research and Engagement*. I would also like to extend my sincere thanks to Mr. Lyndon McKevitt, Dr. Arash Araghi, and Ms. Jet Hunt for their support throughout my candidature.

I owe a special debt of gratitude to my parents and my brother, Ariyan, for their unconditional love and support from afar. I am also deeply thankful to Soroush for being a constant source of motivation and encouragement throughout this journey. And I am thankful to all my friends for their inducement. I could not have been able to complete this journey without them.

Lastly, I thank all the staff of the School of Computer Science for creating a conducive environment for research.

List of Publications

This thesis contains material submitted or accepted for publication as follows:

- Chapter 4 of this thesis has been published as [1] and [2] in the *Workshop on Transparency, Accountability, and User Control for a Responsible Internet at the 28th European Symposium on Research in Computer Security*, and *IETF/IAB Workshop on Barriers to Internet Access of Services (biasws)*, respectively. I performed the majority of the research and technical writing presented in the aforementioned papers.
- Chapter 5 of this thesis has been submitted as [3] in *IEEE Journal of Transactions on Network and Service Management* (**Q1 ranked in SJR**). I performed the majority of the research and technical writing presented there.
- Chapter 6 of this thesis has been published as [4] in the *54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks* (**A ranked in CORE)** and submitted as [5] to the *The International Conference on Web Services 2025* (**A ranked in CORE**), respectively. I performed the majority of the research and technical writing presented there. Moreover, I was the co-author of [6], related to the contents of this chapter, which has been published in the *International Conference on Web Services 2024*. I am also the co-author of [7], related to the contents of this chapter, which we plan to submit to the *28th European Conference on Artificial Intelligence* (**A ranked in CORE**).

I certify that the aforementioned authorship attribution statements are correct and I have received permission from the other authors to include the published materials.

Niousha Nazemi Signature: Date:

As supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Albert Y. Zomaya Signature: Date:

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Internet has evolved significantly since its emergence, transitioning from a decentralized architecture with distributed control to a more centralized one. Centralization, in this context, refers to the concentration of control and resources in the hands of a few dominant companies. This centralization and consolidation significantly influence many aspects of the global Internet infrastructure, which has given rise to new security challenges. One of these major is the presence of single points of failure, which has been observed in various incidents where the failure of centralized services led to major outages, including independent services, and revealed the vulnerability of the current centralized architecture. This thesis focuses on analyzing the impact of Internet centralization on the security of online services, especially examining two core security concepts: availability and data privacy. For the former, we investigate how centralization contributes to a digital divide in the availability of Australian government services. Subsequently, we will broaden our scope and investigate the wider impact of DNS dependencies on service availability. For the latter, we investigate the potential of secure aggregation protocols to address privacy concerns arising from a centralized model aggregation in federated learning.

The centralization of the Domain Name System (DNS) is a significant concern in the context of Internet centralization. DNS is an infrastructure protocol on the Internet

that translates domain names into IP addresses, enabling users to access websites using human-readable names. However, the dominance of a few large providers in the DNS space has introduced potential security weaknesses. For domains that rely on a single DNS provider, a distributed denial-of-service (DDoS) attack targeting that single provider can simultaneously disrupt access to many sites. The concentration of power in the hands of a few DNS providers can increase the impact of widespread outages, and it hence raises concerns about the availability and resilience of online services. Our research aims to investigate the DNS dependencies of Australian government domains to analyze the potential impact of centralization on service availability. This may be particularly grave in the case of already disadvantaged communities; hence, we pay particular attention to these.

Focusing on the Australian geographical location, the Australian government provides a variety of digital services to its citizens, such as healthcare and education, in addition to dedicated support services for indigenous communities. The availability of these services depends on the reliability and quality of the underlying DNS infrastructure. We examine the DNS services of Australian government domains, analyzing the types of DNS providers used by domains serving the general population and those serving indigenous communities. By investigating the distribution of DNS providers across these two groups of domains, we first investigate disparities in the quality and reliability of DNS services that could indicate a digital divide. We then explore whether the choice of DNS providers differs more generally between government domains targeting the general public and those explicitly catering to indigenous communities and how this might impact the accessibility and availability of critical online services and their security.

While the centralization of the Internet creates challenges in the availability of services, another aspect of centralization emerges in the realm of data privacy. In the context of federated learning, a machine learning approach that enables the training of models across multiple decentralized devices while keeping the data localized, the centralization of model aggregation poses privacy risks. The central server in federated learning has access to all the model updates from the clients, which can potentially lead to privacy breaches through model inversion attacks. These attacks aim to reconstruct

the original training data from the shared model updates, compromising the confidentiality of sensitive information. We focus on enhancing Google's secure aggregation protocols, a critical component in federated learning designed to protect the privacy of participants during the model aggregation process. Hence, this research also focuses on addressing the data privacy issue caused by the federated learning system, where a central server tries to infer a client's sensitive data by reverse-engineering the client's weights. We propose a lightweight, secure aggregation protocol to prevent the central server from accessing the raw model weights of each client.

1.1 Aims and Research Questions

This thesis aims to provide an analysis of the impact of centralization on the security of online services, focusing on two core security concepts: availability and privacy. Our research is guided by the following research questions:

1. To what extent can Internet centralization contribute to a digital divide in terms of disparities in the availability of services websites for Australian government domains for the general population and indigenous communities?
2. How do DNS dependencies affect service availability, and what are the vulnerabilities of these services against various types of cyber-attacks?
3. How can secure aggregation protocols in federated learning be enhanced, and what improvements can be made to address privacy concerns arising from the centralization of model aggregation?

To address these research questions, the thesis is structured into the following contribution chapters:

- **Chapter 4** addresses *Research Question 1* by analyzing the DNS providers of Australian government domains serving the general population and those serving indigenous communities. The chapter examines the distribution of DNS providers across these two groups of domains to identify any disparities in the types of DNS providers

used, thereby investigating the presence of a digital divide in terms of DNS infrastructure.

- **Chapter 5** explores the impact of direct and indirect DNS dependencies on service availability and the vulnerability of these services to different types of attackers to address *Research Question 2*. The chapter introduces attacker models based on their strength, resources, knowledge of attack, and intention. By analyzing the direct and indirect dependencies and the geographical IP location of DNS providers, the chapter assesses which providers are more vulnerable to specific types of attackers and how their dependent services, including Australian government domains, would be impacted in terms of service unavailability due to DNS provider disruptions. Additionally, the chapter examines the extent to which the DNS services of Australian government domains are located inside or outside Australia. The chapter also takes into account the differences between sites for the general population and those for the indigenous communities in Australia.
- **Chapter 6** focuses on the privacy issues that arise from the centralization of model aggregation in federated learning to answer *Research Question 3*. In federated learning, the centralized server has access to the raw model updates from the clients, which can potentially lead to privacy breaches through model inversion attacks. The chapter improves the client's data privacy through the aggregation process by enhancing Google's secure aggregation protocol in terms of optimized communication and computation. By employing secure aggregation techniques, the chapter aims to preserve data privacy while still enabling the benefits of federated learning.

The overarching research question that ties these chapters together is: "How does Internet centralization impact the different aspects of security in online services, and what are the implications for security?" Each chapter contributes to answering this question by addressing different aspects of security, that is, availability and privacy in the context of centralization.

1.2 Overview of Methodology

This thesis employs different methodologies to explore the impacts of centralization on the security of online services. Each chapter adopts a specific approach to address the research questions and objectives.

- **Chapter 4** combines desk research and algorithmic techniques to identify Australian government domains serving the general population and indigenous communities of Australia. An algorithmic process is employed to identify the relevant domains by utilizing keyword categories and iterative web searches. The methodology then involves querying the identified Australian government domains to determine the corresponding DNS providers. The chapter focuses on analyzing the DNS providers of the identified domains and categorizes these providers based on their dominance in the market. By investigating the types of DNS providers used by each group, the general population, and indigenous communities, the chapter aims to reveal any disparities in the types of DNS providers used by each group.
- **Chapter 5** expands on the initial findings by exploring both direct and indirect DNS dependencies and their impact on service vulnerability to different types of cyber attackers. By querying the identified DNS providers of Australian government domains across the dependency chains, this chapter then also determines the corresponding DNS providers of the providers. Additionally, the chapter investigates the geographical IP locations of the DNS providers to assess whether the DNS services of Australian government domains are located within Australia or abroad, which could indicate a single point of failure. By introducing a framework of attacker models based on their intentions and resources, the chapter evaluates which DNS dependencies are more susceptible to specific types of attackers. Furthermore, it discusses how different provider diversity settings can help mitigate the impact of each attacker model on each group of population (general and indigenous people) in terms of availability.
- **Chapter 6** focuses on the privacy concerns arising from the centralization of model aggregation in federated learning systems, where a centralized server has access to the model updates from multiple clients. This centralization poses privacy risks, as the server can potentially infer sensitive information about clients' data through

techniques like model inversion attacks. To mitigate these risks, the chapter proposes enhancements to Google's secure aggregation protocol, which aims to protect the privacy of clients' data during the aggregation process. The proposed protocol, called ACCESS-FL (Agile Communication and Computation for Efficient Secure Aggregation in Stable Federated Learning Networks), introduces optimizations to improve communication efficiency and computational performance, making the protocol more suitable for stable network environments where the client dropout is limited, and network variations are low. The methodology involves redesigning the communication patterns and reducing the computational load on both the client and server.

1.3 Significance of Research and Conclusion

This thesis makes contributions to the field of Internet security by providing an analysis of the impact of centralization on the availability of online services and data privacy. The thesis reveals the presence of a digital divide in terms of DNS infrastructure, which has important implications for the accessibility of online services for different population groups. Furthermore, the thesis investigates the vulnerabilities and risks associated with centralized Internet services by exploring both direct and indirect DNS dependencies. By examining the geographical IP locations of DNS providers and introducing a framework of attacker models, the research assesses the vulnerability of different DNS dependency configurations to specific types of attackers. The findings reveal the importance of diversifying DNS providers and implementing appropriate provider diversity settings to enhance resilience against various cyber threats, including DDoS attacks as a recurring threat in environments marked by high centralization, such as the DNS infrastructure discussed above. Because a large share of online services may depend on a single or small set of DNS providers, an attacker can disrupt a substantial fraction of critical websites by overwhelming one central provider with malicious traffic. Protecting against such large-scale disruptions requires analyzing suspicious traffic patterns across multiple organizations (for instance, different DNS

operators or Internet Service Providers). However, sharing raw logs among these organizations poses risks: DNS traffic data is often sensitive, and many providers hesitate to reveal detailed logs that contain user's sensitive information. Regarding the aforementioned challenges, FL approaches can be applicable, in which each DNS operator or ISP can maintain its own data locally while still contributing to a collective defense strategy. In principle, each organization could train a local anomaly detection model on its traffic, using metrics such as query rates or unusual peaks [8], and only send the trained model to the aggregator server. However, if the server received each participant's model updates plainly, it might still piece together private details from certain operators—especially smaller ones that serve vulnerable populations. To avoid this, a secure aggregation protocol is employed to ensure that only aggregated (i.e., summed) updates are visible. Thus, even the central server cannot isolate or reconstruct any one DNS provider's sensitive traffic data. Thus, in chapter 6 of this thesis, we examine the privacy concerns arising from the centralization of model aggregation in federated learning systems. By proposing enhancements to Google's secure aggregation protocol, the research demonstrates an optimized technique that maintains data privacy and the benefits of collaborative learning. The improved secure aggregation protocol offers increased communication efficiency and computational performance, making it more suitable for stable network environments. This contribution has implications for the development of privacy-preserving federated learning systems.

The significance of this research lies in the examination of the impact of centralization on the security and availability of online services, considering both the Australian context and the broader implications for federated learning systems. By investigating the challenges posed by centralization in terms of service availability, DNS dependencies, and data privacy, this thesis contributes to the ongoing discussion on the security and resilience of online services in the centralized Internet environment.

Chapter 2

Background

2.1 Domain Name System (DNS)

The Domain Name System (DNS) [9] is the critical component of the Internet's infrastructure that translates human-readable domain names into machine-readable Internet Protocol (IP) addresses. This translation process is essential for enabling users to access websites and other online resources using easily memorable domain names rather than having to memorize the complex numerical IP addresses that computers use to communicate with each other over the network [10].

2.1.1 DNS Resolution Process

DNS [11, 9] resolution process involves various levels of authoritative servers, each level responsible for a specific part of the domain name. At the highest level of the DNS hierarchy are the root servers [12]. Root servers do not store domain records but instead hold information about the location of Top-Level Domain (TLD) servers. When a DNS resolver receives a query, it first checks its cache for the requested information. If the data is not available in the cache, the resolver starts the resolution

process by querying one of the root servers. The root server responds with the address of the TLD server responsible for managing the domain's extension (such as .com, .net, or .au). There are 13 unique root server addresses operated by various organizations worldwide and distributed via anycast [13]. Anycast is a network addressing and routing methodology where a single IP address is assigned to multiple servers in different geographical locations. When a client sends a request to an anycasted IP address, the request is routed to the nearest server, which helps improve performance and resilience. This results in the existence of hundreds of root server sites, rather than just 13 physical locations. This distributed network provides redundancy and stability to the DNS infrastructure and ensures that the resolution process can continue even if some root servers become unavailable.

The next level in the DNS hierarchy consists of the Top-Level Domain (TLD) servers. TLD servers are responsible for managing the second-highest level of the DNS hierarchy. They handle top-level domains, including generic TLDs (gTLDs) like .com, .org, and .net, as well as country code TLDs (ccTLDs) like .uk for the United Kingdom and .jp for Japan. For Australia, this ccTLD is .au, which serves as a critical part of the country's digital identity and represents Australian businesses, websites, and services on the Internet. The administration of the .au domain is managed by the .au Domain Administration (auDA) [14], the policy authority and industry self-regulatory body for the .au domain space. Within the .au ccTLD, there are several second-level domains (2LDs) designed to cater to different types of entities. For example, .com.au is intended for commercial entities, while .org.au is reserved for non-profit organizations. Other notable 2LDs include .edu.au for educational institutions, .gov.au for government entities, and .net.au for network infrastructure providers. This categorization helps to organize the .au domain space and makes it easier for users to identify the type of entity associated with a particular domain name.

TLD servers do not contain specific domain records. Instead, they store information about the authoritative name servers associated with each domain under their respective TLDs. When a TLD server receives a query from a root server, it forwards the query to the corresponding authoritative name server for the requested domain. Authoritative name servers are responsible for storing the actual DNS records for a domain, such as A (IPv4 address), AAAA (IPv6 address), and MX (mail exchanger) records. When

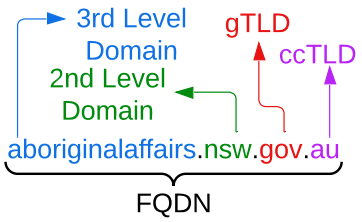

Figure 2.1: An example of the FQDN.

a query reaches an authoritative name server, it provides the specific IP address associated with the requested domain name. Unlike root and TLD servers, authoritative name servers have information records for their respective domains and can respond with the final information needed to complete a DNS query. However, the DNS resolution process may involve multiple levels of authoritative name servers, depending on the domain hierarchy. For example, domains at the third, fourth, or fifth level may have a chain of authoritative servers that need to be queried sequentially. Each level of the domain hierarchy is managed by a different authoritative name server. The final authoritative name server in the chain provides a direct response to the query and eliminates the need for further queries. Below the TLDs, there are the secondary-level domains. The various parts of a Fully Qualified Domain Name (FQDN) are given as the concatenation of so-called labels at the various levels of this hierarchy. Fig. 2.1 shows an example. The DNS resolution process, illustrated in Fig. 2.2, involves multiple steps. In the simplest case, and without caching, this includes: 1) A recursive DNS server initiates the query. 2) It queries a root server, which then directs it to the corresponding TLD server. 3) The TLD server directs the query to the authoritative name server (NS) for the specific queried domain. 4) The authoritative server retrieves the IP address, which is subsequently sent back to the recursive server. User interaction happens via a stub DNS resolver that directs its queries to the recursive server, which is usually provided by an ISP or operated as a public service (e.g., the public resolver of Google).

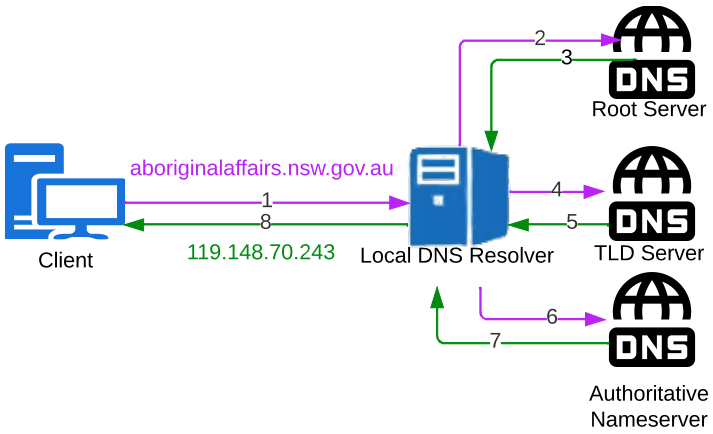

Figure 2.2: DNS resolution process

2.1.2 Internet Centralization

The Internet, initially designed as a decentralized network [15], has undergone a significant shift towards centralization in recent decades. A handful of large corporations and organizations now exert disproportionate control and influence over the online ecosystem [16]. In recent years, efforts to understand how the Internet works and identify critical dependencies have been investigated [17], in terms of how Internet services are set up for different groups have implications for how these groups can access the Internet. In this sense, revealing Internet dependencies can reveal implications for the digital divide. This is also evident in core Internet infrastructure, namely the Domain Name System (DNS). Here, much centralization and consolidation have occurred [18, 19]. This refers to the dominance of a limited number of large service providers who exert significant control over various aspects of the DNS. DNS employs a hierarchical configuration with multiple authoritative name servers to distribute the workload and enhance the name resolution process. However, today, much of this setup is in the hands of very few providers, and major companies such as Microsoft, Amazon, Cloudflare, and Google have significant influence over DNS provisioning. Where operators choose to outsource the operation of their DNS, the implication is that their users also rely on (a possibly limited) number of (possibly centralized) providers to access Internet services of relevance to them. It has been stated that this dependency on a few providers increases the vulnerability to potential attacks and raises concerns about the overall resilience of Internet services [20, 21].

2.1.3 DNS Dependency

DNS dependency refers to the dominance of a limited number of large service providers who exert significant control over various aspects of the DNS. DNS employs a hierarchical configuration with multiple authoritative name servers to distribute the workload and enhance the name resolution process. However, today, much of this setup is in the hands of very few providers, and major companies such as Microsoft, Amazon, Cloudflare, and Google have significant influence over DNS provisioning. Where operators choose to outsource the operation of their DNS, the implication is that their users also rely on (a possibly limited) number of (possibly centralized) providers to access Internet services of relevance to them. It has been stated that this dependency on a few providers increases the vulnerability to potential attacks and raises concerns about the overall resilience of Internet services [20, 21]. DNS is a critical component of the internet infrastructure, enabling the translation of human-readable domain names into machine-readable IP addresses. While the DNS was designed as a distributed system with multiple authoritative name servers to distribute the workload and enhance the name resolution process, the current landscape of DNS provisioning has become increasingly centralized. This centralization has led to a significant dependency on a limited number of large service providers who exert considerable control over various aspects of the DNS [20]. The dominance of a limited number of large service providers that exert significant control over various aspects of the DNS. DNS employs a hierarchical configuration with multiple authoritative name servers to distribute the workload and enhance the name resolution process. However, today, much of this setup is in the hands of very few providers, and major companies such as Microsoft, Amazon, Cloudflare, and Google have significant influence over DNS provisioning. Where operators choose to outsource the operation of their DNS, the implication is that their users also rely on (a possibly limited) number of (possibly centralized) providers to access Internet services of relevance to them. It has been stated that this dependency on a few providers increases the vulnerability to potential attacks and raises concerns about the overall resilience of Internet services [20, 21]. In recent years, major companies such as Microsoft, Amazon, Cloudflare, and Google have gained significant influence over DNS provisioning. These companies operate extensive DNS infrastructure and offer

DNS services to a large number of customers, including domain owners, website operators, and end-users [22]. As a result, a substantial portion of the global DNS traffic flows through the servers and networks controlled by these providers. The concentration of DNS provisioning among a few large providers has several implications for the resilience, security, and privacy of the internet. One major concern is the increased vulnerability to potential attacks. If a large DNS provider experiences a service disruption or becomes the target of a cyberattack, it can have far-reaching consequences for the availability and accessibility of internet services [23]. For example, in October 2016, a Distributed Denial of Service (DDoS) attack on Dyn, a major DNS provider, caused widespread outages and affected numerous popular websites and services, including Twitter, Netflix, and PayPal. Moreover, the centralization of DNS provisioning raises concerns about the overall resilience of internet services. When a significant number of websites and services rely on a single DNS provider, any issues or failures affecting that provider can have a cascading effect, leading to widespread disruptions [21]. This dependency on a few providers creates single points of failure in the DNS ecosystem, making it more vulnerable to both accidental and intentional disruptions. Another aspect of DNS dependency is the potential impact on user privacy. DNS queries contain valuable information about users' online activities, including the websites they visit and the services they use. When a large proportion of DNS traffic is handled by a small number of providers, it enables these providers to collect and analyze vast amounts of user data. This concentration of user data raises concerns about the potential for abuse, such as targeted advertising, user profiling, and surveillance. The centralization of DNS provisioning also has implications for the autonomy and control of internet operators and users. When operators choose to outsource the operation of their DNS to third-party providers, they effectively relinquish control over a critical aspect of their network infrastructure [20]. This reliance on external providers can limit the ability of operators to implement custom DNS configurations, apply security policies, or ensure the privacy of their users' DNS queries.

To address the challenges posed by DNS dependency, various initiatives and technologies have been proposed to promote a more decentralized and resilient DNS ecosystem. One approach is the use of multi-provider DNS architectures, where operators distribute their DNS provisioning across multiple providers to reduce the reliance on any single provider. By leveraging multiple DNS providers, operators can improve

the redundancy and failover capabilities of their DNS infrastructure, mitigating the impact of potential service disruptions or attacks. Another strategy is the adoption of encrypted DNS protocols, such as DNS over HTTPS (DoH) and DNS over TLS (DoT). These protocols aim to protect the privacy of DNS queries by encrypting the communication between users and DNS resolvers [24]. By preventing eavesdropping and tampering, encrypted DNS protocols can help mitigate the privacy risks associated with centralized DNS provisioning. However, the deployment of encrypted DNS also raises concerns about the potential concentration of DNS traffic among a few large providers offering DoH or DoT services.

2.1.4 The Impact of Centralization of DNS Services on Digital Divide

The digital divide is a concept that has attracted significant attention from researchers, policymakers, and society at large over the past two decades. The term was first introduced and defined in the mid-to-late 1990s through a series of reports titled "Falling through the net" [25, 26, 27]. These reports highlighted the disparities in access to and effective use of digital technologies among various segments of the population. At its core, the digital divide refers to the gap between individuals or groups who have access to and effectively use digital technologies, such as the Internet, and those who do not [28]. This divide creates a stark contrast between the advantaged and the disadvantaged, with those who have access to and proficiency in using these technologies being considered privileged, while those who lack access or the necessary skills are at a significant disadvantage [29]. Moreover, the digital divide extends beyond mere access to technology. It also encompasses the skills and knowledge required to effectively navigate and utilize digital tools and platforms. Digital literacy, which refers to the ability to locate, evaluate, and create digital content, is a critical component of bridging the digital divide [30]. Without adequate digital literacy skills, individuals may struggle to fully participate in the digital economy, access educational resources, or engage in civic activities online. The consequences of the digital divide are far-reaching and can have significant implications for individuals, communities, and societies as a whole.

In an increasingly digital world, where access to information, services, and opportunities is increasingly mediated by technology, those who are left behind face numerous challenges. The digital divide can exacerbate existing social and economic inequalities, perpetuating cycles of disadvantage and limiting upward mobility [31]. One of the groups that are particularly vulnerable to the digital divide is economically disadvantaged communities. Socioeconomic factors, such as income and education levels, are strongly correlated with access to and proficiency in using digital technologies [32]. Individuals from low-income households often face barriers to accessing the Internet and acquiring the necessary devices, such as computers or smartphones, due to financial constraints. Additionally, they may lack the support and resources needed to develop digital literacy skills, further widening the gap between the haves and the have-nots. Indigenous communities are another group that faces significant challenges in accessing and effectively utilizing digital technologies [33]. Many indigenous communities, particularly those in remote or rural areas, may lack the infrastructure and connectivity required to participate fully in the digital world. Moreover, cultural and linguistic barriers can further compound the difficulties faced by indigenous peoples in acquiring digital skills and engaging with online content. Addressing the digital divide requires a multifaceted approach that takes into account the various dimensions of the issue. Governments, private sector organizations, and civil society groups have a crucial role to play in bridging the gap and promoting digital inclusion. This can involve initiatives such as expanding broadband infrastructure to underserved areas, providing affordable access to devices and connectivity, and offering digital literacy training programs [34]. Education is a key avenue for addressing the digital divide. Incorporating digital literacy skills into school curricula and providing students with access to technology can help ensure that the next generation is equipped with the necessary tools and knowledge to navigate the digital landscape [35]. Additionally, adult education programs and community-based initiatives can provide opportunities for individuals who may have missed out on digital skills development earlier in life to acquire these competencies. By framing the digital divide as a human rights issue, it becomes clear that addressing this divide is not merely a matter of technological advancement but a question of social justice and equality. The COVID-19 pandemic has further underscored the urgency of addressing the digital divide. With the widespread shift to remote work, online learning, and digital service delivery, those without access to reliable Internet and digital devices have been disproportionately affected [36]. The

pandemic has exposed and exacerbated existing inequalities, highlighting the need for concerted efforts to bridge the digital divide and ensure that no one is left behind in the digital transformation.

2.2 Federated Learning

Neural Networks (NN) [37], a prominent method in Machine Learning (ML) [38], have been widely applied in fields such as Computer Vision (CV) [39] and Natural Language Processing (NLP) [40] for tasks including object detection [41] and text generation [42]. Traditional machine learning, also known as Centralized Learning (CL), involves centralizing data from multiple sources for model training on high-performance servers, which requires direct access to the entire dataset. However, the rapid increase in data volume and diversity generated by mobile and Internet of Things (IoT) devices has made centralized data management and processing increasingly complex and inefficient [43]. In smart healthcare systems, for instance, extensive patient data collected by numerous sensors and devices presents significant challenges for centralized training approaches due to high communication overhead and data privacy concerns [44]. Centralized training methods also risk data leakage, compromising user privacy, particularly with sensitive information like clinical records [45]. To address these issues, McMahan et al. introduced a Federated Learning (FL) framework called Federated Averaging (FedAvg) [46] FL an distributed approach to machine learning where a shared global model is trained under the coordination of a central server by aggregating updates computed locally on clients using their own private data. Unlike traditional centralized learning approaches, the training data is not uploaded to a server, but is instead kept on the clients (e.g. mobile devices or other edge devices) where it is generated. This enables training models on large, sensitive datasets in a privacy-preserving manner. Modern mobile devices have access to a wealth of data suitable for machine learning tasks that can significantly enhance the user experience on the device. For example, language models can improve speech recognition and text entry, while image models can help automatically select good photos. However, this rich data is often privacy sensitive and large in size, making it infeasible to collect and store

on a centralized server for training using conventional machine learning approaches. Federated Learning offers an alternative approach by enabling training models on the data without the data leaving the devices. By bringing the models to the data rather than the data to the models, the privacy of sensitive user data can be better preserved. The decoupling of model training from the need to upload and store the data in the cloud significantly reduces privacy and security risks.

2.2.1 Characteristics of the Federated Learning Setting

There are several key characteristics that distinguish federated optimization problems from traditional distributed optimization:

- **Non-IID Data:** The training data on a given client is typically based on the usage of the mobile device by a particular user, and hence any particular user's local dataset will not be representative of the population distribution. In other words, the data is not independent and identically distributed (non-IID) across clients.
- **Unbalanced Data:** Some users will make much heavier use of the service or app than others, leading to varying amounts of local training data across clients. The data is likely to be unbalanced across clients.
- **Massively Distributed:** The number of clients participating in the optimization is likely to be much larger than the average number of training examples per client. There can be millions of clients, each with small amounts of local data.
- **Limited Communication:** Mobile devices are frequently offline or on slow or expensive connections. Communication is a critical constraint, much more so than in data center settings with reliable, high-throughput networks. The cost of communication in terms of battery power and bandwidth fees must also be considered.

2.2.2 The Federated Averaging Algorithm

Google has proposed a practical Federated Learning algorithm called *Federated Averaging* (FedAvg) that combines local stochastic gradient descent on each client with a server that performs model averaging. The key idea is to perform multiple epochs of SGD on each client's local data before the client sends its updated model to the server. The server then averages the models and the process repeats.

More formally, the goal is to minimize the objective function:

$$\min_{w \in \mathbb{R}^d} f(w) \quad \text{where} \quad f(w) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(w). \tag{2.1}$$

Here w are the parameters of the model, n is the total number of data points across all clients, and f_i is the loss function on the i-th data point. The data is partitioned over K clients, with $\mathcal{P}_k$ the set of indexes of data points on client k, with $n_k = |\mathcal{P}_k|$. The local objective for client k is:

$$F_k(w) = \frac{1}{n_k} \sum_{i \in \mathcal{P}_k} f_i(w). \tag{2.2}$$

The full FedAvg algorithm proceeds as follows:

Initialization: The server randomly initializes the global model w_0.

Client Selection: At each round $t = 1, 2, ...$, the server selects a subset S_t of $m = \max(C \cdot K, 1)$ clients at random, where C is a hyperparameter controlling the fraction of clients that participate in each round.

Broadcast: The server sends the current model w_t to each selected client $k \in S_t$.

Client Update: Each selected client k begins the local update by initializing its local model w^k with the global model parameters w_t, received from the central server. The goal of the client is to minimize its local objective function F_k over its dataset $\mathcal{P}_k$.

To do this, the client performs E local epochs of training. During each epoch, the client processes the data in minibatches of size B. For each minibatch b, the model parameters w^k are updated using Stochastic Gradient Descent (SGD) as follows:

$$w^k \leftarrow w^k - \eta \nabla \ell(w^k; b). \tag{2.3}$$

where η is the learning rate, $\ell(w^k; b)$ represents the loss function computed over the minibatch b, and $\nabla \ell(w^k; b)$ is the gradient of the loss with respect to the model parameters w^k. After completing all E epochs, the updated model parameters w^k are sent back to the central server for aggregation.

Aggregation: Each client k sends its updated model w^k_{t+1} back to the server, which takes a weighted average to get the updated global model:

$$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w^k_{t+1}. \tag{2.4}$$

The algorithm relies on three key parameters that control the amount of computation and communication: - C: The fraction of clients selected to participate in each round - E: The number of epochs each client performs locally before sending its updates - B: The minibatch size used for the local client updates Increasing C increases the amount of multi-client parallelism, allowing updates from more clients to be incorporated in each round. Increasing E and decreasing B results in more local SGD steps being taken by each client between communication rounds. The FedAvg algorithm was evaluated empirically on a variety of model architectures and datasets:

MNIST: Two models were trained on the MNIST handwritten digit recognition task - a multilayer perceptron (199,210 parameters) and a CNN (1,663,370 parameters).

The data was partitioned over 100 clients in both an IID manner and a pathologically non-IID manner where each client only had examples of two digits.

CIFAR-10: A CNN with about 1 million parameters was trained on the CIFAR-10 image classification dataset, partitioned over 100 clients in an IID manner.

Shakespeare: A stacked character-level LSTM model was trained for next character prediction on a dataset constructed from the works of Shakespeare. The data was partitioned by assigning each speaking role to a different client, resulting in a naturally unbalanced and non-IID distribution.

Large-Scale Language Modeling: A large LSTM was trained for next word prediction on a corpus of public social media posts, with the data grouped by user for a total of over 500,000 clients.

The experiments explored the impact of the key FedAvg parameters (C, E, B) on the number of communication rounds needed to achieve a target accuracy. The results showed that FedAvg can significantly reduce the number of communication rounds compared to federated SGD (FedSGD) which takes a single gradient step on each client before averaging (equivalent to FedAvg with $E = 1$, $B = \infty$).

For example, on the MNIST CNN, using $C = 0.1$, $E = 5$ and $B = 10$ allowed reaching 99% accuracy in 173 rounds, a 2.8x speedup over FedSGD. On the Shakespeare LSTM with $C = 0.1$, $E = 5$ and $B = 10$, FedAvg reached the target accuracy in 46x fewer rounds than FedSGD.

Notably, the convergence speedups were even more dramatic for the naturally non-IID Shakespeare data compared to the pathologically partitioned MNIST data. FedAvg was able to reach the target in 95x fewer rounds than FedSGD on the Shakespeare data (vs 8.6x fewer for pathological non-IID MNIST). The large-scale language modeling experiment demonstrated the practicality of FedAvg for a realistic production use case. Using $C = 0.04$, $E = 1$ and $B = 8$, FedAvg trained an LSTM with over 4 million parameters on data from over 500,000 users, reaching the target accuracy in 35 rounds - 23x faster than FedSGD.

The empirical results reveal the ability of FedAvg to train high-quality models on non-IID and unbalanced data distributions in a communication-efficient manner. The algorithm's robustness stems from the fact that model averaging produces models that perform well on the full data distribution even when individual clients' local models may be biased toward their local data. There are several important directions on Federated Learning algorithms:

Differential Privacy: While keeping the training data on client devices provides inherent privacy benefits, incorporating techniques from differential privacy could provide stronger, more formal privacy guarantees. Recent work has explored combining federated learning with secure aggregation protocols and differential privacy.

Personalization: The global models produced by federated learning aim to perform well on the overall population distribution. In some applications, it may be preferable to personalize models to individual users. Techniques for combining global and per-user personalized models is an active area of research.

Non-Convex Optimization: FedAvg has demonstrated strong empirical performance on non-convex objectives like deep networks, but providing theoretical guarantees for non-convex federated optimization remains an open challenge.

Productionization: Deploying federated learning in real-world production systems requires tackling issues like client failures, concept drift in client data distributions over time, and communication and computation constraints for edge devices. Ongoing work is translating federated learning research into practical production systems.

2.3 Secure Aggregation Protocol Implemented By Google

While FL protects user privacy by avoiding direct data sharing, it still faces challenges and vulnerabilities, such as the model inversion attacks [47] that can be applied by an

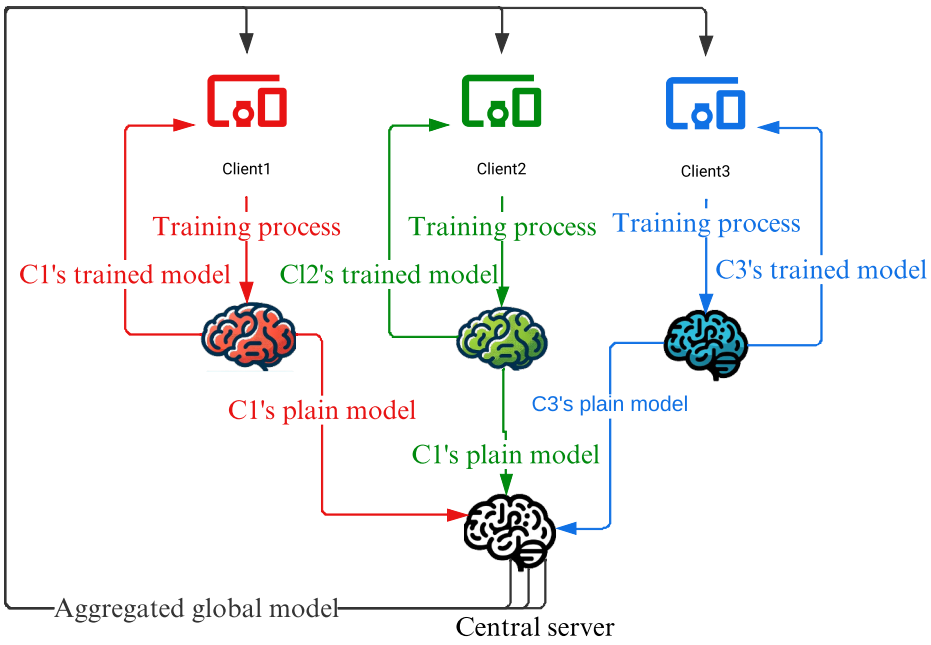

Figure 2.3: Federated Learning without Secure Aggregation.

honest-but-curious server [48, 49] to reconstruct the original client's data by reverse-engineering the local model weights. To address these privacy threats, Google proposed the Secure Aggregation (SecAgg) protocol [50] as a secure multi-party computation (MPC) [51] method based on Diffie-Hellman (DH) key agreement [52] and Shamir's secret sharing [53, 54]. The primary objective of secure aggregation is that the server generates the global model through the aggregation phase without having access to the raw local updated models received from clients. Hence, the privacy of clients is maintained in this approach. Clients in their protocol include mobile devices that arbitrarily drop out of the network.

Figures 2.3 and 2.4 illustrate federated learning schemes with and without secure aggregation. As it is shown in Fig. 2.3, the server has direct access to the individual updates from each client and generates the global model based on the plain client's updated model. However, in 2.4, the updated model from each client is not directly accessible by the server. So, the server does the aggregation based on not knowing the individual updates.

In SecAgg, there are multiple clients that each have their locally trained model. These clients collaborate to allow a central server to compute the aggregate sum of their

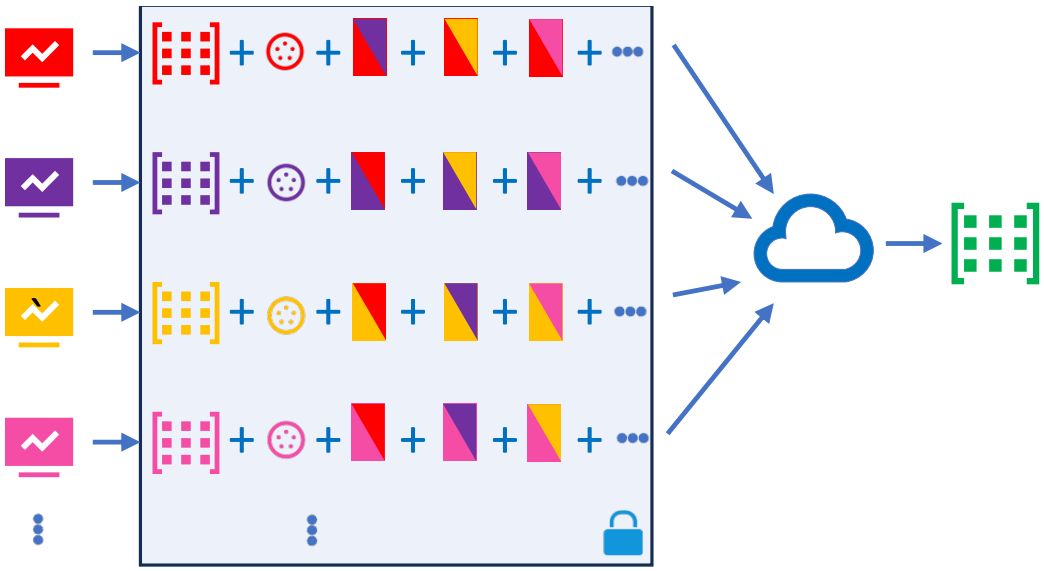

Figure 2.4: Federated Learning with Secure Aggregation.

updates. The objective of the protocol is to enable the server to aggregate the sum of clients' updated models from at least a threshold number of participants without revealing the individual models of any single client. The output of the secure aggregation protocol, in terms of the aggregated global model, is equivalent to what would be achieved if no secure aggregation was involved. This means that while the protocol preserves the privacy of individual updates, the final aggregated result (the global model) is the same as it would be if each client's updated model were openly accessible to the central server. The detailed description of the secure aggregation protocol is as follows:

The secure aggregation protocol developed by Google, as detailed in [50] is structured into a setup phase followed by four main rounds. This initial phase is critical to prepare the prerequisites of the algorithm. The main four rounds are the secure aggregation protocol's core, where the actual aggregation process occurs.

Setup: The protocol is run between a server S and a set of n users $\mathcal{U} = u_1, \ldots, u_n$. It is parameterized by a security parameter k, a threshold $t \leq n$, and the dimensions m and range $[0, R)$ of the data vectors. All arithmetic is assumed to be done modulo R. Users also agree on a finite field $\mathbb{F}$ for Shamir secret sharing. Additionally, the protocol makes use of the following cryptographic primitives, which are assumed to be secure:

- A Key Agreement protocol Π_{KA} enabling each pair of users to agree on a shared secret key.
- A Pseudorandom Generator PRG mapping seeds to vectors in $[0, R)^m$.
- A Symmetric Authenticated Encryption scheme Π_{AE} for users to encrypt messages to each other.
- A Signature scheme Π_{SIG} allowing users to authenticate messages.

Users are assumed to have private and authenticated channels with the server. In the active adversary model, users are additionally assumed to have a Public Key Infrastructure (PKI) allowing them to verify the identities associated with public keys. Each user u_i has a private data vector $x_i \in \mathbb{Z}R^m$. The goal is to compute $\sum i = 1^n x_i$ without revealing any x_i except in aggregate.

Advertising Keys: In Round 0, each user u_i generates two key pairs $(c_i^{\text{PK}}, c_i^{\text{SK}}) \leftarrow \Pi_{\text{KA}}.\text{Gen}(1^k)$ and $(s_i^{\text{PK}}, s_i^{\text{SK}}) \leftarrow \Pi_{\text{KA}}.\text{Gen}(1^k)$. The user sends $(c_i^{\text{PK}}, s_i^{\text{PK}})$ to the server. The server collects these messages from at least t users (otherwise it aborts). Let $\mathcal{U}1$ denote this subset of users. The server broadcasts the set $(u_j, c_j^{\text{PK}}, s_j^{\text{PK}})u_j \in \mathcal{U}_1$ to all users in $\mathcal{U}_1$.

Sharing Keys: In Round 1, each user $u_i \in \mathcal{U}_1$ first verifies the received set from the server - that it contains at least t users, all public key pairs are unique, and all signatures verify under the PKI. If not, the user aborts. Next, the user samples a random seed $b_i \leftarrow \mathbb{F}$. It then secret shares its private key s_i^{SK} as $(u_j, s_{i,j}^{\text{SK}})_{u_j \in \mathcal{U}1} \leftarrow \text{SS.Share}(s_i^{\text{SK}}, t, \mathcal{U}1)$ and secret shares the seed as $(u_j, bi, j)u_j \in \mathcal{U}_1 \leftarrow \text{SS.Share}(b_i, t, \mathcal{U}_1)$. For each other user $u_j \in \mathcal{U}1 \setminus u_i$, the user encrypts the shares as:

$$e_{i,j} \leftarrow \Pi_{\text{AE}}.\text{Enc}\left(\Pi_{\text{KA}}.\text{Agree}\left(c_i^{\text{SK}}, c_j^{\text{PK}}\right), s_{i,j}^{\text{SK}} \,||\, b_{i,j}\right). \tag{2.5}$$

If any operation fails, the user aborts. Otherwise, it sends all ciphertexts $e_{i,j}$ to the server. The server collects ciphertexts from at least t users (otherwise it aborts). Let $\mathcal{U}_2 \subseteq \mathcal{U}1$ denote this subset. For each $u_i \in \mathcal{U}2$, the server sends the set $ej, iu_j \in \mathcal{U}_2$ to u_i.

Masking and Collecting Inputs: In Round 2, each user $u_i \in \mathcal{U}2$ first decrypts the received ciphertexts as $(u'_j||u'_i||sj, i^{\text{SK}}||b_{j,i}) \leftarrow \Pi_{\text{AE}}.\text{Dec}(\Pi_{\text{KA}}.\text{Agree}(c_i^{\text{SK}}, c_j^{\text{PK}}), e_{j,i})$ and verifies that $u'_i = u_i$ and $u'_j = u_j$. If any decryption fails or the server sent fewer than t ciphertexts, the user aborts. Next, for each $u_j \in \mathcal{U}2 \setminus u_i$, the user computes a pairwise mask as:

$$p_{i,j} = \text{PRG}(\Pi_{\text{KA}}.\text{Agree}(s_i^{\text{SK}}, s_j^{\text{PK}})) \quad \text{if } i > j, \tag{2.6}$$

$$p_{i,j} = -\text{PRG}(\Pi_{\text{KA}}.\text{Agree}(s_i^{\text{SK}}, s_j^{\text{PK}})) \quad \text{if } i < j. \tag{2.7}$$

The user also computes a private mask $p_i = \text{PRG}(b_i)$. It then masks its private input as:

$$y_i = x_i + p_i + \sum_{u_j \in \mathcal{U}2} pi, j. \tag{2.8}$$

If any operation fails, the user aborts. Otherwise, it sends y_i to the server. The server collects y_i from at least t users (otherwise it aborts). Let $\mathcal{U}_3 \subseteq \mathcal{U}_2$ denote this subset. The server sends $\mathcal{U}_3$ to each user in $\mathcal{U}_3$.

Unmasking: In Round 4, each user $u_i \in \mathcal{U}_4$ verifies that $\mathcal{U}_4 \subseteq \mathcal{U}_3$, $|\mathcal{U}4| \geq t$, and that $\Pi\text{SIG}.\text{Verify}(d_j^{\text{PK}}, \mathcal{U}_3, \sigma'_j) = 1$ for all $u_j \in \mathcal{U}_4$ (otherwise it aborts). The user sends to the server the set of shares $s_{j,i}^{\text{SK}} : u_j \in \mathcal{U}_2 \setminus \mathcal{U}3 \cup bj, i : u_j \in \mathcal{U}_3$. The server collects responses from at least t users in $\mathcal{U}_4$ (denote this set $\mathcal{U}_5$). For each $u_j \in \mathcal{U}2 \setminus \mathcal{U}3$, it reconstructs $s_j^{\text{SK}} \leftarrow \text{SS.Recon}(sj, i^{\text{SK}} u_i \in \mathcal{U}5, t)$ and recomputes pi, j for all $u_i \in \mathcal{U}_3$ using the PRG. For each $u_j \in \mathcal{U}3$, the server reconstructs $b_j \leftarrow \text{SS.Recon}(bj, i_{u_i \in \mathcal{U}_5}, t)$ and recomputes p_j using the PRG. Finally, the server computes the sum of the private inputs as:

$$z = \sum_{u_i \in \mathcal{U}3} x_i = \sum u_i \in \mathcal{U}3 y_i - \sum u_i \in \mathcal{U}3 p_i + \sum {}_{u_i \in \mathcal{U}_3 \ u_j \in \mathcal{U}_2 \setminus \mathcal{U}3} pj, i. \tag{2.9}$$

Protocol Summary The main points in SecAgg is as:

- Each pair of users (u_i, u_j) agrees on a pairwise random mask $p_{i,j} = -p_{j,i}$ that cancels out when the server computes the final sum.

- Each user u_i adds to its input x_i all its pairwise masks $p_{i,j}$ as well as a private mask p_i. This hides the input from the server.
- If a user drops out after Round 1, the other users send shares of its private key and PRG seed to the server, allowing the server to remove that user's uncancelled masks from the sum.
- Each user secret shares both its pairwise masks and private mask, to handle the case where it drops out after sending its masked input.
- A consistency check in Round 3 ensures the server cannot trick users into unmasking with different user sets.
- As long as at least t users remain in each round, the server can always unmask and compute the final sum, guaranteeing robustness.

In the honest-but-curious model, this protocol guarantees that the server only learns the sum $\sum_{i=1}^{n} x_i$ and nothing else about the individual x_i's, while each user learns nothing. In the active adversary model, a coalition of up to $t - 1$ users cannot learn anything beyond their own inputs and the final sum, even if they collude with the server.

2.3.1 Diffie-Hellman Algorithm

Diffie-Hellman (DH) is a seminal cryptographic algorithm that enables secure key exchange over an insecure channel. Developed by Whitfield Diffie and Martin Hellman and published in their 1976 paper, "New Directions in Cryptography" [52], the DH algorithm has become a foundation of modern cryptography. The algorithm's security relies on the computational difficulty of the discrete logarithm problem (DLP) [55], which ensures the confidentiality of the exchanged keys. The original DH algorithm, now referred to as the "classical" or "finite field" Diffie-Hellman (FF-DH), operates within a multiplicative group of integers modulo a prime number. The security of FF-DH is based on the presumed hardness of the DLP in finite fields [56]. The DH key exchange protocol allows two parties, commonly referred to as Alice and Bob, to establish a shared secret key over an insecure communication channel without any prior knowledge of each other's secret information. The DH key exchange protocol can be described in the following steps: Alice and Bob agree on a large prime number p and

a generator g of the multiplicative group of integers modulo p. Alice selects a secret integer a and computes $A = g^a \bmod p$. Similarly, Bob selects a secret integer b and computes $B = g^b \bmod p$. Alice sends A to Bob, and Bob sends B to Alice over the insecure channel. Alice computes the shared secret key $K = B^a \bmod p$, while Bob computes the same key as $K = A^b \bmod p$. The security of the DH protocol is based on the assumption that it is computationally infeasible for an eavesdropper, who knows p, g, A, and B, to calculate the shared secret key K. This assumption is rooted in the hardness of the DLP, which states that given g, p, and A, it is computationally difficult to find the secret integer a such that $A = g^a \bmod p$ [57]. The DH algorithm has been extended and adapted to various mathematical structures to enhance security and efficiency. One notable variant is the Elliptic Curve Diffie-Hellman (ECDH) algorithm, which utilizes the algebraic structure of elliptic curves over finite fields. The use of elliptic curves in cryptography was independently proposed by Neal Koblitz [58] and Victor S. Miller [59] in the mid-1980s.

The security of ECDH relies on the assumed hardness of the elliptic curve discrete logarithm problem (ECDLP) [60]. The DH algorithm and its variants have been widely adopted in various cryptographic protocols and systems to ensure secure communication over insecure networks. The Transport Layer Security (TLS) protocol [61], which is used to secure web browsing, email, and other internet applications, employs DH key exchange to establish secure communication channels. Similarly, the Internet Key Exchange (IKE) protocol [62], used in virtual private networks (VPNs), used for secure remote access, rely on DH key exchange for secure communication. The security of the DH algorithm has been extensively studied, and various attacks and vulnerabilities have been discovered over the years. One notable vulnerability is the man-in-the-middle attack, where an attacker intercepts the communication between Alice and Bob and establishes separate shared keys with each party, allowing the attacker to decrypt and modify the exchanged messages [52]. To mitigate this vulnerability, authentication mechanisms, such as digital signatures or certificates, are used in conjunction with DH key exchange to verify the identities of the communicating parties [56].

The DH algorithm has also been adapted to support key exchange in group settings,

where more than two parties need to establish a shared secret key. The Group Diffie-Hellman (GDH) protocol [63] extends the basic DH algorithm to allow multiple parties to contribute to the generation of the shared key.Various flavors of GDH have been proposed, including the Burmester-Desmedt protocol [64], each offering different trade-offs in terms of computational efficiency and communication overhead. The advent of quantum computing has posed a significant challenge to the security of the DH algorithm and its variants. Shor's algorithm [65], a quantum algorithm for solving the DLP and the integer factorization problem, has the potential to break the security of DH key exchange in polynomial time. To address this threat, researchers have been exploring post-quantum cryptographic algorithms that are resistant to quantum attacks. Lattice-based cryptography [66], supersingular isogeny-based cryptography [67], and code-based cryptography [68] are among the leading candidates for post-quantum key exchange. Despite the challenges posed by quantum computing, the DH algorithm remains a fundamental building block of modern cryptography. Its simplicity, effectiveness, and versatility have made it a cornerstone of secure communication protocols. As the field of cryptography continues to evolve, researchers and practitioners are working to develop new key exchange algorithms that can withstand the threats of quantum computing while maintaining the efficiency and security properties of the DH algorithm.

2.3.2 Shamir's Secret Sharing

Shamir's Secret Sharing [53] is a cryptographic algorithm developed by Adi Shamir, designed to divide a secret into multiple parts. Each part is known as shares. The essence of this method is that the secret can be reconstructed only when a predefined number of shares, or the threshold, are combined. The process begins by encoding the secret as a number. A polynomial of degree $k - 1$ is then generated randomly, where k is the threshold, or the minimum number of shares required to reconstruct the secret. The secret itself is embedded as the constant term of the polynomial, and the other coefficients are chosen at random. To create the shares, the polynomial is evaluated at several points, and the resulting values are distributed to the participants.

A fundamental aspect of Shamir's scheme is that any set of k or more shares can be used to interpolate the polynomial and recover the secret, typically through techniques like Lagrange interpolation. However, with fewer than k shares, the secret remains secure, as these shares do not provide enough information to solve for the polynomial. Shamir's Secret Sharing finds its utility in situations where both security and redundancy are paramount. It's particularly effective for protecting cryptographic keys, allowing them to be split and stored among different parties. In such a setup, no single individual has complete access to the key. However, the key can be reassembled when necessary by gathering the requisite number of shares.

2.3.3 Variations and Optimizations of Google's Secure Aggregation Protocol

Some variations and optimizations of the protocol include:

- Users only need to share keys with a subset of other users, as long as the subsets do not form disjoint clusters.
- The communication cost can be reduced to a constant factor over just sending the inputs, rather than quadratic in the number of users, by using more efficient secret sharing schemes.
- The computational cost for the server can be reduced from cubic to quadratic, by caching the intermediate values in Lagrange interpolation during the unmasking phase.
- By adding differential privacy noise to the partial sums over large enough subgroups before unmasking, formal differential privacy guarantees can be provided.

2.3.4 Key Concepts and Techniques in Google's Secure Aggregation Protocol

The core idea behind the Secure Aggregation protocol is the use of pairwise masking vectors that sum to zero. In this scheme, each pair of clients generates a masking vector

in such a way that the sum of the two vectors cancels out. Consequently, each client possesses one mask for every other client, and every pair of these masking vectors is generated to sum to zero. When clients send their inputs to the server, they add all of the masking vectors they have generated. Since the masking vectors are sampled randomly, the masked inputs also appear random. However, when the masked inputs from all clients are added together, the pairwise masks cancel out, resulting in the sum of the inputs without any masks. This allows the server to aggregate the updates to machine learning models without learning the individual models themselves.

2.3.4.1 Efficient Agreement on Masking Vectors

One challenge in implementing the Secure Aggregation protocol is the efficient agreement on the masking vectors, as these vectors can be quite long. To address this issue, the protocol employs the Diffie-Hellman key agreement scheme. In this scheme, there is a public generator g and a prime modulus p. Each client generates a secret key for each run of the protocol, and the corresponding public key is computed as $g^{secret\ key} \bmod p$. The server then broadcasts all the public keys to the clients.

Upon receiving the public keys, each client raises all the received public keys to the power of its own secret key. This operation gives each client a shared secret with every other client. These shared secrets are scalar values that can be used as seeds for a pseudo-random number generator (PRNG). By running the PRNG with these seeds, clients can generate masking vectors of the required length without the need to transmit the entire vectors during the key agreement step. This approach offers several advantages:

1. It eliminates the need to transmit large masking vectors during the key agreement step, making the protocol more efficient.
2. Broadcasting the public keys via the server is suitable for mobile phones, which may not support direct peer-to-peer communication.
3. Having fewer secrets stored on the phones makes it easier to recover in case of dropouts.

2.3.4.2 Handling Client Dropouts

Another challenge in the Secure Aggregation protocol is dealing with client dropouts. If some clients fail to send their inputs, the pairwise masks they generated with other clients will not cancel out, and the server will not be able to recover the sum of the inputs correctly. To address this issue, the protocol employs a well-known technique called threshold secret sharing, specifically Shamir's secret sharing scheme. In this scheme, a secret is divided into n shares, and any party possessing k or more shares can reconstruct the secret perfectly. However, any party possessing fewer than k shares learns nothing about the secret. In Shamir's secret sharing, a polynomial of degree $k-1$ is constructed such that the y-intercept is equal to the secret. The shares are then generated as points along this polynomial. If a party possesses fewer than k points, they cannot interpolate the polynomial and, consequently, have no information about the secret (i.e., the y-intercept). However, if a party has k or more points, they can easily interpolate the polynomial and recover the secret.

In the Secure Aggregation protocol, each client generates a random polynomial and selects random points along that polynomial. These points are then shared with the other clients through the server, encrypted using the keys derived from the Diffie-Hellman key agreement step. If clients drop out after everyone else has inputted their data, the server requests the remaining online clients to provide their shares of the dropped-out clients' secret keys. These secret keys are only used for a single run of the protocol. The online clients respond with the points on the polynomial (i.e., the secret shares), and the server interpolates the polynomial to recover the secret keys of the dropped-out clients. With these secret keys, the server can regenerate the masking vectors that the dropped-out clients would have used if they had been online. By generating all the masking vectors, including those of the dropped-out clients, the server ensures that all the masks cancel out, allowing it to recover the sum of the inputs from the online users. The goal is for the server to learn the sum of the inputs as long as a certain threshold of users remain online. This approach provides a level of security, as even a coalition of the server and some dishonest users cannot reconstruct the secret if the threshold is set high enough and there are sufficient online users.

2.3.5 Addressing Privacy Risks

While the protocol described above is effective in handling client dropouts, it introduces a privacy risk in real-world scenarios involving mobile phones. Due to network delays, a user's input may arrive late, making them appear dropped out even though they are still online. In such cases, if the server is honest but curious, it can recover the "dropped out" client's private key, regenerate their mask, and learn their individual input. To mitigate this privacy risk, the protocol introduces individual masks for each client in addition to the pairwise masks. These individual masks do not cancel out with any other masks, but the server should not be able to learn both the individual mask and the pairwise mask for any client. During the secret sharing step, clients create shares of two secrets: the Diffie-Hellman key and the individual mask key. Each client then has two shares for every other client. After users have sent their inputs and some users have dropped out, the server requests one share for each client. For clients that are still online, the server requests a share of their individual key. For clients that have dropped offline, the server requests a share of their Diffie-Hellman key. The clients respond with the appropriate shares, sending only one of the two shares for each client as requested by the server. The server can then reconstruct the keys it needs and start canceling out the masks. For every dropped-out client, the server reconstructs their Diffie-Hellman key and computes the pairwise masks. For the clients that did not drop out, the server reconstructs their individual key and eliminates the individual masks. By eliminating both the non-canceling individual masks and the pairwise masks, the server obtains the sum of the inputs from the online users. This approach protects the privacy of the online users, as their pairwise masks are not revealed to the server. If an offline user's input arrives late, their privacy is still protected. The server has already reconstructed their Diffie-Hellman key but not their individual mask key, and since it has already requested one of the two shares, it cannot ask for the other.

2.3.6 Security Against Different Adversary Models

The protocol described above is secure against honest but curious adversaries. However, with a few extra steps and additional communication, it can be made secure

against malicious adversaries and adapted to different settings of adversarial behavior.

Honest but Curious Server and Malicious Clients: In the case where the server is honest but curious, and the clients may be malicious, the threshold can be set to any desired value. This threshold determines the maximum number of users whose updates the server can learn without compromising privacy.

Malicious Server and Honest but Curious Clients: When the server could be malicious and the clients are honest but curious, an additional assumption is required. Either there is a private key infrastructure in place, allowing clients to recognize each other's identities and ensure they are not subject to a Sybil attack, or the Diffie-Hellman key agreement step is assumed to occur honestly. With such an assumption, the threshold can be set high enough to prevent a server-only adversary from attacking users' privacy. The reason for setting a higher threshold is to prevent a specific attack the server could perform if the threshold is too low. In this attack, the server could split the users into two cohorts, A and B, and inform all users in cohort A that cohort B dropped out and vice versa. If the threshold is less than half the number of clients, the server can reconstruct both keys for each client, compromising their privacy. However, if the threshold is set higher than half the number of users, the server learns nothing, and privacy is maintained. In this case, any sum computed by the server is guaranteed to include at least the threshold number of inputs, and the protocol remains resilient to up to half of the users dropping out.

Coalition of Malicious Server and Clients: In the case of a coalition between a malicious server and up to a third of the clients, the threshold must be set even higher. The server could create three cohorts: A, B, and malicious. It can have "A plus malicious" send one set of keys and "B plus malicious" send the other set of keys, giving the server two-thirds of the keys for each client. As long as the threshold is greater than two-thirds, the server learns nothing, and privacy is preserved. In this scenario, the server will learn the sum of the threshold number of inputs, except that the corrupt clients could always set their inputs to zero. Consequently, the server will learn the sum of (threshold - number of corrupted clients) inputs together. The protocol remains robust against up to a third of the clients dropping out.

Chapter 3

Literature Review

3.1 Internet Centralization

The phenomenon of centralization on the Internet and the concentration of infrastructure in the hands of a few dominant service providers have been the subject of investigation in previous studies. Authors in [18] discuss the potential issues and risks associated with the centralization of Internet infrastructure services, particularly focusing on the example of centralized DNS resolution. The paper highlights three main problems: single point of failure, surveillance, and concentration of information. Centralized systems can become vulnerable to accidental failures or targeted attacks, affecting the entire system. Additionally, centralized DNS resolvers can learn about users' Internet usage patterns, leading to privacy concerns and potential misuse of data. Moreover, centralization may provide certain service providers with an unfair advantage in collecting data and offering services built on top of the Internet infrastructure. The author emphasizes that changes in Internet infrastructure have far-reaching consequences across all users and traffic types, as compared to the impact of individual applications. Arkko's main contribution is to provide recommendations for addressing these issues, suggesting that, whenever possible, centralized designs should be avoided in Internet infrastructure services. Instead, the focus should be on deploying important features

like protected signaling or encryption while ensuring that no single points of failure or centralized information storage are created in the process. The paper draws from the Internet community's historical success in developing distributed solutions for critical services, such as DNS root services, certificate authorities, and mail services, and emphasizes the importance of adhering to the Internet's original design principles of decentralized control and heterogeneity. This work is significant in the context of understanding the challenges and risks associated with the growing trend of centralization in Internet infrastructure. It provides a comprehensive overview of the potential problems and offers high-level recommendations for maintaining a decentralized and resilient Internet architecture. The authors in [69] investigate the reliance of universities on public cloud infrastructure in seven countries (U.S., U.K., Germany, Switzerland, Austria, the Netherlands, and France) and in the Times Higher Education Top100 between January 2015 and October 2022. The main problems highlighted are the potential issues and risks associated with the centralization of Internet infrastructure services, particularly focusing on the example of centralized DNS resolution, email, learning management systems (LMS), and video conferencing tools. The authors discuss how this centralization can lead to single points of failure, surveillance concerns, concentration of information that may affect other services, and the far-reaching consequences of changes in Internet infrastructure across all users and traffic types. The paper's main contribution is to provide a longitudinal study of the migration to public clouds among universities in the selected countries and institutions. The authors find that cloud adoption differs between countries, with one cluster (Germany, France, Austria, Switzerland) showing a limited move to clouds, while the other (U.S., U.K., the Netherlands, THE Top100) frequently outsources universities' core functions and services, starting long before the COVID-19 pandemic. They attribute these differences to several socio-economic factors in the respective countries, including the general culture of higher education and the administrative paradigm taken towards running universities. The authors analyze and interpret their results, finding that the implications reach beyond individuals' privacy towards questions of academic independence and integrity. They recommend that whenever it comes to Internet infrastructure services, centralized designs should be avoided where possible, and institutions should focus on deploying important features like protected signaling or encryption while ensuring that no single points of failure or centralized information storage are created in the process. Additionally, authors in [70] investigate the critical issue of centralization and consolidation within the web hosting

industry, utilizing active DNS measurements spanning 19 TLDs from 2017 to 2021. The authors shed light on the alarming concentration of a substantial portion of the domain namespace in the hands of a select few large hosting providers, predominantly based in the United States. This centralization raises significant concerns regarding potential single points of failure and the accumulation of power within a limited number of market players, which could have far-reaching implications for the stability, security, and fairness of the internet ecosystem. Zembruzki et al. make several notable contributions to the understanding of the hosting industry landscape. Firstly, they reveal the extensive centralization present in the hosting market, demonstrating that a mere five US-based hosting providers are responsible for hosting a staggering one-third of the domains across the analyzed TLDs. This finding underscores the vulnerability of the internet infrastructure to the decisions and actions of a handful of dominant players, potentially compromising the resilience and diversity of the online ecosystem. Secondly, the authors illuminate the influence of geographical proximity and shared language ties on the hosting industry. While European country-code TLDs (ccTLDs) exhibit a robust local hosting industry, the study uncovers an intriguing trend among German-speaking countries, such as Austria, Switzerland, and Liechtenstein, which tend to utilize each other's hosting providers. This observation highlights the role of cultural and linguistic factors in shaping the hosting market, suggesting that businesses and individuals may gravitate towards providers that cater to their specific language and regional preferences. Additionally, the authors note that Canada's .ca TLD heavily relies on US-based hosting companies, indicating the significant influence of the United States' hosting industry on its northern neighbor. Thirdly, Zembruzki et al. reveal a concerning trend regarding the increasing market share of US-based hosting companies among popular domains across most TLDs. This finding poses a significant challenge to the European Union's digital sovereignty goals, as it suggests a growing dependence on foreign hosting providers for critical online infrastructure. The authors highlight Russia's ccTLDs as a notable exception to this trend, with the majority of popular domains hosted by local companies. This observation raises important questions about the role of government policies and regulations in shaping the hosting industry and protecting national interests in the digital realm. The study employs a comprehensive methodology to analyze the centralization and consolidation of the hosting industry, examining the concentration of domains per hosting provider across different TLDs, the geographical location of hosting companies, and the popularity of domain names as a proxy for

different market segments. In [71], the authors analyzed direct dependencies by focusing on the Mirai Dyn attack and its impact on the top 100k Alexa websites [72] that relied on the Dyn DNS server as their DNS provider. Their findings revealed that about 90% of the top 100K websites listed by Alexa depend on third parties. Among these websites, 50% to 70% are prone to service outages if prominent DNS providers (Cloudflare, AWS DNS, and DNSMadeEasy) fail.

3.1.1 DDoS Attack

The centralization of the Internet infrastructure has made it more vulnerable to Distributed Denial of Service (DDoS) attacks. DDoS attacks aim to overwhelm targeted servers or networks with a flood of traffic and make them inaccessible to users. The concentration of critical Internet services and infrastructure among a few dominant companies has created attractive targets for attackers, as a successful DDoS attack on these centralized providers can have escalating consequences, affecting numerous websites and services that rely on their infrastructure [73]. Numerous studies conducted to identify and analyze DDoS attacks on the Internet. A notable contribution. Moore et al. in [74] developed a technique to detect Randomly Spoofed Denial of Service attacks (RSDoS). Their research involves monitoring expansive address spaces to detect denial of service activities. These activities are identified through the analysis of backscatter traffic found in the Internet's background radiation (IBR). They also established specific threshold values, which enhance the accuracy in distinguishing genuine attack signals from regular data noise. Expanding upon this foundation, Jonker et al. conducted a comprehensive study providing a macro-level view of DDoS attacks across the Internet [75]. Their research particularly delved into the dynamics that drive the adoption of DDoS Protection Services (DPSs).

3.1.2 Security Dependencies

Authors in [76] illustrated explicit and implicit trust plus formed leverage for dependencies in the web ecosystem by observing Alexa's top-200k websites [77]. Due to their findings, around half of the websites establish chain dependency, and the most commonly implicitly trusted websites are the famous ones, such as Google. Then, they came up with a classification of third-party domains in terms of the possibility of being malicious in the invasion of privacy and mismanaging of users' data. Although they detected a small portion of service providers as suspicious, approximately a quarter of the websites have at least suspicious third parties in their dependency chain. Finally, they proposed an overall understanding of trust dependencies without being specific to the attack, distinguishing between the services and considering the vulnerable groups. Researchers in [78] looked into website dependencies more precisely. Same as authors in [10] presented a chain of dependencies and called it TPTs (Third Party Threes) in order to reveal the hierarchical dependencies between each website and its third-parties services providers. The remainder of their study seemed irrelevant to our literature reviews as they measured the dependencies from the inter-web site view. However, authors in [71] analyzed the dependencies focusing on the Internet infrastructural aspect. They measured the Mirai Dyn attack, the Dyn DNS server, by watching out the Alexa's top-100K websites' [77], before the Alexa list gets closed, vulnerabilities as the consequence of security dependencies from both direct and indirect dependability aspects. Finally, they proposed an analysis on whether the websites reduced their dependencies after a security event or decided to make redundancy by being dependent on multiple providers for the same services to avoid failure in case one of the third-party service providers experiences a security incident leading to unavailability of the website's services. Their findings empowered the results from [76] and revealed that less than 90% of Alexa's top 100 K websites depend on third parties, such as DNS hosters. On the other hand, they indicated to what extent the dependability on the centralized services would lead to security issues. Due to their results, 50 to 70 percent of the most popular websites would be unavailable if the top three DNS hosters, namely Cloudflare, AWS DNS, and DNSMadeEasy, fail. Although they analyzed Internet services from the infrastructural perspective, the same as previous authors, they did not consider vulnerable groups in their studies and did not have longitudinal analysis. Additionally,

since not all the websites rely on the service providers, it can be implied from the number of chosen websites in [76] and [71], which were top-200K and top-100K, respectively, and the fact that Alexa ranking [17] demonstrates the websites' popularity, that as the popularity rate of the websites improves, their tendency to be dependant on the cloud-services is higher.

3.1.3 DNS Dependencies

The prevalence and impact of third-party dependencies have been analyzed by Kashaf et al. [79] and Urban et al. [78], focusing on vulnerabilities and the concentration of dependencies on third-party service providers. The vulnerability of government domains has been investigated in [80]. The authors studied the availability of DNS records for government domains across more than 190 countries, including an investigation of the increasing reliance on a single third-party DNS service provider and of vulnerabilities to hijacking due to defective delegations. The authors also found that government domains are vulnerable to DNS misconfigurations, which can lead to service degradation or even service interruption.

3.1.4 Vulnerable Population Due to Dependency

Many researchers measured the Internet to reveal the impact of any incidents on users in a certain period of time. For example, Covid-19, as the most recent global event, has directly affected Internet services and users' behaviors. According to their findings, there is an increase in the rate of user demand for the Internet from different perspectives in different parts of the world, for example, the growth in the amount of time invested by people in online activities [81] , the higher demand for the use of social media like Facebook [82] , moving on to the leading universities to the online solution [83] and as a whole the higher Internet traffic, particularly remote-working-related applications like video conferencing. Although all the previous research took user groups into account, and even their findings can be extended from the security

point of view, none looked at the Internet services at the infrastructural level.

3.1.5 Digital Divide

The digital divide has been investigated in numerous works, including [84, 85, 86].In [84], Wang et al. investigated the digital divide through the lens of energy poverty and found that it negatively impacts the usefulness of the Internet. The extreme remoteness and isolation of indigenous communities in Australia contribute to the existing digital divide that reduces the quality of Internet connectivity and limits access to Internet services [85, 86].

3.2 Secure Aggregation Protocol

Several works have addressed the challenges of secure aggregation in federated learning, with approaches focusing on enhancing security, reducing computational overhead, and adapting to resource-constrained environments.

In [87], the authors critically examine the Verifiable and Oblivious Secure Aggregation (VOSA) protocol, initially proposed by Wang et al. for privacy-preserving federated learning. The paper identifies critical vulnerabilities in VOSA, demonstrating its susceptibility to forgery attacks by a malicious aggregation server (AS), thus compromising the protocol's claimed aggregate unforgeability and verifiability. By highlighting these security flaws and showcasing specific attack scenarios, the paper advances the understanding of secure aggregation's limitations and points out areas for improvement. Similarly, EdgeSA, introduced by [88], tackles the challenges of privacy-preserving federated learning in edge computing environments, where resource constraints are a significant concern. EdgeSA employs pairing-based cryptography for homomorphic encryption and bilinear signature schemes, alongside a Decentralized Key Generation (DKG) approach to eliminate the need for a trusted party in the initialization phase. The work's key contributions include the design of a secure

aggregation technique that suits the resource-constrained nature of edge devices, enabling secure and efficient federated learning in such environments. Additionally, [89] presents Falkor, a protocol that addresses secure aggregation in federated learning, particularly within a multi-server scenario. The protocol uses AES in counter mode for stream cipher-based masking of local models and leverages GPU acceleration to enhance computational efficiency. Falkor's focus on scalability and performance, especially when dealing with a large number of clients, distinguishes it from earlier works.

Addressing the problem of high computation and communication overhead in urban sensing systems, [90] propose the Resource Adaptive (ReAd) Turbo-Aggregate protocol. This protocol adapts space and time complexity to match the participating users' network, processing, and battery resources. The proposed solution is verified on a synthetic dataset for a noise mapping application, demonstrating significant reductions in time complexity as the grouping parameter increases, with acceptable information loss. For the domain of smart farming, [91] introduce a blockchain-based secure aggregation scheme aimed at improving agricultural practices in India's extensive agricultural sector. By incorporating IoT and blockchain technology, the proposed system enables smart tracking and aggregation of agricultural data, contributing to better decision-making and potentially increasing crop yields. Pejo et al. [92] focus on the issue of quality inference in federated learning with secure aggregation. They address the challenge of estimating the quality of individual datasets and propose intuitive scoring rules to assess and reward or punish participants based on their contributions. The approach demonstrates how quality inference can stabilize training performance, measure individual contributions, and detect misbehaving participants. Flamingo, introduced by [93], aims to optimize the process of secure aggregation, particularly in the face of client dropouts. The authors highlight the limitations of existing approaches, such as pairwise masking and multi-party computation (MPC), and propose Flamingo to reduce the overall round-trip complexity for multiple sums, improving both client and server efficiency. Flamingo's performance is evaluated through comprehensive experiments, showcasing its effectiveness in realistic federated learning environments. In [94], the authors address the challenge of high communication-computation overheads in secure aggregation for federated learning networks. By combining techniques like "Sparse," "HeteroSAg," and "Autotune," they propose a new protocol that achieves

lower communication and computation overheads, even when scaled to a larger number of clients with longer update vectors. The authors demonstrate the effectiveness of their protocol using a sparse assignment graph to optimize device collaboration and quantization steps. Moreover, [95] introduce GroupSecAgg, a novel protocol designed for secure aggregation in federated learning with uncoded groupwise keys. This approach minimizes communication costs from the users to the server, particularly in the presence of user dropouts. GroupSecAgg demonstrates the same optimal communication cost as the best-coded key protocols when the group size exceeds a certain threshold, showcasing considerable improvements in key sharing and model aggregation times. Finally, in [96], the authors provide a comprehensive analysis of secure aggregation (SA) protocols based on cryptographic schemes for federated learning. The paper highlights the lack of systematic study and clear comparison of SA solutions, addressing this gap by providing a formal definition of secure aggregation and categorizing existing solutions based on underlying cryptographic techniques, including masking, additive homomorphic encryption, and multi-party computation. It also addresses challenges such as client failures, large input dimensions, and inference attacks, providing a valuable overview for researchers seeking to enhance SA in federated learning.

3.2.1 Active Adversary

The presence of active adversaries in federated learning systems poses significant challenges to the security of aggregation protocols. Several approaches have been developed to mitigate these risks while maintaining efficiency in terms of communication and computation costs.

In [97], the authors propose SAFELearn, an efficient design for privacy-preserving federated learning (FL) systems. SAFELearn is designed to protect against inference attacks by using secure aggregation without relying on cryptographic primitives that may introduce high computational overhead. A notable feature of SAFELearn is its adaptability to client dropout scenarios without the need for a trusted third party. It employs federated-averaging aggregation, ensuring that each client receives an encrypted

global model, which is then decrypted for local training. The clients subsequently send encrypted updates back to the aggregator. Depending on the application, SAFELearn implements various secure methods such as Fully Homomorphic Encryption (FHE), Secure Multi-Party Computation (MPC), or Secure Two-Party Computation (STPC) to prevent the aggregator from accessing clients' model updates. SAFELearn demonstrates impressive efficiency, aggregating 500 models with over 300K parameters in under 0.5 seconds, showcasing its practicality for real-time applications. Building on this theme, ELSA [98] tackles the issue of secure aggregation under an active adversary model. The authors highlight the inefficiency of existing protocols and the lack of security measures for malicious actors. To address these limitations, ELSA introduces a protocol based on distributed trust across two servers, ensuring that individual client updates remain private, provided at least one server remains honest. This method significantly improves performance compared to previous protocols, outperforming the single-aggregator RoFL by up to 305 times and the distributed trust Prio by up to 8 times. Additionally, ELSA is particularly efficient in handling bandwidth-constrained clients and can tolerate client dropouts without runtime degradation. A key innovation in ELSA is the delegation of cryptographic correlation generation to the clients, which speeds up the protocol and strengthens security without a substantial increase in communication overhead. Unlike SAFELearn, which focuses on inference attacks, ELSA specifically targets malicious adversaries, further extending the scope of active adversary defenses.

Further enhancing the security of aggregation against active adversaries, [99] propose Efficient Security Enhancement (ESE). This protocol is an improvement on the secure aggregation protocol developed by Bonawitz et al. [50], which is primarily designed for semi-honest adversaries but is vulnerable to active adversary attacks, such as the Eclipse attack. ESE introduces a four-round protocol that adjusts the authentication timing and embeds timestamps into messages to counteract replay and Eclipse attacks. The protocol exploits mobile device synchronization capabilities to enhance the security of aggregation in federated learning systems. ESE addresses the vulnerabilities in Bonawitz et al.'s protocol by providing a robust defense against active adversaries while minimizing communication and computation overhead. In [100], the authors propose a scalable privacy-preserving aggregation scheme that addresses the challenge of participant dropout in federated learning, a common issue that can be exploited by

active adversaries. This protocol modifies Shamir's secret sharing scheme to ensure that even if participants drop out after uploading their masked models, their data can still be unmasked and contribute to the aggregated model. The scheme is more efficient and simpler than existing solutions, such as Google's secure aggregation protocol [50], offering a practical alternative for dropout-resilient aggregation in privacy-preserving machine learning. Compared to the SAFELearn and ELSA protocols, which focus on communication efficiency and defending against malicious adversaries, this scheme emphasizes the importance of dropout resilience while maintaining security against both semi-honest and active adversaries. In conclusion, addressing the challenges posed by active adversaries in federated learning remains a critical concern. Protocols like SAFELearn [97] offer efficient solutions for semi-honest adversaries, while ELSA [98], ESE [99], and the protocol proposed by [100] extend protections against more advanced threats such as active adversaries, client dropouts, and system-level vulnerabilities. Each of these protocols offers unique insights into achieving secure aggregation in federated learning under various adversarial models.

3.2.2 Edge Computing

Several approaches have been proposed to enhance privacy and efficiency in secure data aggregation within IoT networks, edge computing, and federated learning environments. These methods aim to address the dual concerns of data privacy and energy efficiency, while also improving overall system performance. In [101], the authors propose a Blockchain-based Secure Data Aggregation (BSDA) strategy, which integrates blockchain with edge computing to tackle privacy and energy efficiency challenges in IoT networks. BSDA employs a security label attached to the block header, which includes the task security level (SL) and completion requirement (CR). These labels ensure that only authorized mobile data collectors can access sensitive tasks, preventing unauthorized access. The strategy also includes a deep reinforcement learning-based method, Improved Self-adaptive Double Bootstrapped Deep Deterministic Policy Gradient (IDDPG), to optimize data collection routes, further improving energy efficiency. Through simulations, BSDA is shown to outperform existing methods in

terms of throughput, transaction latency, aggregation ratio, and energy cost, demonstrating its effectiveness in enhancing system performance.

In a similar context, [102] present a privacy-preserving data aggregation scheme specifically designed for Mobile Edge Computing (MEC) in IoT applications. Their approach focuses on reducing communication costs while ensuring the privacy, authenticity, and integrity of data. This scheme involves three key participants: terminal devices (TD), edge servers (ES), and a public cloud center (PCC). TDs encrypt the data before sending it to the ES for aggregation, and the PCC can decrypt the aggregated data using its private key. This approach preserves privacy and guarantees the integrity of the data while reducing communication overhead. Similarly, [103] propose a novel privacy-preserving data aggregation scheme using Simulated Annealing Module Partition (SAMP) and Differential Aggregation Encryption (DAE) models. Their approach introduces noise at the end sensor level, which is then encrypted using the Diffie-Hellman algorithm. The edge node decrypts and aggregates the data, ensuring protection from unauthorized access while minimizing computation and communication overhead. This method offers a balance between security and efficiency in edge computing environments. In FL, secure aggregation plays a critical role in enhancing privacy and efficiency across distributed learning systems. The reviewed contributions, including BSDA, MEC-based privacy-preserving schemes, and novel encryption approaches, provide innovative solutions for improving secure data aggregation in diverse applications such as urban sensing, smart agriculture, and edge computing.

3.2.3 Vulnerabilities and Security Enhancements

The vulnerabilities in secure aggregation protocols for federated learning (FL) have been well documented, driving the need for continuous improvements. Wu et al. [87] introduce the Verifiable and Oblivious Secure Aggregation (VOSA) protocol, specifically its susceptibility to cyber-attacks. Addressing similar concerns, authors in [96] provide an evaluation of secure aggregation protocols for FL to identify gaps in the

existing literature, which they use as a foundation for further research into security improvements. Several security enhancements have emerged in response to these identified vulnerabilities. Rathee et al. [98] propose ELSA, a protocol designed to mitigate threats from active adversaries. ELSA reduces communication and computation overhead while maintaining privacy. The paper [100] contributes to a scalable, dropout-resilient aggregation protocol, which preserves privacy and adapts efficiently to client dropout scenarios. Other protocols have also addressed the need for more efficient and adaptable systems. Falkor is proposed in [89] by Georgieva et al. that is based on GPU acceleration and stream cipher-based masking to make a scalable and efficient aggregation across multiple servers while maintaining privacy. Similarly, authors in [90] focus on urban sensing systems with the Resource Adaptive Turbo-Aggregate protocol, designed to adapt aggregation processes based on available network resources, to ensure efficiency without sacrificing security. In addition to these aggregation-focused improvements, addressing model poisoning has also gained attention. In [104], Wang et al. present the Client Selection Secure Collaborative Learning (CSSCL) algorithm, which uses similarity metrics to detect and mitigate model poisoning attacks.

Efforts to address both security and resource constraints in edge computing have also been explored. Bouamama et al. [88] propose EdgeSA, a privacy-preserving protocol for edge computing in FL, utilizing pairing-based cryptography and decentralized key generation to eliminate reliance on trusted parties. EdgeSA makes the secure aggregation process suitable for resource-constrained edge systems. In another paper [91], the integration of blockchain technology with IoT in smart farming has been investigated to develop a secure aggregation protocol that enhances data management and optimizes agricultural practices. Moreover, authors in [101] propose a Blockchain-based Secure Data Aggregation (BSDA) strategy that integrates a security label system to ensure task integrity and confidentiality in IoT networks.

Chapter 4

Investigating the Impact of Internet Centralization on DNS Services as a Digital Divide

The concept of Internet centralization [19] is closely linked to the digital divide, as both affect the distribution of access and control over digital resources. Internet centralization refers to the consolidation of Internet infrastructure and services within a few dominant providers, which can lead to disparities in service quality, reliability, and availability. These disparities often emerge across different regions and population groups inside a country. The digital divide [28], on the other hand, represents the gap between those with access to digital technologies and the internet and those without. This divide is deepened by Internet centralization, as it can limit access to essential services and information, particularly for marginalized communities such as Indigenous people of Australia [33]. This chapter investigates the relationship between the digital divide focusing on critical dependencies in Internet infrastructure, such as DNS provisioning [17]. DNS (Domain Name System) is a vital service in the Internet infrastructure. It has become common for network and website operators to outsource the operation of their DNS services to a (limited) number of specialized DNS providers. Depending on the choice of provider, a network or site may achieve better or worse

availability, especially under adversarial conditions (power outages, attacks, etc.). By analyzing DNS provisioning and dependencies for Australian government websites. this chapter aims to identify a possible digital divide. More specifically, we investigate setups with respect to potential drawbacks in terms of availability or domestic control over the setup. We choose sites whose audience is primarily the indigenous population and sites that target the broader, general population. We can indeed identify differences between the DNS dependencies, in particular with respect to the use of hyperscalers, domestic vs. international providers, and dedicated government infrastructure. The implications for availability and control are more subtle and require further investigation. However, our results show that Internet measurement can detect signals of possible digital divides.

4.1 Motivation

The concept of the digital divide has been the subject of much research and discussion over the past 20 years. The term was first introduced and defined in the mid-to-late 1990s in a series of reports titled “Falling through the net” [25, 26, 27]. The definition refers to the gap between individuals or groups who have access to and effectively use digital technologies and those who do not. This includes access to technologies such as the Internet [28]. Individuals who have access to and utilize these technologies are considered advantaged, while those who lack access or proficiency are at a disadvantage [29]. A digital divide often affects already economically disadvantaged groups. The indigenous people of Australia consist of two distinct cultural groups: the Aboriginal peoples of the Australian mainland and Tasmania and the Torres Strait Islander peoples from the seas between Queensland and Papua New Guinea. It is known that indigenous communities face challenges in accessing digital information and acquiring the necessary skills for effective utilization [33]. In recent years, understanding how the Internet works and identifying critical dependencies have come into efforts [17]. Differences in how Internet services are set up for different groups have implications for how these groups can access the Internet. In this sense, revealing Internet dependencies can reveal implications for the digital divide. This is also evident in core

Internet infrastructure, namely DNS. Here, much centralization and consolidation have occurred [18, 19]. This refers to the dominance of a limited number of large service providers who exert significant control over various aspects of the DNS. DNS employs a hierarchical configuration with multiple authoritative name servers to distribute the workload and enhance the name resolution process. However, today, much of this setup is in the hands of very few providers, and major companies such as Microsoft, Amazon, Cloudflare, and Google have significant influence over DNS provisioning. Where operators choose to outsource the operation of their DNS, the implication is that their users also rely on (a possibly limited) number of (possibly centralized) providers to access Internet services of relevance to them. It has been stated that this dependency on a few providers increases the vulnerability to potential attacks and raises concerns about the overall resilience of Internet services [20, 21].

The question we ask in this chapter is whether DNS dependencies impact how vulnerable groups can access Internet-based services. We focus on analyzing the impact of the digital divide on indigenous communities in Australia regarding their DNS-mediated access to government websites. Given their geographically dispersed nature, service outages can significantly impact this vulnerable group. We examine the disparities in DNS dependencies of governmental services for the indigenous and general populations. Our findings imply differences between the setups do exist: sites for the indigenous population use different cloud providers, and when they use smaller providers, these are often domestic rather than international. While sites for the general population are sometimes run on what seems to be government-owned infrastructure, we find no such setups for sites for the indigenous population.

In the following section, we explain how we created lists with the domain names of the relevant services provided by the Australian government for the indigenous populations as well as the general population, and how we retrieved their DNS records. Our objective is to create two lists: one with the domain names of Australian government websites that provide services to the general population and one with domain names of Australian government websites that provide services for the indigenous populations. To the best of our knowledge, there are no existing open-access data sets for this purpose. We adopt a desk research approach to identify the domains of interest. While we go beyond second-level domains and consider subdomains (which may have their

own authoritative name servers), we use the general term “domain” or “domain name” to refer to all of these jointly. We undertook the following steps in the first quarter of 2023. To achieve two distinct sets of domain names for the indigenous and the general population, we perform the steps below in two rounds. In the first round, we add the following indigenous-related terms: *indigenous*, *Aboriginal people and Torres Strait Islanders*, and *first nations* to keywords to collect domain names dedicated to services for the indigenous population. In the second round, we use keywords without these terms to capture domain names for the general public.

4.2 Identify and Categorize Australian Government Domains

Indigenous people in Australia face unique challenges and disparities in various aspects of life, including health, education, employment, and cultural preservation. To address these issues, it is crucial to identify and provide access to essential services and support tailored to the specific needs of indigenous communities. This section explores the key services and support mechanisms available to indigenous people in Australia. One approach to identifying relevant services is to search for dedicated domains or subdomains using search engines like Google, Bing, or Yahoo, employing specific keywords related to indigenous services. Social media platforms such as Facebook, Twitter, and LinkedIn can also be utilized to find dedicated groups or communities. To identify the specific domains, web crawling techniques can be employed to discover websites that deliver services for indigenous people. The selection of keywords for web searching should be based on the essential services that are assigned to the needs of indigenous communities.

4.2.1 Service Category Exploration

The initial phase involves desk research on online resources (such as government web pages, news websites, and blog posts) to identify the scope of services the Australian government offers to both general and indigenous populations. Our exploration begins with *Services Australia*[105] portal, a foundational government resource that includes information about a considerable range of services accessible to Australians. Despite the information provided by *Services Australia* and other government resources, to the best of our knowledge, there is no single, comprehensive, and categorized list of services that could be directly utilized for our research. Therefore, we conducted desk research to identify the **broad categories of government services** relevant to our study. This preliminary exploration lays the groundwork for the subsequent data collection phase. We identified 16 main categories of government services. These categories and their corresponding keywords are presented in Table 4.1. Each category is extracted by initial analysis of the web page's content to identify the most relevant and frequently occurring terms and phrases that best describe the services within each category. These categories include a wide range of essential services. For instance, Australia's **healthcare** system operates as a complex network involving public and private sector organizations that work together to provide health services to all Australians [106]. In parallel to healthcare, the Australian government offers **support services for individuals with disabilities**, ranging from specialized medical care to providing assistive technology and financial assistance to enhance the quality of life and ensure inclusion in society [107]. These selected categories serve as the primary keywords for our data collection process to identify the specific government domains that deliver essential services to the general and indigenous populations in Australia. By focusing on these services and finding the relevant government domains, particularly in relation to the indigenous population, we understand how DNS dependencies and geographic distribution of servers might affect access to essential government services.

These services may vary depending on the specific needs and circumstances of indigenous communities, and may also be influenced by local, state, or federal policies and programs. Table 4.1 shows the lists of services and keywords used.

Table 4.1: Keywords sorted based on the identified 16 categories of government services.

Healthcare	**Disability support**	**Family support**
Preventive care	Rehabilitation services	Child support
Chronic conditions	Assistive technology	
Specialist care	Improve accessibility	Childcare
Telehealth services	Promote social inclusion	Youth support
Vaccination	Community program	Adolescent support
Medical services		Violence prevention
		Residential care
Education	**Housing**	**Community development**
Training programs	Homelessness support	Individuals support
School programs	Affordable housing	Cultural maintenance
Vocational training	Appropriate housing	Social connection
Adult education	Home-ownership	
Disaster relief	**Economic development**	**Women support**
Emergency services	Employment services	Women health
Rebuilding homes	Job training	Accommodation service
Infrastructure improvement	Job seeking	Support groups
Temporary accommodation	Financial assistance	Employment opportunities
Distribution of food	Financial stability	Domestic violence
Retirement Support	**Cultural preservation**	**Mental health**
Age pension	Language program	Well-being
Superannuation savings	Traditional arts and crafts	Counselling services
Legal services	**Environmental programs**	**Business support**
Legal aid	Land management	Business training
Resolving disputes	Protect sacred sites	Business mentoring
Justice	Traditional lands	Entrepreneurship
Family law	Natural resources	Procurement policies
Criminal law	Preserve cultural heritage	Provide funding
		Business networking
Addiction support		
Substance abuse		
Treatment service		

Table 4.2: List of Australian government services providing DNS services.

Government of Australian Capital Territory (Department of Education and Training)
Australian Antarctic Division
APRA (Australian Prudential Regulation Authority)
Department of Defence
Department of Education, Skills, and Employment
Government of New South Wales (Department of Customer Service)
Government of New South Wales (Department of Social Services)
Government of Queensland (Department of Housing and Public Works)
Government of South Australia (Department of Premier and Cabinet)
Tasmania Department of Premier and Cabinet
Government of Victoria (Department of Premier and Cabinet)
Government of Western Australia (Department of Premier and Cabinet)
National Library of Australia
New South Wales Department of Education and Communities
Queensland Department of Education and Training
Services Australia

4.2.2 Domain Collection

In this section, we outline the domain collection process, specifically targeting Australian government websites that provide a range of services to the general and indigenous population. The types of services identified in section 4.2.1 serve as the primary keywords for this process. By utilizing these principal keywords, we identify the actual government services and their corresponding domains. There are no open-access data sets of Australian domains categorized based on the target population group that we could have used directly for our study. Hence, we employ a web crawling technique to identify them. We follow these steps to collect governmental domains for the general and indigenous people of Australia:

1. **Web search**: Utilizing Google search, we fetch pertinent governmental websites using keywords associated with the 16 service categories identified in the previous step. These keywords, derived from the service category exploration phase, serve as search queries to identify government websites offering services in these specific areas. Our search is restricted to websites with the *.gov.au* suffix, guaranteeing the inclusion of only authorized Australian government websites.

2. **URL extraction**: We identify the first 100 URLs obtained from Google search results and add them to the URL dataset.
3. **Keyword extraction**: This step involves crawling the URLs of the visited web pages and using a word cloud technique to extract the top five most common and contextually relevant words from each web page. A manual check is then conducted to verify relevancy. These extracted keywords are cross-checked against the existing ones in the set, and those that are new and relevant are added to the set for further searches.
4. **Domain extraction**: we parse the web page URLs and extract the corresponding domains, which are then collected and stored in a dedicated data set.
5. **Iteration**: We iterate until no additional keywords or domains are identified.

To create two unique sets of domains for the indigenous and general populations, we conduct the process from step 1 to step 5 in two rounds. In the first round, we include keywords related to indigenous terms, such as *Aboriginal and Torres Strait Islander*, to collect domains dedicated to indigenous services. In the second round, we exclude these terms to collect domains targeting the general public. After capturing these domains, we manually validate that they accurately match the respective target group. This process ensures the creation of two distinct and relevant domain sets. Data collected through this paper is accessible via IEEE DataPort [108]. Fig. 4.1 illustrates the flowchart that outlines the steps to identify and categorize Australian government domains, focusing on services for the general population and indigenous communities.

We proceed to retrieve the authoritative name servers (NS) for the collected domain names by querying every authoritative NS to whom we observed a delegation. We utilize standard tools for DNS look-ups provided by the Linux operating system, as speed is no concern. To maximize coverage, we follow the delegations from the root servers, which allows us to capture the authoritative NS records. We follow the delegations until we reach the final authoritative name servers (we performed retries for several domains in Tasmania, while no such errors or timeouts were encountered in other instances). This process took place until the end of March 2023. In addition, we also utilize the WHOIS command to gather information about the associated provider for each identified name server. We create the delegation graphs to analyze the dependencies. The relationship between domains and their name servers can be categorized as either direct or indirect dependencies. A direct dependency is a domain being directly

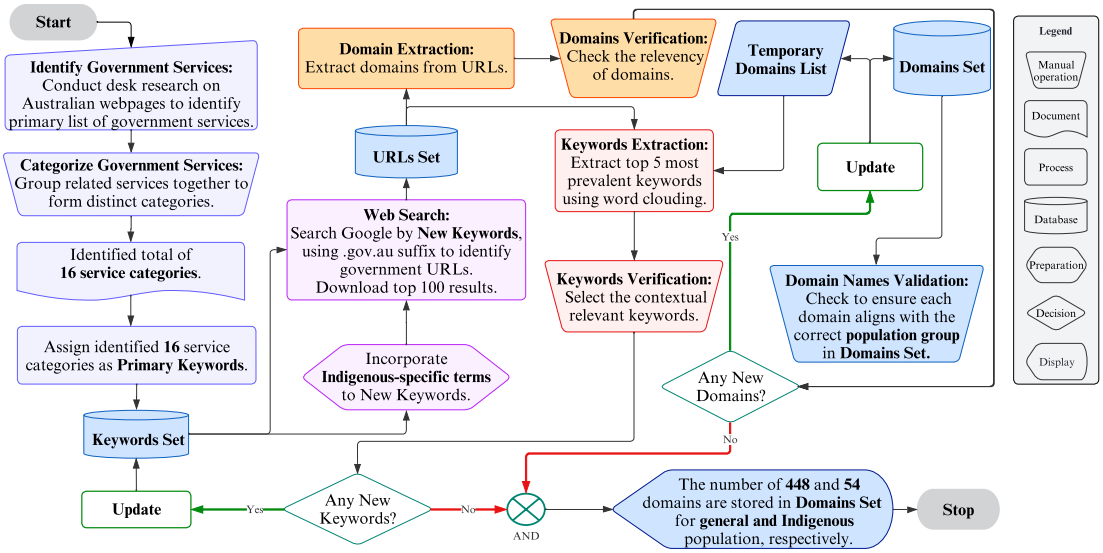

Figure 4.1: Flowchart for identifying and categorizing Australian government domains by general and indigenous population.

associated with its designated name servers. These associations indicate an immediate connection between a governmental website and its corresponding DNS service provider.

4.3 Analytical Results

We analyze the dependency patterns for domains for the general and indigenous populations across various DNS providers. Table 5.1 provides key statistics on the dependencies we find for various provider types. The table also presents the percentage of domains with a dependency on a single provider versus a dependency on multiple providers. We distinguish between the following kinds of DNS providers:

- **Leading providers:** We use the term "leading providers" to refer to prominent DNS service providers with a significant market presence and influence. These are widely known cloud providers often referred to as hyperscalers. They are often US-headquartered and relied on by a very large number of domains. Understanding dependencies on such

leading providers enables us to assess the concentration of control within the DNS infrastructure of the domain we investigate. On the one hand, if many domain names on our lists are served by the same leading provider, an outage or attack may take them all offline. Similarly, one vulnerability in a hyperscaler may impact a vast number of customers. On the other hand, such leading providers also have the resources to fend off attacks and generally have specialists to deal with security issues. Outages and vulnerabilities are hence (very) low frequency–very high impact scenarios. Hyperscalers are a common choice when services must be reachable quickly across a wide geographic area. However, the fact that they are generally headquartered in another country also implies a certain amount of loss in digital sovereignty when they are chosen over a local, domestic provider. The observed leading providers in our data set are: Amazon, Microsoft, Cloudflare, Akamai, EasyDNS, Google, Microsoft, Neustar Ultra DNS, and DNS Simple.

- **Non-leading providers** is our term for DNS providers outside the group of the leading (hyperscaler) providers. They generally have a smaller market share and fewer cloud resources and represent a wide and diverse range of DNS service providers. Many domestic (Australian) providers fall into this category. Non-leading providers are usually unable to offer the reliability and scalability of hyperscalers. Their availability and security stance vary widely, although it is plausible that at least their availability is lower than that of a hyperscaler, and they may be less capable of fending off a sophisticated, large, or sustained attack such as one may expect from state actors.
- **Intra-government providers** are those where the respective governmental sections are responsible for hosting and managing their DNS infrastructure, including offering DNS provisioning for other government sections. We filter the name servers with the *.gov.au suffix to find government-owned providers.
- **Undisclosed providers:** For about two percent of general domains, we could not further identify the DNS providers from either the WHOIS or the domain names of the NS records. We label them as “undisclosed”.

Table 4.3: Dependency on third-party DNS providers for general and indigenous domains.

Population group	General		Indigenous	
	Absolute	Relative	Absolute	Relative
Number of domains	448	100%	54	100%
Depends on...				
... leading providers	219	48.9%	29	53.7%
... non-leading providers	140	31.3%	25	46.3%
... intra-government providers	113	25.2%	0	
... single provider	412	92%	54	100%
... multiple providers	36	8%	0	0
... intra-government + 3rd party providers	19	4.2%	0	0
Undisclosed	8	1.8%	0	0

4.3.1 Analysis by Provider Type

Fig. 5.2 illustrates the relationships between domains and DNS providers that we group as “leading”, “non-leading”, and “intra-government” dependencies. While some general domains have implemented a multi-provider strategy, possibly to mitigate risks associated with a single, critical dependency, the practice is not widespread. It is particularly noteworthy that it is absent for domains for the indigenous population.

4.3.2 Single-provider setups

We first investigate how many domains rely on a single DNS provider, which is a critical metric: outage of this provider will make the relying services unavailable. We find that 92% of all domains for the general population rely on a single provider. *All* of the domains for the indigenous populations do so. This implies a generally unsatisfactory state across all government domains, but it is also a first hint that there is a difference between the services for the two population groups.

4.3.3 Multi-provider setups

Having multiple DNS providers offers benefits in terms of redundancy and resilience. In the event of a service outage or disruption from one provider, the availability of DNS services can be maintained through the alternative provider. Inequalities in the use of multi-provider strategies hence reflect differences in access to information and online services. Fig. 5.6 shows the distribution of domains with a multi-provider dependency for the general population (none of the indigenous websites have multiple DNS providers). We find that 20% of setups have a dependency on two distinct leading DNS providers (Amazon and Microsoft); this was observed for eight domains of the Victorian government. More than 50% of setups use a governmental provider along with a third-party DNS as an alternative server.

4.3.4 Use of leading providers

Hyperscalers may offer higher availability and potentially better security than smaller providers. Approximately half of the domains for both the general and indigenous populations rely on a single leading DNS provider. Only around 2% of the domains for the general population employed *two leading* providers, with the remainder using either a second non-leading or intra-government provider. Fig. 5.3 shows a breakdown of the leading DNS providers for our domains. For the general population, 48.9% of domains rely on leading providers, with Amazon being the most utilized provider at 21.4%. Microsoft is the second most commonly used provider at around 17%, followed by Cloudflare at 6%. Other leading providers, such as Akamai, UltraDNS, Google, DNSimple, and EasyDNS, are used in less than 5% of DNS services for the general population. Regarding domains for the indigenous population, 53.7% of them rely on leading providers. Microsoft is the most utilized provider at 31.5%, followed by Cloudflare (11.1%) and Amazon (7.4%). No other leading providers are in use for these domains. Comparing the two groups of domains, we identify a common preference for leading providers, although the preferred providers differ starkly. Cloudflare offers a free tier, which may explain this common choice in the second group of domains. There is slightly less variety in the chosen providers in the case of the domains

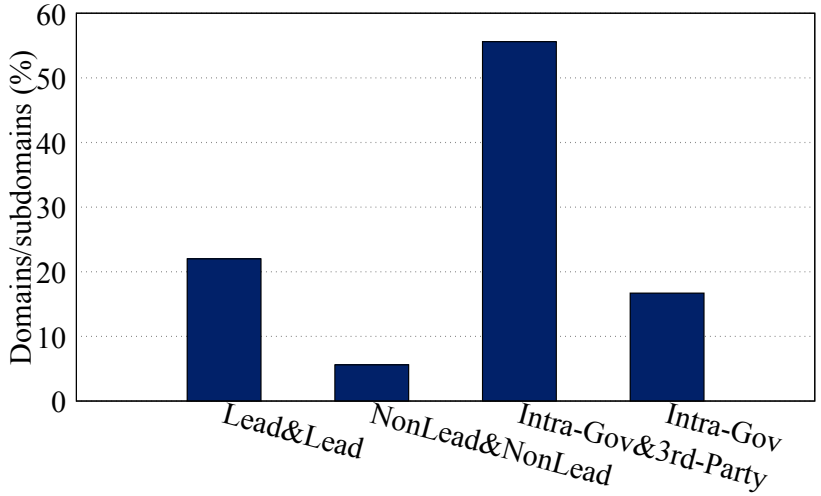

Figure 4.2: Multi-DNS-provider setups. Note that no domains for the indigenous population use such a setup.

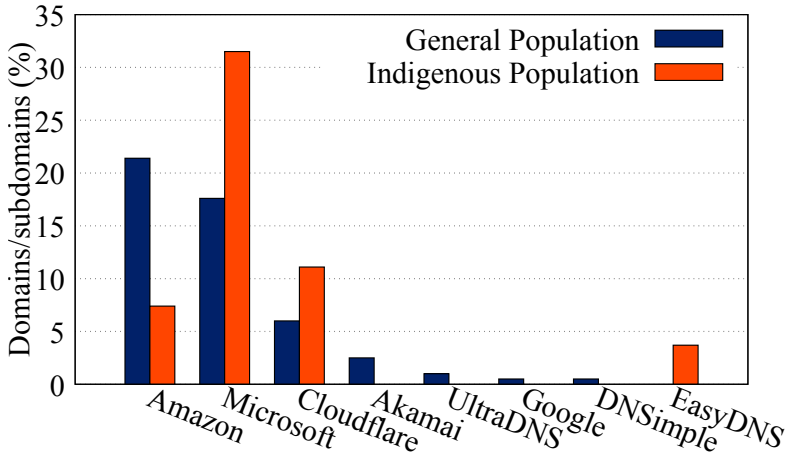

Figure 4.3: Leading DNS providers.

for the indigenous population.

4.3.5 Use of Non-leading Providers and Intra-government Providers:

As we see in Fig. 5.2, slightly more than half of domains for the general population rely on at least one non-leading provider or an intra-government provider, with an almost equal split between the latter two. We do not observe this for the domains of the indigenous population: here, 46.3% of the domains rely on non-leading providers,

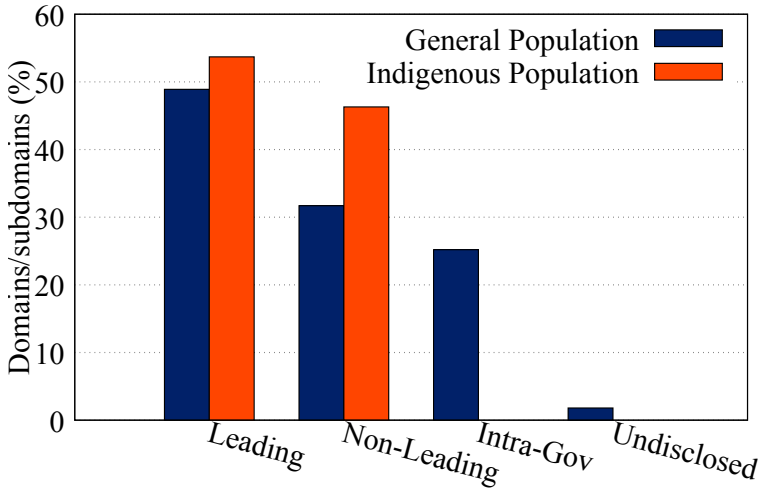

Figure 4.4: DNS providers by category.

and none use intra-government providers. Government-hosted providers would be required to comply with Australian standards and government regulations, and using these providers implies a certain level of coordination and collaboration. We list the government sections we observe in the domain name of the NS records in Table 4.2. While we observe only about 15 government agencies operating name servers, we see that they serve well over 100 different domains. It seems curious that no single service for the indigenous population is among these. Fig. 4.5 shows whether the non-leading providers are domestic or international. The fraction of domestic providers is significantly higher for both domain groups. Concerning domains for the general population, 23.4% of domains rely on local DNS providers, indicating a preference for domestic services. The percentage of the domains for the indigenous population is considerably higher (but recall that domains of this group do not use intra-government provisioning). Fig. 5.5 breaks down the numbers for domestic DNS providers. Telstra, as Australia's largest telecommunications company by market share [109], is the most commonly used DNS provider. Macquarie Telecom is the second most utilized provider, followed by the Centre for Information Technology and Communication [110]) and WebCentral. While all previous, mostly common used providers are domestic, 14 domains for the general population rely on the US-based company Verizon. For the domains for the indigenous population, the order is similar, except for two providers that domains for the general population never use and that are not as well known (OPC IT and Three-AMWeb); also, Optus is not used at all. Out of the 33 non-leading providers observed,

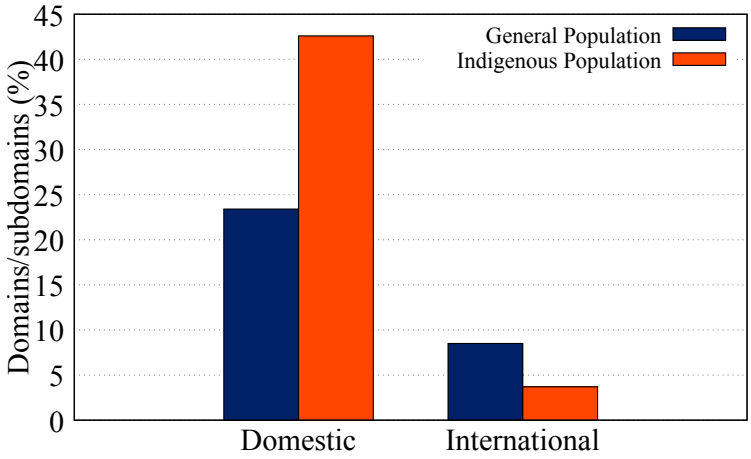

Figure 4.5: Use of non-leading providers.

22 are domestic. Our primary finding here is the curious lack of intra-government provisioning for sites for the indigenous population, which comes with (or results in) the comparatively more common use of more domestic providers. On the whole, a diverse range of non-leading DNS providers is used for both the general and indigenous populations, with limited reliance on non-Australian companies.

4.4 Limitations

Our study has several limitations owing to the early stage of our work.

- **Indirect dependencies:** Naturally, dependencies higher up the chain of DNS delegations have an impact on availability and security properties as well. Our analysis of indirect dependencies has only begun. So far, we have found several cases where a leading DNS provider is actually a delegation from a non-leading one. Understanding precisely in which cases this is problematic is the subject of ongoing work.
- **Longitudinal observations:** Our current study is a snapshot in time. It would be helpful to study the DNS dependencies over a more extended period to understand the dynamics of DNS provisioning.
- **Small sample:** There are significantly fewer domains for the indigenous population than for the general population. This is expected, but one needs to pay attention

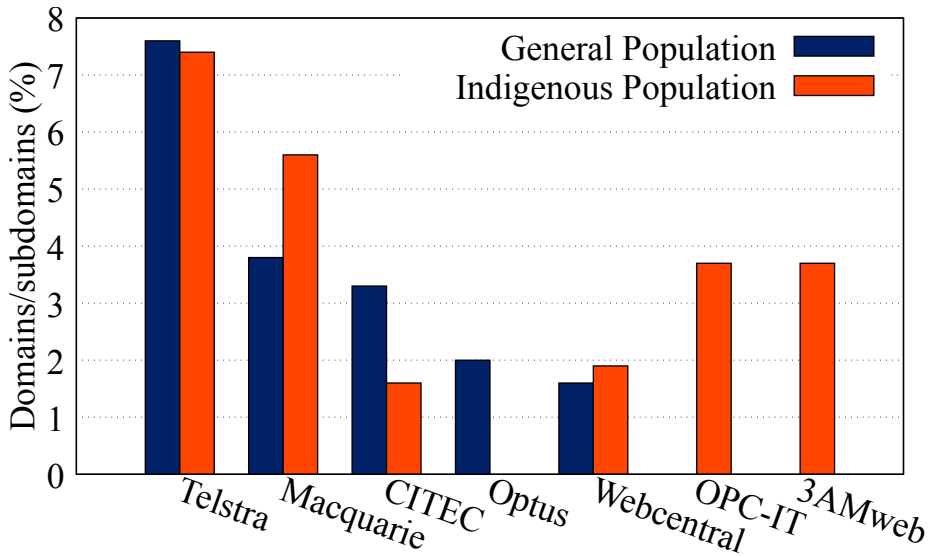

Figure 4.6: Domestic providers.

when comparing small percentages between the groups. It may also offer a partial explanation for the smaller variety of non-leading providers we find for this domain group.

- **Specific focus:** Our findings are specific to the Australian indigenous population and the local DNS landscape. While our study is centered on analyzing DNS dependency among Australian indigenous governmental domains compared to the general population, our methodology can be adapted for broader applications. The approach we have utilized to assess DNS dependencies is transferable to other vulnerable groups within and beyond Australia. Contextual considerations should be considered when considering the applicability of results.
- **Other forms of outsourcing:** We currently use DNS names and the WHOIS to identify the operators of authoritative name servers. However, future work will also need to investigate the ownership of the IP ranges where the authoritative nameservers reside. Although one would not expect many such configurations, it is possible to hide the identity of an actual DNS operator to varying degrees. For example, it is possible to outsource DNS provisioning to organizations that hint at their existence in neither the names of the authoritative nameservers nor the WHOIS. This would be a possibility in the case of sub-contracting. Similar forms of sub-contracting may also occur between different branches of government and be hidden in what we currently call intra-government provisioning.

- **Geographical distribution of the DNS servers:** Another important limitation in this study is the lack of disclosure regarding the geographical distribution of DNS servers, specifically whether they are located inside or outside of Australia. A concentration of DNS servers in a limited number of geographical locations may increase the vulnerability of services to regional outages and geographically targeted attacks.
- **Security vulnerabilities:** The reliance on a few dominant DNS providers creates significant risks, including the potential for single points of failure and vulnerability to cyberattacks. Indigenous domains may be more prone to these risks due to limited redundancy in their DNS provider configurations.

4.5 Summary

This chapter found a significant concentration of DNS services among a few providers for both domain groups: about half of the domains in each group use leading providers. Only some domains for the general population use a multi-provider setup; otherwise, this approach to increase availability and resilience is never used. The leading providers differ between the domain groups, with Amazon being more commonly used in the group of domains for the general population and Microsoft in the one for the indigenous population. Cloudflare is also much more common for the latter group. The company's free tier may be a reason, although this needs to be investigated in more detail. Domains for the indigenous population also use a smaller number of leading providers overall; but here, we need to caution that the number of domains in this group is much smaller. The possibly most interesting difference between the two domain groups can be found in the use of intra-government provisioning. The latter does not occur for domains for the indigenous population but is common for domains for the general population.

Implications: We set out to identify possible disparities in the DNS dependencies for sites for different population groups. We find evidence that dependencies for the indigenous population are indeed differently configured, and we view our evidence

as indicative of different provisioning concepts being employed. However, the exact implications of this are much less clear. In particular, does this result in a tangible digital divide? It seems clear to us that follow-up measurements will be needed to decide this question. In the following, we offer some more detailed thoughts. The lack of intra-government provisioning for indigenous population domains is noteworthy, but there may be practical or legal reasons why we do not find such setups. A qualitative study could shed light on this. As single-provider setups are so common, it is too early to speak of a digital divide in terms of availability. In particular, it is unclear whether intra-government provisioning or the use of smaller domestic providers will improve availability, which can be decided with Internet measurements.

We observed some lack of provider diversity among both domain groups, particularly in the case of leading providers used by domains for the indigenous population. Here, Cloudflare was also more common (possibly because of their free tier). Together with the fact that over 40% of indigenous domains use domestic DNS providers, this may indicate a desire to improve DNS resolution but an inability or unwillingness to move to the cloud. Again, a qualitative study could help illuminate this. Finally, we observe that nearly half of the domains use domestic DNS providers (non-leading or intra-governmental), across both domain groups, which means less reliance on international corporations. In this respect, the nature of the divide is different (non-leading vs. intra-governmental provisioning), but not the quantity. More precisely, we argue that it is a worthwhile undertaking to add measurements of digital divides to the agenda, using both quantitative and qualitative methods. In addition to investigating DNS dependencies, we recognize the significance of considering other measurements that might contribute to a comprehensive assessment of the digital divide. These include availability measurements by using datasets such as Common Crawl [111] or OONI (Open Observatory of Network Interference) [112], routing measurements, and measuring the use of web content management systems. Data from active DNS measurement (OpenINTEL [113]), passive DNS observation, or data from CT (Certificate Transparency) [114] may also be helpful data sets. In the future, we need to qualitatively assess the criticality of services for different population groups and explore the correlation between popularity and criticality. However, it is important to note that the statistical significance of popularity in the case of less popular domains remains unclear. Based on our preliminary results, we have started investigating more in-depth,

beginning with indirect dependencies. We plan to continue with more detailed investigations of the various setups to understand possible reasons and weaknesses. This will include long-time monitoring of availability and changes in providers. We will also analyze which services tend to be supported by intra-government provisioning. Finally, we plan to extend our analysis to other countries around the globe.

Chapter 5

Uncovering Hidden Security Vulnerabilities: Exploring the Impact of Centralization on DNS Dependencies

This chapter investigates the impact of internet centralization on DNS provisioning, particularly its effects on vulnerable populations such as the indigenous people of Australia. We analyze the DNS dependencies of Australian government domains that serve indigenous communities compared to those serving the general population. Our study categorizes DNS providers into leading (hyperscaler, US-headquartered companies), non-leading (smaller Australian-headquartered or non-Australian companies), and Australian government-hosted providers. Then, we build dependency graphs to demonstrate the direct dependency between Australian government domains and their DNS providers and the indirect dependency involving further layers of providers. Additionally, we conduct an IP location analysis of DNS providers to map out the geographical distribution of DNS servers, revealing the extent of centralization on DNS services within or outside of Australia. Finally, we introduce an attacker model to categorize potential cyber attackers based on their intentions and resources. By considering

attacker models and DNS dependency results, we discuss the security vulnerability of each population group against any group of attackers and analyze whether the current setup of the DNS services of Australian government services contributes to a digital divide.

5.1 Motivation

The concept of Internet centralization [19] refers to consolidating control over the Internet's infrastructure and services. The Domain Name System (DNS) is a core Internet service. In the context of the DNS, centralization refers to the control that is held by a limited number of dominant companies [115] such as Amazon Web Services (AWS) Route 53 [116], Microsoft Azure [117], Cloudflare DNS[118], Google Cloud DNS [119], and Akamai Edge DNS [120]. This centralization leads to secondary effects on Internet services and infrastructure, such as the availability of the daily services used by a large population of Internet users [121].

In this regard, the significance of authoritative name servers becomes apparent in the context of service dependencies. The latter term refers to a situation where a service's functionality or effectiveness relies on another's correct functioning. Hence, the efficiency and accuracy of the DNS resolution process are highly dependent on the performance and security of the authoritative name servers. A single point of failure is a major risk in such a system. For example, if an authoritative name server in a centralized setting fails, it can disrupt access to many websites and online services despite redundancies. This is not impossible, and scenarios with cascading failures due to centralization have been observed in the recent past [122, 123].

In this chapter, we ask how failures due to DNS centralization can impact different population groups. We choose the Australian setting as our focus area for two reasons: first, Australia has a sizable vulnerable population, namely the indigenous community. Second, Australia has a highly developed digital infrastructure, and many government

services are available online. We analyze the DNS dependencies of Australian government domains that serve Indigenous communities compared to those serving the general population. Our study makes three main contributions:

- **Digital Divide:** We examine whether there is a difference in the DNS service providers used for indigenous and general population domains. Our purpose is to understand whether such variations contribute to a possible digital divide between the general population and indigenous peoples.
- **Geographical Dependencies:** We analyze the geographical distribution of DNS servers linked to Australian governmental domains. We focus particularly on servers operated by hyperscaler companies and aim to determine these servers' physical locations, distinguishing those within Australia from those abroad.
- **Security Problems:** By considering the DNS dependency results and the provider IP networks, we discuss the vulnerability of each domain group to different types of attacker models. We categorize potential attackers into representative groups of varying strengths and analyze their potential impact on different domain groups.

Our methodology consists of four main phases: Australian government service identification, data collection, retrieving authoritative name servers, and constructing DNS dependency graphs. We categorize DNS providers into leading (hyperscaler, US-headquartered companies), non-leading (smaller Australian-headquartered or non-Australian companies), and Australian government-hosted providers. While Chapter 4 focused exclusively on the digital divide, in Chapter 5, we broadened our scope significantly to include security analyses and analysis of geographic dependencies.

The chapter is structured as follows: We begin with background information on DNS and related concepts, followed by a review of related work. We then detail our methodology and present our results, focusing on direct dependency analysis, indirect dependency analysis, and geographic analysis. The discussion section interprets these results in the context of the digital divide, geographical dependencies, and security implications, including an analysis of attacker models. We conclude with a summary of our findings and their significance for DNS provisioning in Australia.

Data Collection The data collection method was explained in detail in Chapter 4; here,

Algorithm 1: Extracting keywords from URLs to identify domains in each services category.

Create C_i; Set of initial keywords in category i
$D_i \leftarrow \emptyset$; Set of domains for category i
$K_i \leftarrow C_i$;
while *true* **do**
    $Check \leftarrow 0$;
    Use K_i to crawl up to n URLs and save them in U
    for $\forall u \in U$ **do**
        Extract domain d
        if $d \notin D_i$ **then**
            $D_i \leftarrow D_i \cup d$;
            $Check \leftarrow 1$;
        end
        Extract new keywords K_i
        $C_i \leftarrow C_i \cup K_i$;
    end
    if $Check == 0$ **then**
        Return;
    end
end

we provide a brief overview. Our objective is to create two lists: one with the domain names of Australian government websites that provide services to the general population and one with domain names of Australian government websites that provide services for the indigenous populations. To the best of our knowledge, there are no existing open-access data sets for this purpose. We adopt a desk research approach to identify the domains of interest. While we go beyond second-level domains and consider subdomains (which may have their own authoritative name servers), we use the general term "domain" or "domain name" to refer to all of these jointly. We undertook the following steps in the first quarter of 2023. To achieve two distinct sets of domain names for the indigenous and the general population, we perform the steps below in two rounds. In the first round, we add the following indigenous-related terms: *indigenous*, *Aboriginal people and Torres Strait Islanders*, and *first nations* to keywords to collect domain names dedicated to services for the indigenous population. In the second round, we use keywords without these terms to capture domain names for the general public.

1. **Initialization:** By *manual* investigation, we identify 16 categories of services offered by the Australian government, including healthcare, disability support, education programs, and housing support [105]. The category names serve as the primary set of keywords to facilitate the search for relevant domains and websites.
2. **Web search:** We use Google to fetch pertinent governmental websites using our seed keywords. We restrict our search to websites with the *.gov.au* suffix to guarantee we include only official government websites.
3. **Crawling:** We download the top 100 Google search results and store them.
4. **Keyword extraction:** We employ a word cloud technique to extract the top five most prevalent and contextually relevant words from each relevant web page. The relevancy check is performed manually. These extracted keywords are then compared with existing keywords in the set, and new keywords are added to the set for further web search.
5. **Domain names:** We also add the domain names of the sites to our list if we identify them (manually) as relevant.
6. **Iteration:** We iterate through steps 2-5 until we can identify no additional keywords or domain names.

This process is illustrated in Algorithm 1. Once the domain names are obtained, we also perform manual validation to ensure that the collected domain names align with the intended target audience. We finally obtain two lists with unique and relevant domains, each for the respective target audience (**448 domains** for the general population group and **54 domains** for the indigenous group; the list of these domains and their DNS records are uploaded to IEEE DataPort [108] for public access.

5.2 Authoritative Name server Retrieval

This section involves obtaining the authoritative name servers for the domains we collected. We develop a custom script based on command-line tools to obtain the DNS records, specifically focusing on NS records that indicate the authoritative DNS servers for each domain. We send queries to all 13 root servers [124] as of the 22nd of February 2023 [12, 125]. Once we have obtained the name servers for the domains, our next

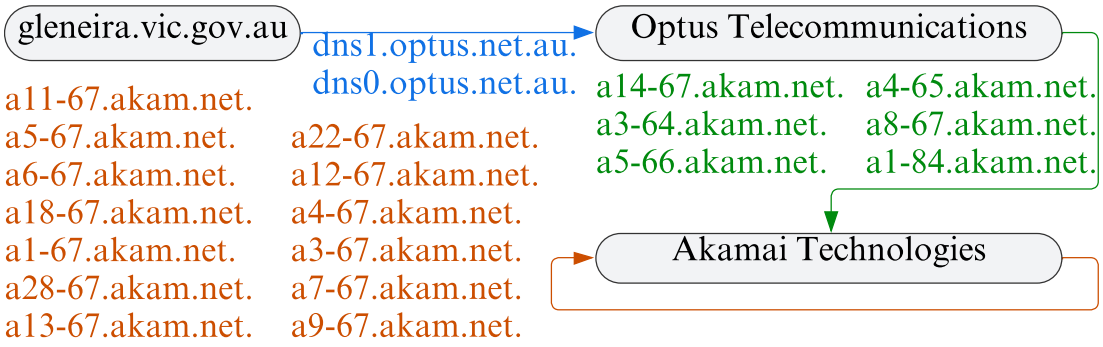

Figure 5.1: DNS Dependencies: lines represent name servers.

objective in this stage is to establish two chains of dependencies—one for indigenous domains and another for general domains. This involves querying the name servers to acquire subsequent name servers, continuing this process until we reach the final authoritative name servers in the chain. This approach gives us a high chance of capturing the majority of the available NS records. Additionally, we leverage WHOIS to collect information about the provider company associated with each captured name server. The results obtained from this stage are utilized to visualize the DNS dependency graph in the next stage.

5.3 Dependency graph construction

We construct two dependency graphs that illustrate the chain of dependencies between the DNS providers and the Australian government domains for both the general population and indigenous people. The are two types of relationships between domains and their name servers: direct and indirect dependencies. **Direct dependency** refers to a domain's immediate reliance on the name servers that are explicitly designated in its DNS configuration without considering any further dependencies of those name servers themselves. An indirect dependency, on the other hand, arises when the name servers directly associated with a domain further rely on other name servers or DNS providers for their own resolution. This creates a chain of dependencies involving intermediate providers. This process results in a chain of dependencies from the domain/subdomain to the final authoritative name servers as well as the governmental

websites and the final DNS providers, respectively. Fig. 5.1 illustrates an example of how nodes are connected in the dependency graph. Optus is the DNS provider of *gleneria.vic.gov.au.* Optus itself depends on Akamai as one of the hyperscaler-type providers. By visualizing the DNS dependency graph, we gain a clear and comprehensive understanding of how these entities are interconnected, allowing us to expose the impact of DNS dependencies on the indigenous and general domains.

5.4 Analytical results

We present our findings on DNS dependencies and their geographical distribution for Australian government domains serving the general population and indigenous communities.

5.4.1 Direct Dependency Analysis

Direct dependency, referring to the immediate relationships between Australian government domains and their primary DNS providers, was investigated in Chapter 4. In this chapter, we provide a more detailed explanation. We analyze the distribution of these dependencies across different types of providers, including leading global companies, local Australian providers, and government-hosted services. Our findings reveal potential disparities in DNS provisioning between domains serving the general population and those targeted at indigenous communities. In Fig. 5.2, the dependency relationships between domains and DNS providers are illustrated, with providers classified as **leading**, **non-leading**, and **governmental** types. Leading DNS providers refer to the dominant and well-established companies in the industry, often associated with tech giants or hyperscalers. Examples of leading DNS providers mentioned in this chapter include AWS Route 53, Microsoft Azure DNS, Cloudflare DNS, Google Cloud DNS, and Akamai Edge DNS. These providers have a significant market share and are known for their extensive infrastructure and global presence. Non-leading DNS providers, on the other hand, refer to smaller or lesser-known companies that

offer DNS services. These providers may have a more limited market share compared to the leading providers and may operate on a regional or national scale. Examples of non-leading DNS providers in the Australian context are Telstra [126], Optus [127], and Webcentral [128]. Governmental DNS providers, in this chapter, are those that are operated and managed by the Australian government sector. In this chapter, these providers are specific to the government sectors that are used to host DNS services for some government websites and domains.

Table 5.1 provides a summary of dependencies on third-party DNS providers for both general and indigenous domains. The data is divided into two categories: general and indigenous populations, encompassing a total of 448 and 54 domains, respectively. The analysis encloses various aspects of dependency, including the reliance on leading providers, non-leading providers (both domestic and international), as well as governmental providers. The analysis also examines two key aspects of DNS provider dependency: *critical dependency* and *diversified dependency*. Critical dependency refers to a domain's reliance on a single DNS provider for its name resolution services. When a domain critically depends on a single provider, it becomes vulnerable to any disruptions, outages, or security issues affecting that provider. In contrast, diversified dependency refers to a domain's reliance on multiple DNS providers, i.e., a site distributes its DNS infrastructure across different providers. This can enhance its resilience and availability, as a site is less likely to be impacted by issues affecting a single provider. It also allows for redundancy and fail-over mechanisms. Our results reveal a considerable amount of critical dependencies where just one DNS provider is used, with 92% and 100% for the general and indigenous domains, respectively. Specifically, 48.9% of general domains and 53.7% of indigenous domains rely on leading providers. Conversely, non-leading providers account for 31.3% of dependency in the general population and 46.3% in the indigenous populations. Moreover, the percentage of dependency on non-leading Australia-headquartered providers is almost doubled for indigenous people compared to the general domains. The dependency on non-leading non-Australian providers is relatively low for both groups at 8.5% and 3.7% for the general and the indigenous populations, respectively. Furthermore, the results reveal that 25.2% of domains in the general population are dependent on intra-government providers, something we do not find for the indigenous group. Additionally, a critical dependency on non-leading Australian providers is observed in

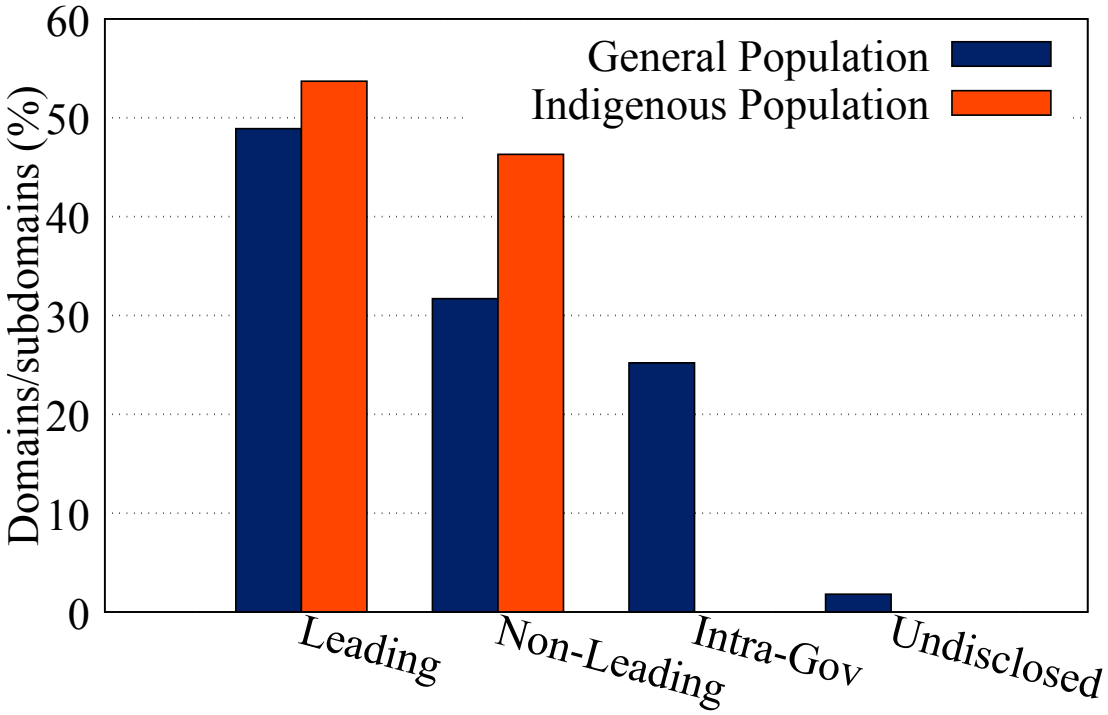

Figure 5.2: Dependency on various DNS provider types: general vs. indigenous domains.

20.8% of general domains and 42.6% of indigenous domains. These findings demonstrate a significant prevalence of critical dependencies on DNS providers in Australia, particularly within the indigenous population, where 100% of domains are critically dependent on a single provider. For 2% of the general domains, we were unable to identify the exact companies that act as DNS providers from the information in the DNS records or from the WHOIS (as information there was redacted). Around 50% of the domains serving both general and indigenous populations depend on leading DNS providers. This indicates that few providers handle a large portion of DNS services which could lead to potential risks of service outages during large-scale attacks. On the other hand, the other half of the indigenous services and 31.3% of general services rely on non-leading DNS providers.

Fig. 5.3 and Table 5.2 specifically focus on the utilization of leading DNS providers by Australian governmental websites and show the percentages of domains that depend on specific leading providers. For instance, 21.4% of general domains rely on Amazon DNS services, while only 7.4% of indigenous domains do the same. The

Table 5.1: Direct and indirect dependencies on third-party DNS providers. Percentages reflect the proportion of the total number of domains that depend on the given provider type and do not sum to 100% due to overlapping dependencies.

	General (448)		**Indigenous (54)**
Provider Type	Direct	Indirect	Direct
Leading	219 (48.9%)	18 (4%)	29 (53.7%)
Non-Leading	140 (31.3%)	12 (2.7%)	25 (46.3%)
Non-Leading (domestic)	105 (23.4%)	10 (2.2%)	23 (42.6%)
Non-Leading (int'l.)	38 (8.5%)	4 (0.9%)	2 (3.7%)
Intra-Governmental	113 (25.2%)	N/A	N/A
Not-disclosed	8 (1.8%)	3 (0.7%)	N/A
Critical Dependency	Direct	Indirect	Direct
Any provider type	412 (92%)	26 (5.8%)	54 (100%)
Leading	202 (45.1%)	17 (3.8%)	29 (53.7%)
Non-Leading	122 (27.2%)	9 (2%)	25 (46.3%)
Non-Lead (domestic)	93 (20.8%)	7 (1.6%)	23 (42.6%)
Non-Leading (int'l.)	29 (6.5%)	2 (0.4%)	2 (3.7%)
Diverse Dependency	Direct	Indirect	Direct
Any provider type	36 (8%)	3 (0.7%)	N/A
Multiple Leading	8 (1.8%)	N/A	N/A
Multiple Non-Leading	2 (0.4%)	2 (0.4%)	N/A
Multiple Gov sections	6 (1.3%)	N/A	N/A
Intra-Gov & Leading	1 (0.2%)	N/A	N/A
Intra-Gov & Non-Leading	18 (4%)	N/A	N/A

analysis also reveals the usage of other leading DNS providers, such as Microsoft, Cloudflare, Akamai, UltraDNS, Google, DNSimple, and EasyDNS, with varying proportions across the two population groups. That is, 31.5% of indigenous websites use Microsoft DNS providers. For general domains, this is around 17%. The rate of dependency on Cloudflare DNS providers for the indigenous population is 83% more than the share of the general population. In contrast, the general population uses Akamai, UltraDNS (Neustar), Google, and DNSimple providers in less than 5% of DNS services. However, indigenous websites do not use them. EasyDNS is only used by the indigenous-focused domains (5%). The final row of the table indicates a relatively small number of general domains that depend on two DNS providers, namely Amazon

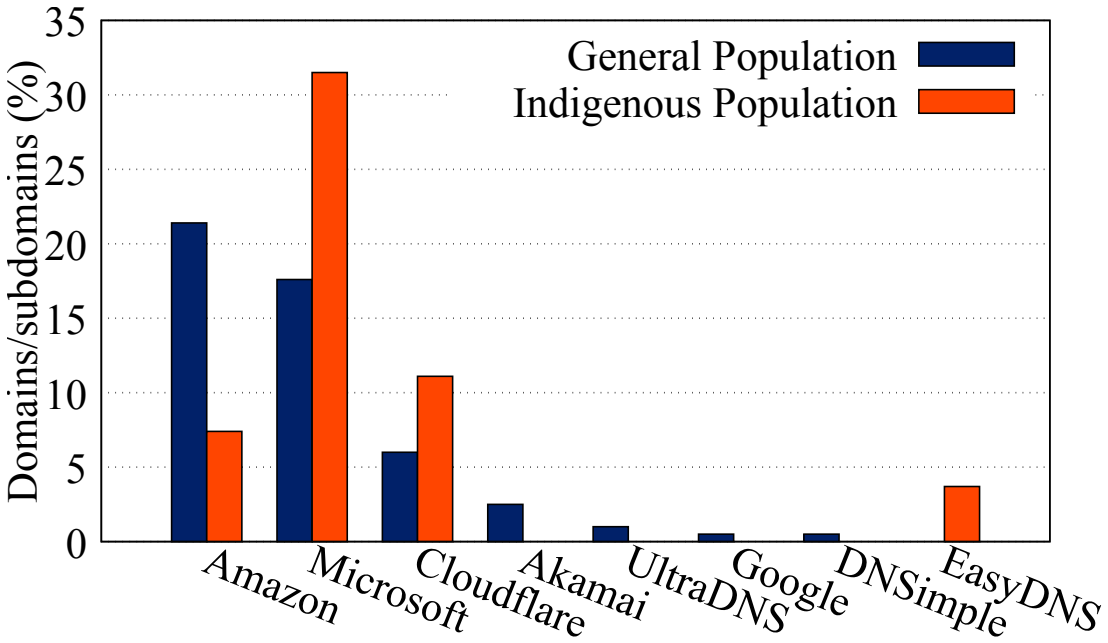

Figure 5.3: Dependency on leading DNS providers: general vs. indigenous domains.

and Microsoft. This configuration is not observed for the indigenous group.

Similarly, the dependency on non-leading DNS providers utilized in Australia is shown in Fig. 5.4. Approximately 24% of general domains rely on local DNS providers, whereas this proportion increases to over 40% for indigenous domains. In contrast, indigenous websites mainly depend on less than 5% of non-leading international DNS providers. This is close to 9% for the general population.

Fig. 5.5 and Table 5.4 focus on the most frequently used domestic DNS providers. These results indicate that a significant proportion of both the general and indigenous populations depend on non-leading DNS providers. Telstra emerges as the most commonly used DNS provider for both populations, followed by Macquarie Telecom and CITEC. Telstra, Australia's largest telecommunications company (by market share[109]), has slightly less than 8% of the share from both general and indigenous domains. The share of Macquarie Telecom is less than 4% for the general population domains, and it is used by 6% of indigenous websites. CITEC (Centre for Information Technology and Communication [129] [110]), the Queensland Government's primary ICT services provider, has only a 4% share of general domains. This number is only 2% for the indigenous population. Webcentral has an equal share of less than 2% from

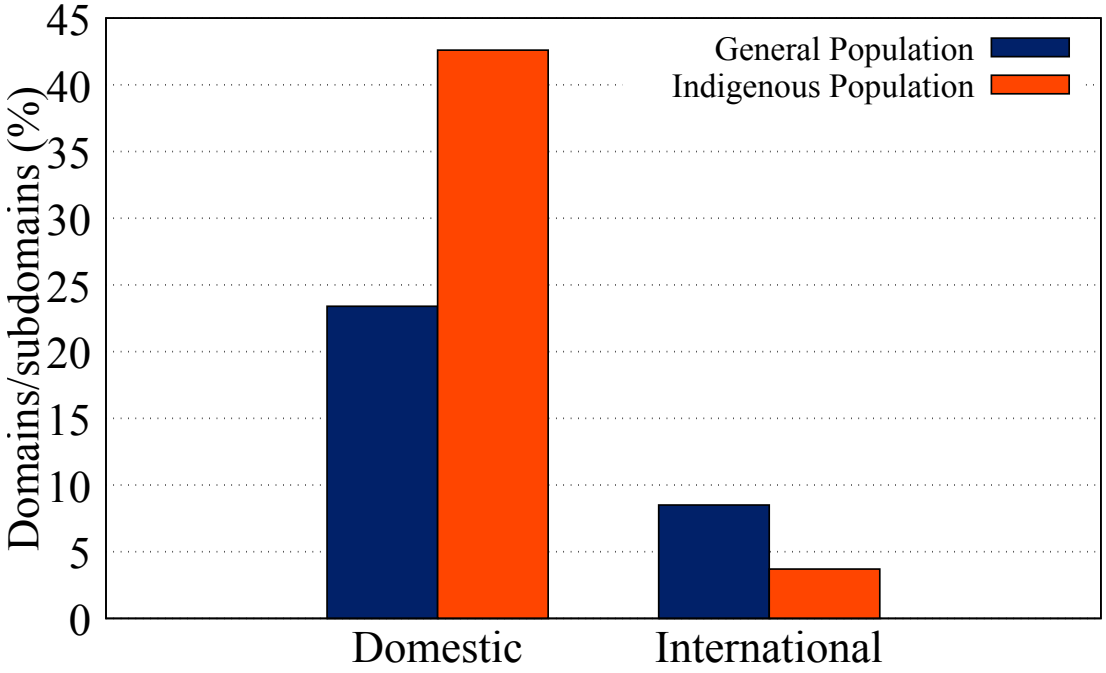

Figure 5.4: Dependency on non-leading DNS providers: general vs. indigenous domains.

both general and indigenous domains. 4% of indigenous DNS services are dependent on less-known Australian DNS providers such as OPC IT and ThreeAMWeb. General domains use Optus (which is a large subsidiary of a Singaporean telecommunications company). A significant outage of the Optus network occurred on 8 November 2023, affecting numerous customers and services across Australia [130]. This incident shows the potential risks associated with relying on a single non-leading DNS provider. No indigenous domains were found to depend on non-Australian companies, while 14 general domains rely on the US-based company Verizon. These results show that there is a diverse range of DNS providers used by both populations, with some reliance on domestic providers and a limited reliance on non-Australian companies, and indigenous domains expose a higher reliance on non-leading providers.

In Fig. 5.6, we illustrate how domains are distributed among the general population with multiple providers. Notably, none of the indigenous domains are dependent on multiple DNS providers. As shown in this figure, 20% of domains have two distinct leading DNS providers (Amazon and Microsoft), and less than 5% have two different non-leading providers. Additionally, less than 20% of domains have intra-government

Table 5.2: Leading DNS providers for the indigenous and general population domains

Number of domains	**General**	**Indigenous**
Total	219(48.9%)	29(53.7%)
Dependent on Amazon	11(21.4%)	4(7.4%)
Dependent on Microsoft	79(17.6%)	17(31.5)%)
Dependent on Cloudflare	27(6%)	6(11.1%)
Dependent on Akamai	11(2.5%)	0
Dependent on UltraDNS	4(0.9%)	0
Dependent on Google	1(0.2%)	0
Dependent on DNSimple	1(0.2%)	0
Dependent on EasyDNS	0	2(3.7%)
Dependent on both Amazon and Microsoft	6(1.3%)	0

Table 5.3: Direct/indirect dependencies, general domains.

No. of domains	**Direct**	**Indirect**
5	Not-disclosed	Leading
1	Intra-gov	Leading
11	Intra-gov	Non-leading (domestic&international)
11	Intra-gov	Non-leading (domestic)
2	Intra-gov	Non-leading (international)
12	Non-leading	Leading
2	Non-leading	Non-leading (international)

DNS providers that point to different government sectors. Over half of general domains use both a government-hosted provider and a third-party DNS provider.

5.5 Indirect DNS Dependency

Indirect dependencies represent a more complex web of relationships in DNS provisioning. We investigate these to disclose hidden vulnerabilities and centralization patterns that may not be evident from direct dependencies alone. We explore how indirect dependencies occur in general domains (Table 5.3), focusing on the relationship between a domain's immediate DNS provider type (namely, leading and non-leading

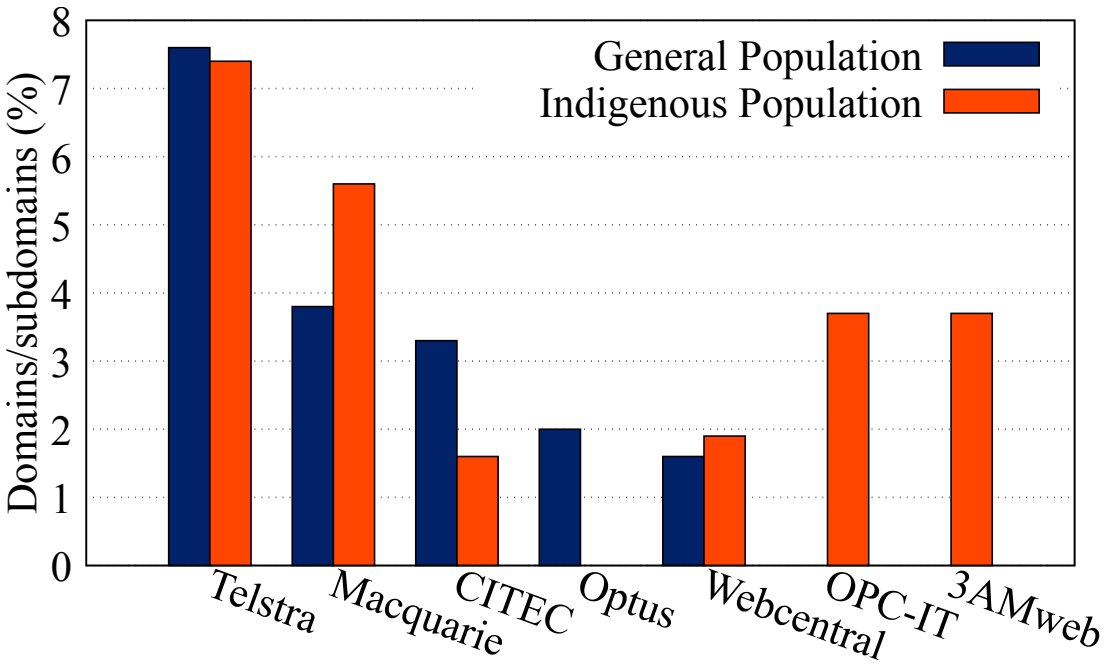

Figure 5.5: Dependency on most commonly used domestic DNS providers: general vs. indigenous populations.

provider, with either Australian headquarters or non-Australian headquarters) and its reliance on the indirect DNS provider type, and discuss the implications for service reliability and security. In our observation, there is no evidence of indirect dependencies for government domains dedicated to indigenous people. Our examination reveals a dependence on major companies such as Amazon and Akamai. Interestingly, one domain employing government-hosted DNS services also has an indirect dependency on Amazon. While certain domains directly depend on intra-government services, their indirect dependencies often lie with non-leading, both domestic and international, providers such as Telstra, Optus, and CITEC. This diversity in indirect providers could offer redundancy and reduce single points of failure while also introducing complexity and potential exposure to vulnerabilities. A significant observation is the indirect dependency of 12 domains on leading providers despite their direct reliance on non-leading ones. This pattern underscores the pervasive influence of hyperscalers such as Amazon and Akamai in the DNS infrastructure. The contrast between direct and indirect dependencies reveals a critical aspect of DNS infrastructure management. For general population domains, this leads to centralization around a few hyperscaler entities.

Table 5.4: Non-leading DNS providers.

Number of domains	General	Indigenous
Total	142(31.7%)	25(46.3%)
Dependent on Telstra	34(7.6%)	4(7.4%)
Dependent on Macquarie Telecom	17(3.8%)	3(5.6%)
Dependent on CITEC	15(3.3%)	1(1.6%)
Dependent on Verizon	14(3.1%)	0
Dependent on Optus	9(2%)	0
Dependent on Webcentral	7(1.6%)	11(1.9%)
Dependent on NEC	7(1.6%)	0
Dependent on OPC IT	0	2(3.7%)
Dependent on Three AM Web	0	2(3.7%)

5.6 Geographical IP Locations of DNS Providers

The physical location of DNS servers plays a crucial role in the performance and resilience of online services. By examining the geographical distribution of IP addresses associated with DNS servers, especially those serving specific domains such as Australian government domains dedicated to indigenous peoples, we can disclose patterns of reliance on third-party providers. We map the geographical distribution of DNS servers associated with Australian government domains and analyze how server locations differ for domains serving the general population versus those focused on indigenous communities. For this, we leveraged the **IP2Location** database. In addition to the geographical location, IP2Location also provides information about the usage type of IP addresses. The usage types relevant to our analysis are as follows: 1) ISP (Internet Service Provider): IP addresses associated with Internet service providers 2) DCH (Data Center/Web Hosting/Transit): IP addresses belonging to data centers, web hosting companies, or transit providers (the latter category is less likely to contain authoritative name servers) 3) CDN (Content Delivery Network) 4) GOV (Government): IP addresses assigned to government entities, implying that the DNS server is operated by a government organization.

A problem that is common to geolocation databases is that IP addresses, especially of hyperscalers, may be attributed to the wrong country, in addition to IP allocations also

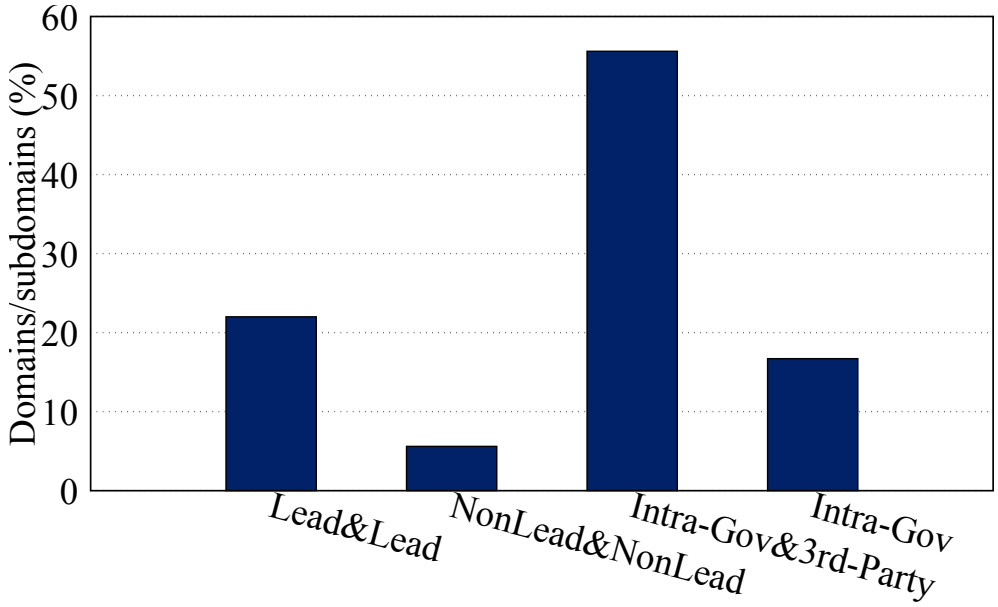

Figure 5.6: Diversified dependencies in general domains.

ever-changing (which may change the usage type). Given our small sample size, our results must be taken with a grain of salt. For example, the geographical distribution of the DNS servers associated with Australian government domains for indigenous peoples spans across five countries: Australia, Canada, Hong Kong, Japan, and the United States. The Australian setting is much more plausible than that of Hong Kong or Japan, which may be such cases of wrong attributions. However, it would still imply the use of powerful providers and hyperscalers. In Australia, the presence of ISP/DCH usage types indicates a mix of local Internet service provisioning and data center hosting. The United States is also a plausible hosting location: it hosts the highest number of DNS servers, indicating a heavy reliance on US-based infrastructure. However, it is still possible that these servers are physically located in Australia. An international dependency may well be more vulnerable, both from a technical and operational point of view.

Both domain groups have DNS servers primarily located in Australia and the United States, as shown in Fig. 5.7, 5.8, 5.9, and 5.10. The breakdown into usage types in Fig. 5.8 indicates an overall diversified approach to DNS service provisioning for the general domains. The presence of ISPs suggests localized service delivery, while CDNs (Content Delivery Networks) highlight efforts to optimize content distribution and

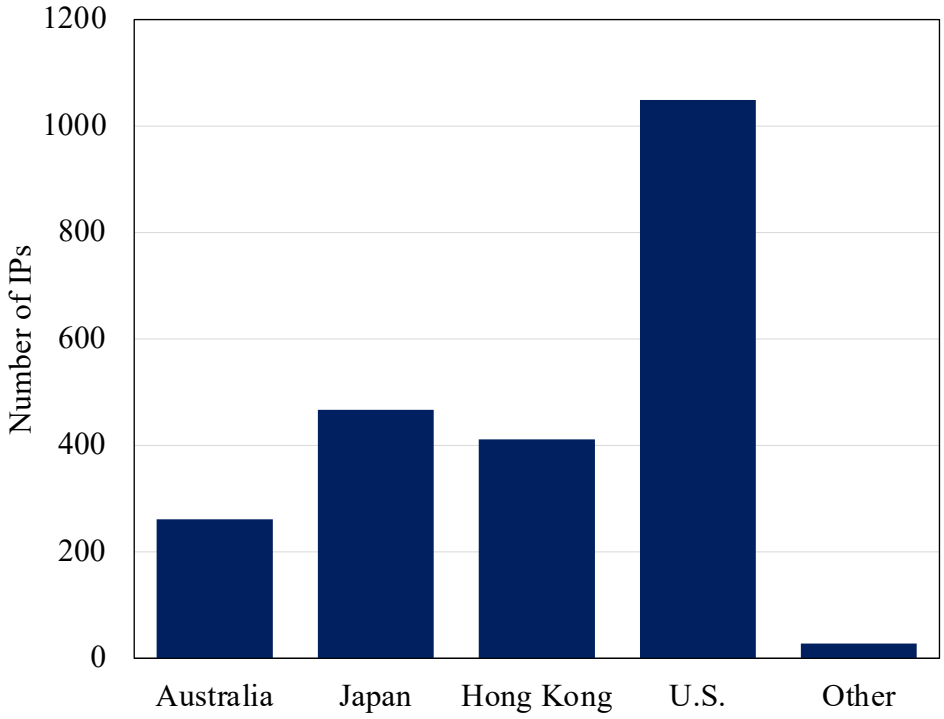

Figure 5.7: Geographical location of general domains' NS.

performance. A small proportion is classified under "Gov", indicating government-managed servers, which may signify an intention for self-reliance and control over DNS infrastructure. This hints at a balance between globalized service provision and localized service delivery and at strategic approaches to deployment and performance optimization. The presence of government-managed servers is also noteworthy, especially as they are not present in the indigenous-focused infrastructure. The DNS infrastructure for the general population seems to have a greater emphasis on diversity and global distribution compared to that of the indigenous peoples. The inclusion of government servers for the general population suggests an element of direct governmental control or self-reliance. For the indigenous-focused domains, the emphasis appears to be on a mix of local and international service provisioning with a significant dependency on DCH and CDN providers.

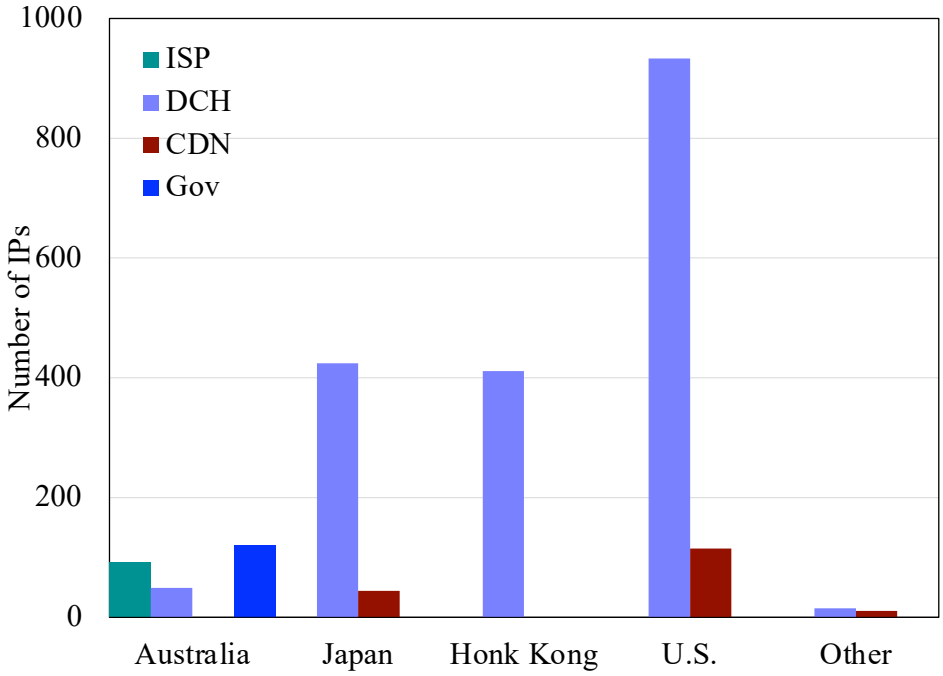

Figure 5.8: Usage type for general domains' NS.

5.7 Discussion

Our analysis of DNS dependencies for Australian government domains reveals implications for the digital divide, geographic dependencies, and security vulnerabilities. We discuss these findings in the context of our three main research questions.

5.7.1 Digital Divide

Our results provide evidence of a digital divide in DNS provisioning between general and indigenous domains. The main difference is the critical dependency on single providers: 100% of indigenous domains rely on a single DNS provider, compared to 92% of general domains. This higher dependency makes indigenous domains more vulnerable to service disruptions and potential attacks. Furthermore, indigenous domains show a greater reliance on non-leading DNS providers (46.3%) compared to general domains (31.3%). While this might reflect a preference for local services, it also exposes these domains to potential reliability issues, especially given that some of

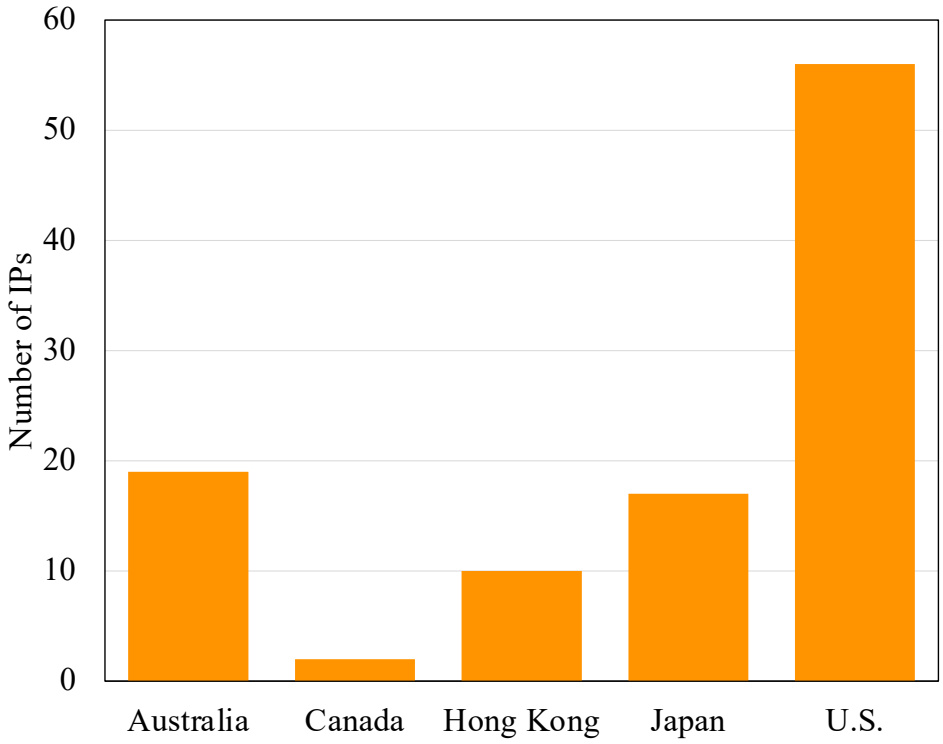

Figure 5.9: Geographical location of indigenous domains' NS.

these providers may have limited resources for security and maintenance. Surprisingly, none of the indigenous domains utilize government-operated DNS infrastructure. This absence requires further investigation to understand its causes and implications for security and reliability.

5.7.2 Geographic Dependencies

The geographical distribution of DNS servers reveals another aspect of DNS provisioning, although we caution that geolocation information can be misleading, and hence our results must be viewed in that light. Nevertheless, some trends are evident. General population domains benefit from a broader global spread. In contrast, indigenous domains show a higher concentration of servers in fewer locations. The limited geographic diversity for indigenous domains could lead to increasing vulnerability to regional outages, imposing potential latency issues for users in different regions and greater exposure to geographically targeted attacks. The preference for domestic DNS

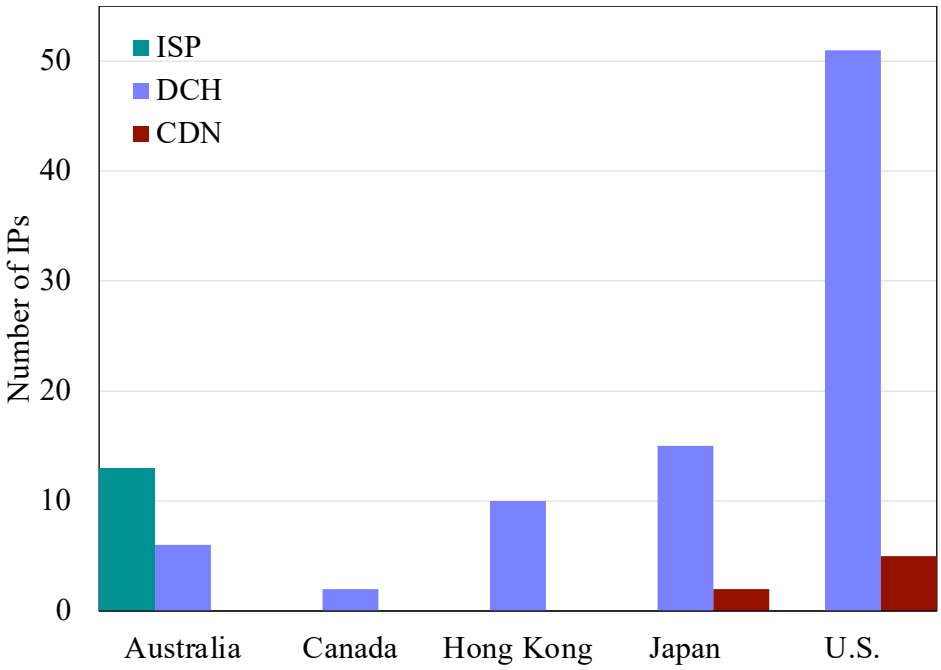

Figure 5.10: Usage type for indigenous domains' NS.

providers by over 40% of indigenous domains, while potentially driven by a desire for geographically closer services, may increase the vulnerability to region-specific disruptions or attacks.

5.7.3 Security Analysis

For our analysis of how various forms of dependencies lead to a site being more or less vulnerable, we define three different attacker models. We categorize potential attackers into three representative groups of varying strengths: lone wolves (including individual attackers) [131, 132], the organized fringe attackers (such as hacktivists) [133, 134, 135, 136], and global actors (e.g., nation states) [137, 138]. Each group has different resources, intentions, and targets, ranging from limited-resource attackers targeting specific populations to well-funded, expert attackers operating on a global scale.

- **lone wolf:** We call our first attacker "lone wolf—weak but targeted". This attacker

does not have many resources (hardware, network access, deep technical knowledge, finances) at their disposal. However, they act out of a strong, personal motivation, and they only target particular organizations. An example of such strong motivation would be the hate of a particular societal minority (for instance, racist sentiments, resentment of the LGBTQ+ community, etc). Their likely target is organizations representing or aiding such minorities, and their tools are limited to what limited financial means can afford—for example, paying an underground service a small amount of money to stage a limited DDoS attack. In the context of our results, this attacker is more concerned with the indigenous domains. The lack of DNS service diversity makes indigenous domains particularly vulnerable. Even with limited resources, a lone wolf attacker could potentially disrupt services by focusing on the single provider used by a domain.

- **organized fringe:** We call our second attacker "organized fringe—limited and targeted". This attacker is typically a group that can raise some relevant resources (financial contributions by members, donations) and gain access to relevant knowledge that can be used in an attack, such as members with a deep technical background. The motivation is broader than in the previous attacker model and represents, for instance, political fringe movements: anti-democratic forces that also target minority groups or direct resentment at them in a bid to win more support. The likely targets are the same as before as well as various government services. This attacker can invest substantially more resources—a more powerful DDoS attack is a possibility, as is a takeover of the IT services of some smaller companies with less stringent security. This attacker type is of concern for both domain groups, but still more so for the indigenous domains. The concentration of indigenous domains on Australian providers makes them particularly attractive targets.
- **global actor:** We call our third attacker a "global actor—resourceful and impact-oriented". This attacker has means at their disposal that one would usually associate with organized crime or, at worst, even rogue nation-states. They can make substantial investments into sustained attacks and have access to skilled hackers. The motivation here may be much broader than before: the attacker may be less interested in hurting smaller organizations and more focused on causing widespread damage that has economic ramifications and will draw attention from politics, often at a global level. Damage to services aiding minorities may be a welcome side effect, but the targets would more commonly be large, professional enterprises that offer cloud services—in

the more extreme cases, even global hyperscalers, such as the DDoS attack on Amzon's DNS service in October 2019 [139]. While these sophisticated attackers pose a threat to all domains, the lack of diversity and redundancy in indigenous domain DNS infrastructure makes them particularly vulnerable to widespread, high-impact attacks.

To mitigate the risks mentioned above, we recommend the following:

1. Implementing multi-provider setups for all domains, especially indigenous ones.
2. Diversifying the geographic locations of DNS servers.
3. Incorporating government-operated DNS services as an additional layer of resilience.

These measures would enhance the security against at least the two less powerful forms of attackers and improve overall service reliability.

5.8 Summary

This chapter investigated the impact of Internet centralization on DNS provisioning and its security implications, particularly focusing on the indigenous and general populations in Australia. The study categorized DNS providers into leading (hyper-scaler, US-headquartered companies), non-leading (smaller Australian-headquartered or non-Australian companies), and Australian government-hosted providers and constructed dependency graphs to demonstrate direct and indirect dependencies of Australian government domains on these providers. Our main findings included as follows:

1. **Digital Divide:** There is a significant difference in the DNS service providers used for indigenous and general population domains, which imposes potential disparities in service quality and reliability. Indigenous domains revealed a higher dependency on non-leading DNS providers compared to general population domains, indicating a digital divide in DNS provisioning.
2. **Geographic Dependencies:** The geographical distribution of DNS servers revealed the concentration of servers in fewer locations for indigenous domains, which could

lead to increased vulnerability to regional outages and potential latency issues. The DNS servers for general population domains benefit from a broader global spread.

3. **Security Problems:** The study demonstrated vulnerabilities of each domain group to different attacker models. Indigenous domains are more vulnerable due to their critical dependency on single providers and limited geographic diversity. The analysis categorized potential attackers into lone wolves, organized fringe attackers, and global actors, each posing different levels of threat to the DNS infrastructure.

The chapter implied that addressing these disparities and vulnerabilities requires implementing multi-provider setups, diversifying geographic server locations, and incorporating government-operated DNS services to enhance security and service availability, which could be the future research effort on DNS dependency.

Chapter 6

Design and Implementation of a Communication-efficient Secure Aggregation Protocol for Stable Federated Learning Systems

The centralization of Internet services has benefits, such as providing a trusted source, as highlighted by RFC9518 [121]. However, it also raises significant security concerns, particularly regarding data privacy. In this chapter, we focus on centralization from the perspective of Federated Learning (FL) [46] systems and their vulnerability to model inversion attacks. Federated Learning (FL) is a promising distributed learning framework designed for privacy-aware applications. FL trains models on client devices without sharing the clients' data and generates a global model on a server by aggregating model updates. Traditional FL approaches risk exposing sensitive client data when plain model updates are transmitted to the server, making them vulnerable to security threats such as model inversion attacks. In such attacks, the server can infer a client's original training data by monitoring the changes in the trained model across different rounds. Google's Secure Aggregation (SecAgg) protocol addresses this threat by employing double-masking techniques, secret sharing, and cryptographic computations in

honest-but-curious and adversarial scenarios, even when client dropouts occur. However, in scenarios without an active adversary, the computational and communication costs of SecAgg significantly increase as the number of clients grows.

This chapter proposes ACCESS-FL, a communication- and computation-efficient secure aggregation method designed for honest-but-curious scenarios in stable FL networks with a limited rate of client dropout. ACCESS-FL reduces computation and communication costs to a constant level, independent of network size, by generating shared secrets between only two clients and eliminating the need for double masking, secret sharing, and cryptographic computations. To evaluate the performance of ACCESS-FL, we conduct experiments using the MNIST, FMNIST, and CIFAR datasets. The evaluation results demonstrate that our proposed method significantly reduces computation and communication overhead compared to state-of-the-art methods such as SecAgg and SecAgg+.

6.1 Motivation

Federated Learning (FL) [46] has emerged as a promising approach for privacy-preserving collaborative learning, enabling multiple parties to train machine learning models without sharing sensitive data. FL allows distributed clients to collaboratively train a model while keeping their data decentralized and private. In FL, each client trains a global model using its local dataset and then sends the trained model update to a central server, which builds a new global model by aggregating all updates. While FL protects user privacy by avoiding direct data sharing, it still faces challenges and vulnerabilities [140], such as model inversion attacks [47], in which an honest-but-curious server [48, 49] may reconstruct the original client data by reverse-engineering the local model weights. To address these privacy threats, Google proposed the Secure Aggregation (SecAgg) protocol [50], a secure multi-party computation (MPC) [51] method based on Diffie-Hellman (DH) key agreement [52] and Shamir's secret sharing [53, 54]. In SecAgg, each client generates a shared secret for every other client participating in a training round. This secret is used in a Pseudo-Random Generator (PRG) function

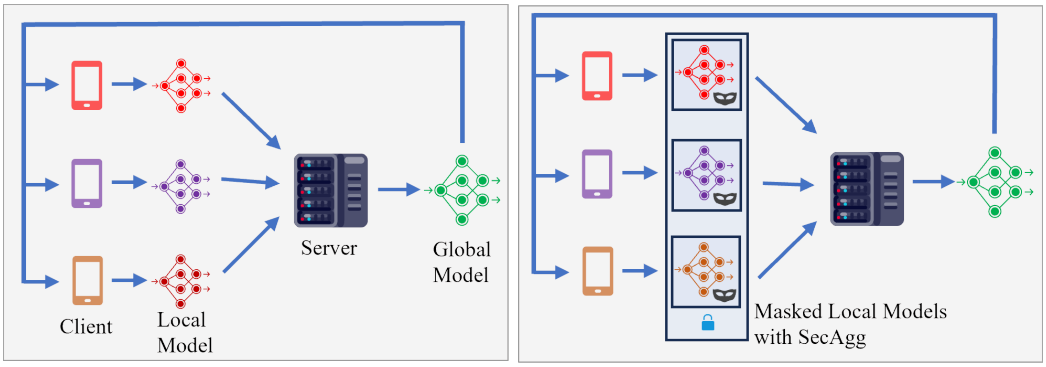

Figure 6.1: Comparison between vanilla FL and FL with SecAgg.

[141, 142] to create a mask that is added to the model update, concealing individual updates during transmission and preserving privacy through a double-masking technique. This double-masking technique, which involves shared masks between clients and a self-mask for each client, ensures that the server learns only the aggregated result, rather than the model weights of individual clients. SecAgg also manages client dropouts and message delays by reconstructing the shared mask of dropped clients and the self-mask of participating clients. Figure 6.1 illustrates the SecAgg approach compared to the traditional FL scheme.

SecAgg is effective against active adversaries [48, 143], where malicious clients and the server actively collude to determine the masks added to model updates, aiming to infer sensitive information about participating clients' private data. However, as the network size grows, SecAgg incurs significant computational costs. Each client must perform cryptographic operations and key generation for every other client in the network. Additionally, the server's communication overhead increases as it reconstructs masks for dropped clients. To address these limitations, Google proposed SecAgg+ [144], an improved version of SecAgg. In SecAgg+, instead of generating shared secrets with every other client, the server generates a random k-regular graph for $k = \log(|C|)$, where $|C|$ is the number of clients. Clients then generate shared masks with their neighbors in the graph. While SecAgg+ reduces communication and computation costs compared to SecAgg, it still incurs unnecessary costs in stable networks, making it more suitable for unstable networks with frequent client dropouts. In both SecAgg and SecAgg+, the server must reconstruct the self-masks of participants.

Even when there are no client dropouts and shared masks cancel out automatically, the server still needs to remove the self-masks of participants. This requirement adds unnecessary overhead in stable networks with infrequent client dropouts.

To design an efficient secure aggregation mechanism suitable for FL systems with stable network conditions, we propose ACCESS-FL, an enhanced secure aggregation protocol based on key agreement. ACCESS-FL is intended for honest-but-curious FL scenarios with stable network conditions, such as fraud detection in financial applications [145], privacy-preserving systems against money laundering by IBM [146], and AI applications in healthcare systems [147]. In these applications, the network typically exhibits low delay variations and limited client dropouts. The main contributions of this chapter are:

- By generating shared secrets between only two client devices, regardless of the network size, we significantly reduce the communication and computation overhead of ACCESS-FL compared to state-of-the-art methods in honest-but-curious FL scenarios with stable network conditions. ACCESS-FL eliminates the need for each client to perform cryptographic operations and key generation for every other client in the network.
- We introduce a dynamic client pairing mechanism based on a deterministic function and a secret seed, ensuring that the pairing remains unknown to the server, thereby enhancing data privacy (Figure 6.2).
- We simplify the secure aggregation process in ACCESS-FL by eliminating the double-masking technique and the associated cryptographic computations. This approach maintains the same level of communication as in traditional federated learning for the server, with additional communication required only in the case of client dropouts.

Our experiments on the MNIST dataset [148], Fashion-MNIST [149], and CIFAR-10 [150] demonstrate that ACCESS-FL significantly reduces communication and computational costs for both clients and the server while maintaining the same level of security against model inversion attacks as SecAgg and SecAgg+ in honest-but-curious FL scenarios with stable network conditions and a limited number of client dropouts. The implementation and source code for ACCESS-FL are publicly available at [151].

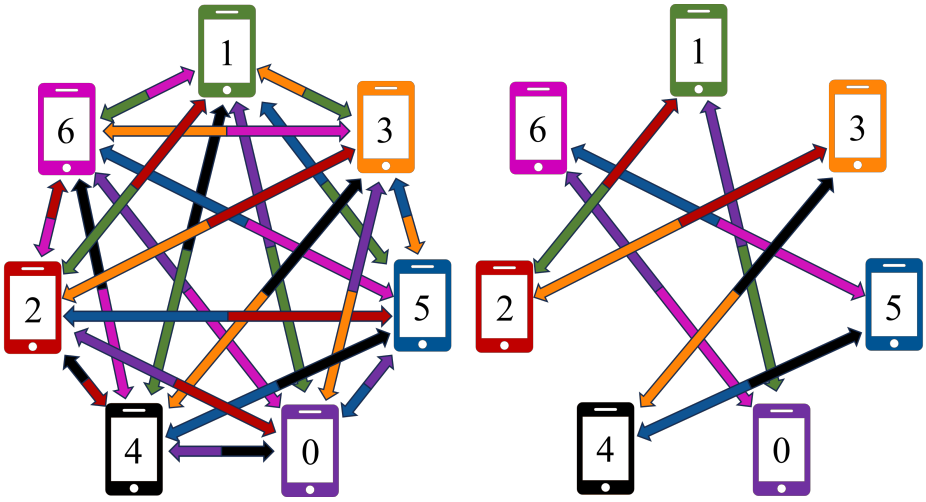

Figure 6.2: SecAgg (left) versus ACCESS-FL (right) in finding pairs and creating shared secrets.

The rest of this chapter is organized as follows: Section 6.2 provides an overview of SecAgg and SecAgg+. Section 6.3 describes the ACCESS-FL protocol in detail. Section 6.4 presents a theoretical analysis of the correctness and security of ACCESS-FL. Section 6.5 reports the experimental evaluation of ACCESS-FL, comparing it to SecAgg and SecAgg+. Section 6.6 discusses the limitations of ACCESS-FL and offers suggestions for future research. Finally, Section 6.7 concludes the chapter and summarizes our contributions.

6.2 Preliminary Study: SecAgg and SecAgg+

In this section, we provide an overview of Google's Secure Aggregation (SecAgg) protocol and its improved variant, SecAgg+. We explain the process of message passing and discuss its challenges in the context of honest-but-curious FL scenarios with stable network conditions, where client dropouts are limited and delay variations are low. Table 6.1 summarizes the main notations used throughout this chapter.

6.2.1 Message Passing in SecAgg

The section explains the SecAgg protocol as follows:

(1) **Broadcasting the global model:** The server broadcasts the initial global model to clients.
(2) **Key pair generation:** Each client i generates **two** private-public key pairs as (SK_i^1, PK_i^1) and (SK_i^2, PK_i^2). Then, it sends its public keys to the server.
(3) **Broadcasting public keys:** The server broadcasts public keys to all clients.
(4) **Client-side preparation:** Each client $i \in C$ generates a random element b_i, then divides b_i and SK_i^1 into $|C|$ parts and assigns each part to a client pair $j \in C$ ($b_{i,j}$, $SK_{i,j}^1$). Then, Client i encrypts a message $(i||j||b_{i,j}||SK_{i,j}^1)$ for each pair j (by using a key generated from SK_i^2 and PK_j^2) to create a cipher text $e_{i,j}$. Finally, the client sends all generated $e_{i,j}$ to the server.
(5) **Distribution of cipher texts:** The server collects these cipher texts and puts participating clients in the C_1 set. Here, we assume that $|C_1| = |C|$, hence the server sends $(|C| - 1)$ encrypted values to every client.
(6) **Masked model generation:** Each client i creates $(|C| - 1)$ shared secrets with every other client j by using SK_i^1 and PK_j^1. Then, client i expands these created shared secrets and its random element b_i by the pseudo-random generator function PRG to create a self-mask m_i and shared masks $m_{i,j}$ $\forall j \in C_1$. By using these masks, client i computes the masked model w_i^{mask} from its trained model w_i, which is sent to the server.
(7) **Participants awareness:** The server creates a set C_2 from clients that sent their masked models. Then, the server sends the set C_2 to all the clients $j \in C_2$. Then, each client i identifies the participants and decrypts the received encrypted values $e_{j,i}$ by using a key generated from SK_i^2 and PK_j^2. Thus, client i obtains $b_{j,i}$ $\forall j \in C_2$ for participants and $m_{j,i}$ $\forall j \in C_1 \setminus C_2$ for dropped-out clients, and sends them to the server.
(8) **Global model aggregation:** The server gathers $(|C| - 1)$ portions of random elements of participants and dropped-out clients, then expands each reconstructed value by PRG to generate self-mask m_j $\forall j \in C_2$ and shared masks $m_{j,i}$ $\forall j \in C_1 \setminus C_2$.

Table 6.1: Declaration of main notations

Notations	Definition
n	Round number of FL
C	The list of participating clients in an FL round
$\|C\|$	Number of clients in the FL system
$param$	Public parameters for key pair generation
SK_i	Private key of client i
PK_i	Public key of client i
PK_{all}	List of public keys
PRG	Pseudo Random Generator function
$s_{i,j}$	Shared secret generated by SK_i and PK_j
fp_i	First pair of client i for generating shared secret
sp_i	Second pair of client i for generating shared secret
$m_{i,j}$	Shared mask between client i and client j
w_i	Trained model of client i
w_i^{masked}	Masked model of client i
$W_{aggregated}^{masked}$	Aggregation of all masked models
G	Global model

Finally, it aggregates the global model by:

$$\sum_{i \in \{C_2\}} w_i^{masked} - \sum_{i \in \{C_2\}} m_i + \sum_{i \in C_2, j \in \{C_1 \setminus C_2\}} m_{j,i}. \tag{6.1}$$

Based on [50], in SecAgg, the communication cost for a client and the server are $O(|C|)$ and $O(|C|^2)$, respectively, where $|C|$ is the number of participating clients.

6.2.2 Challenges of SecAgg in Stable FL

- **Handing Client Dropout or Delayed Messages:** In a practical FL scenario, issues such as the unstable Internet connection can interrupt the process of creating the global model. In SecAgg, Google uses a double-masking technique to ensure that each client's model updates remain secure against model inversion attacks, even in cases of user dropout or delayed updates. However, this involves several cryptographic operations, including two key pairs generation (public and private keys for creating shared secrets and secure communication), creating shared and self secrets,

utilizing Shamir's secret sharing (a method for dividing a secret into parts), conducting encryption and decryption operations, and calling a pseudo-random generator function for generating the masks. All mentioned steps can significantly increase computational and communication costs, even in stable networks with low dropout rates. For example, in a healthcare FL system, where patient data is being used for training, an unstable connection can cause delays. While the double-masking technique ensures privacy, it significantly increases computational and communication overhead, and each client must compute extensive cryptographic operations.

- **High computation cost:** In SecAgg, each of the $|C|$ clients generates $|C|-1$ shared secrets regarding every other client device and creates a unique random element for itself. These values are then expanded using a pseudo-random generator function to create shared and individual masks. This process significantly increases the computational complexity of the client device to $O(|C|^2)$ which becomes particularly challenging in large FL systems such as Google Gboard with one billion clients. The server also has substantial computation overhead in SegAgg, as it must reconstruct shared masks for dropped-out clients and regenerate self-masks for participating clients (those who have sent their masked updates). These tasks elevate the server's computational cost to $O(|C|^2)$.
- **High communication cost:** The communication cost for each client in SecAgg is $O(|C|)$ due to sending two public keys and encrypted shares of both the private key and the random element, along with their masked model updates. The server has a higher communication cost, as it is responsible for distributing encrypted values to clients and broadcasting public aggregated model updates. This leads to a quadratic increase in communication cost ($O(|C|^2)$) for the server. In large-scale FL scenarios, such as smart city applications with thousands of participating devices, the server's communication load becomes a significant bottleneck.

6.2.3 SecAgg+

SecAgg+ [144] is an improvement over SecAgg designed to reduce the computational and communication costs associated with secure aggregation. Instead of generating shared secrets regarding every client, the server generates a random k-regular graph for

Algorithm 2: Client-side algorithm in ACCESS-FL to generate a key pair (in the first training round).

Wait for the server to send the initial G;
Wait for a trusted third party to send public parameters;
Client i generates a key pair with public parameters
$(SK_i, PK_i) \leftarrow$ `key_gen(`*param*`)`;
Store SK_i securely;
Send PK_i to the server;
Wait to receive PK_{all} from the server;

$k = (\log |C|)$, where $|C|$ is the number of clients. Clients only generate shared masks with their neighbors in this graph. While SecAgg+ reduces the costs compared to SecAgg, for a much larger number of clients (e.g., billions), it still leads to unnecessary overhead in stable networks with low rates of client dropouts.

6.2.4 Advantages of ACCESS-FL

While SecAgg and SecAgg+ provide secure aggregation mechanisms for FL, they face challenges in terms of high computation and communication costs in honest-but-curious FL scenarios with stable network environments. Our proposed ACCESS-FL protocol aims to address these limitations by the following:

1) reducing the number of shared masks to only **two masks per client regardless of the network's size**
2) eliminating the need for executing cryptographic operations with high computational cost such as encryption, decryption, or Shamir's secret sharing on client devices
3) eliminating the need for sharing any values other than one public key and the masked model on client devices, (which leads to a substantial decrease in the number of messages transmitted through the network)
4) reducing the server's computational cost by removing the need for handling mask cancellation

6.3 ACCESS-FL Protocol

In this section, we introduce ACCESS-FL, our proposed communication and computation-efficient secure aggregation protocol for generating masked models during the FL process. ACCESS-FL is designed for stable networks with limited client dropouts and low delay variations in honest-but-curious scenarios. The main stages of ACCESS-FL are: 1) initialization, 2) pairs selection, 3) generating shared masks, 4) local training and mask application, and 5) updating the global model. In the following, we explain details of each stage.

1) Initialization: First, the server broadcasts the initial global model to all clients. Each client receives common public parameters from a *trusted third party*; these public parameters are generated through the key agreement algorithm with a specified key size. Then, each client generates a unique public-private key pair using these public parameters through the DH key generation function and sends the public key to the server. The server then broadcasts the list of public keys to all clients. **In ACCESS-FL, each client generates only one key pair once through all FL rounds (algorithm 2)**. However, in SecAgg, each client needs to create two key pairs in every FL training round. This is because, in SecAgg, the server reconstructs the self-mask of participants and the shared masks of dropped-out and delayed clients. If a client is delayed, the server in the current round computes its shared masks, and in the next round, if the client remains a participant, the server reconstructs its self-mask. If the key pairs used to generate the shared masks of this client do not change in the next round, the server can possess both the shared masks and self-mask of this client, allowing it to calculate the client's trained model. However, in ACCESS-FL, the server is not responsible for removing masks from the aggregated function. Thus, the server cannot recompute the shared masks of the clients, and there is no need to generate new key pairs in every round. Consequently, **the server broadcasts the public keys once in ACCESS-FL for all FL rounds**. In contrast, in SecAgg, the server needs to broadcast the newly generated public keys in every round. Therefore, computation and communication costs associated with key pair generation and distribution are significantly reduced in ACCESS-FL.

Algorithm 3: Client-side algorithm in ACCESS-FL to find two pairs (during all training rounds).

Client i calculates the distance value to find its pairs
$distance_i^n = \text{RandInt}(\text{set}_i^n)$;
$set_i^n = \{d \mid \forall d \in [1, \lfloor \frac{|C|-1}{2} \rfloor], d \neq \text{distance}_i^{n-1}\}$;
Client i finds its pairs from the sorted participant list
$fp_i \leftarrow (i + distance_i^n) \mod |C|$; # First pair's index
$sp_i \leftarrow (i - distance_i^n + |C|) \mod |C|$; # Second pair's index

2) Pairs Selection: Upon receiving the public keys, each client identifies two peers to create shared secrets with them. These shared secrets are constructed using the peer's public key and the client's own private key. In ACCESS-FL, clients are sorted into a participating list. Hence, each client has a position as an index in this list. **To determine the pair's position, each client uses a deterministic function provided by a trusted third party. This function generates a random integer within the range** $[1, \lfloor \frac{|C|-1}{2} \rfloor)]$ **based on the training round number** (a variable which is known by all clients), ensuring that the distance varies each round. All clients run this distance generator function and calculate the same distance value, ensuring consistency across the network. The distance value ensures that clients have unique pairs each round. To prevent identical pairs in different rounds, the distance generated in each round is a random value within the given domain, excluding the distances used in previous rounds. Once calculated, this position is added to and subtracted from the client's own index in the sorted participant list to identify the two corresponding pair clients for the creation of shared secrets. As an example, **for client** i**, two pair indexes are calculated as (**$(i + distance) \mod |C|$**) and (**$(i - distance + |C|) \mod |C|$**), where** $|C|$ **is the number of clients in the participating list**. This dynamic pair selection process ensures that clients have different pairing partners in each round, enhancing the privacy and security of the protocol. Algorithm 3 shows the process of finding pairs in ACCESS-FL for client i.

3) Shared Masks Generation: After finding pairs, each client uses the private key and the public keys of its peers to generate a shared secret based on the key agreement algorithm. The shared secret, unique to each pair, is then used as the seed for

Algorithm 4: Client-side algorithm in ACCESS-FL to generate its masked model (during all training rounds).

Client i generates shared secret with its two pairs.
$s_{i,fp_i} \leftarrow$ `key_agree(`SK_i, PK_{fp_i}`)`;
$s_{i,sp_i} \leftarrow$ `key_agree(`SK_i, PK_{sp_i}`)`;
Client i creates its shared masks through `PRG` function.
$m_{i,fp_i} \leftarrow$ `PRG(`s_{i,fp_i}`)`;
$m_{i,sp_i} \leftarrow$ `PRG(`s_{i,sp_i}`)`;
Determine signs based on indices
$sign_{fp} \leftarrow$ if $fp_i < i$ then -1 else 1;
$sign_{sp} \leftarrow$ if $sp_i < i$ then -1 else 1;
Client i calculates its masked model.
$w_i^{masked} \leftarrow w_i + sign_{fp} \times m_{i,fp_i} + sign_{sp} \times m_{i,sp_i}$;
Send w_i^{masked} to the server;
Wait to receive the new G from the server;

the PRG function to generate shared masks. Based on each client's index in the participant list, the one with a smaller index gets the -1 coefficient added to its shared masks. Since we use the key agreement algorithm, the shared secret generated by the private key of client i and the public key of client j equals the shared secret generated by client j's private key and client i's public key. These shared secrets are then input into the PRG function, which produces identical outputs given the same input. Thus, $shared\_mask_{i,j} = shared\_mask_{j,i}$. **By using a coefficient of -1 for the client with the smaller index, we ensure that $shared\_mask_{i,j}$ and $shared\_mask_{j,i}$ cancel out each other in the aggregation process**. This approach simplifies the masking process and reduces the computational burden on clients compared to SecAgg's double-masking technique.

4) Local Training and Mask Application: Each client trains the global model by using its local dataset. **ACCESS-FL is designed to be independent of the model and data distribution types, making it versatile for various applications**. After training the model locally, each client applies the masking vectors to its model updates to generate its masked model. The masked model is then sent to the server, ensuring that the server does not have access to the plain-trained model of each client (algorithm 4). This step ensures the privacy of individual clients' models while allowing the server to aggregate the masked models into a global model.

Algorithm 5: Server-side algorithm in ACCESS-FL.

```
# First training round
Broadcast initial G;
Wait for all clients to send their public keys (Algorithm 2);
for ∀i ∈ C do
    PK_all ∪ [PK_i]; # List of public keys
end
Broadcast list of PK_all;
# Second training round onwards
Wait for clients to send their masked models (Algorithm 4);
if number of masked models < |C| then
    # Server updates C with participants
    C ← list of participants who sent masked models;
    # Server sends updated C to all clients
    for ∀i ∈ C do
        Send updated C to client i;
    end
    Wait for clients to send their masked models (Algorithm 4);
end
# Server aggregates masked models
W^masked_aggregated ← 0;
W^masked_aggregated ← Σ_{i∈C} w_i^masked; # Sum of masked models
G ← W^masked_aggregated;
Broadcast new G;
```

5) Global Model Update: In this phase, the server waits to receive the masked models from all clients within a specific time frame. If the server gets the masked models from all clients within this period, it aggregates them to generate the new global model. Algorithm 5 shows the process of ACCESS-FL running at the server. However, if a client drops out or the server receives a masked model after the specified time, the server broadcasts the sorted list of participating clients that have sent their masked models and waits for the new masked model from these participating clients. In this scenario, **all clients recalculate the distance to find new pairs and send their new masked models (algorithm 6). This step is mandatory to remove the shared masks associated with the dropout client**, preventing any deviation in the aggregation result. The goal is to ensure that the aggregation output is equivalent to traditional FL aggregation output. ACCESS-FL is designed for stable FL environments with limited client dropouts and low delay variations, providing the server does not get stuck in the

Algorithm 6: Client-side algorithm in ACCESS-FL for handling client dropout or delayed updates.

Client i calculates the $new\_distance_i^n$
$new\_distance_i^n = \text{RandInt}(\text{set}_i^n)$;
$set_i^n = \{d \mid \forall d \in [1, \lfloor(|C|-1)/2\rfloor], d \neq distance_i^{n-1} \ \& \ distance_i^n\}$;
$distance_i^n \leftarrow new\_distance_i^n$;
Find new pairs with new distance by Algorithm 3;
Calculate w_i^{masked} by Algorithm 4 ;
Send w_i^{masked} to the server;
Wait for the server to send new G;

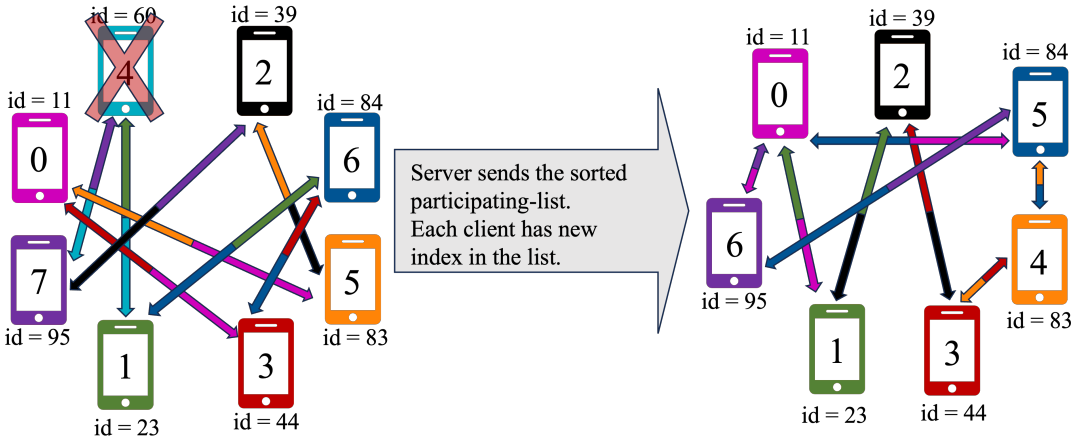

Figure 6.3: Finding new pairs in the presence of a client drop-out.

same training round waiting for new masked models. Figure 6.3 shows an example where a client dropout occurs among 8 clients, and the participants need to find new pairs to generate new shared masks. By handling client dropouts and delayed updates, ACCESS-FL maintains the integrity of the aggregation process while minimizing the computational overhead. Through the stages mentioned above, ACCESS-FL facilitates the privacy-preserving aggregation that allows multiple clients to contribute to an aggregated result without exposing their individual trained models while maintaining low communication and computation costs. The diagram in Figure 6.4 abstracts the mechanisms of SecAgg compared to ACCESS-FL for one round of FL. This figure visualizes the process of generating masked models among four clients participating in a federated learning system. On the left side, representing SecAgg, we observe multiple clients (indicated by lightning bolt icons of different colors), each contributing to the FL process. These clients generate a masked model by combining three elements:

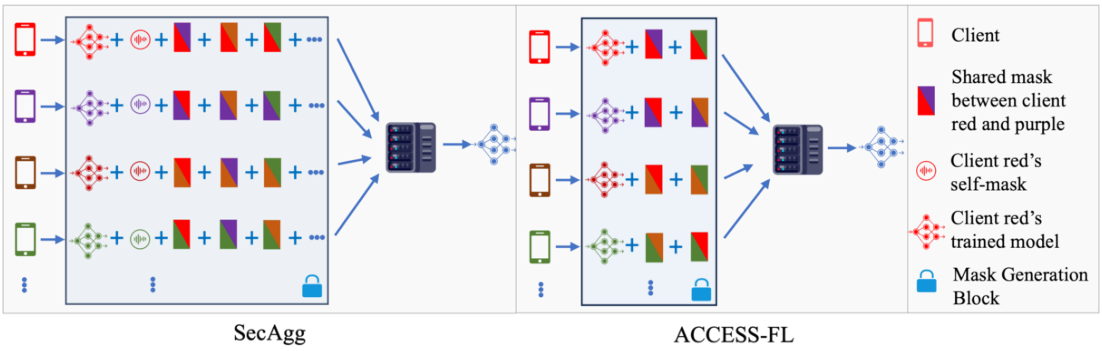

Figure 6.4: Comparison between SecAgg and ACCESS-FL.

their individual trained models' weights (shown by grid icons), shared masks created between themselves and every other client (illustrated by differently colored geometric shapes), and their own self-masks (shown as noise). As shown in the figure, as the number of clients increases, the number of shared masks each client needs to generate increases, leading to high communication and computation costs. Combining these elements results in a double-layered masking technique to conceal the trained models. Each client's masked model is then sent to a centralized server for aggregation. The server outputs an aggregated model (blue grid icon), which incorporates the knowledge learned from all participating clients without revealing any individual client's trained model and data. On the right side, ACCESS-FL is demonstrated. Each client still produces a trained model (grid icons) in this figure, but the masking process is simplified. Instead of creating a shared mask with every other client, each client only generates shared masks with two different clients (represented by the connection between the same colored geometric shapes and grid icons). As illustrated in the figure, the number of generated masks per client is independent of the network size; regardless of the number of clients, each client only needs to create shared masks with two other clients rather than all participants, which results in reduced message volume compared to SecAgg. These shared masks are added to the trained model to create a masked model, which is then sent to the server. The server aggregates these masked models into a new global model (blue grid icon). This diagram shows the contrast in complexity and message volume between the two protocols. SecAgg requires a more significant number of messages to be exchanged, as every client generates shared masks with every other client. ACCESS-FL, however, reduces the communication overhead by limiting the creation of shared masks with two clients, thereby reducing message volume and

potentially increasing the overall efficiency of the FL process. The simplified masking process and reduced message volume in ACCESS-FL highlight its advantages over SecAgg in terms of communication and computation efficiency.

6.3.1 Message Passing in ACCESS-FL

This section analyzes the total number of messages exchanged between the server and C number of clients over n training FL rounds in ACCESS-FL. The process of messages passing in ACCESS-FL is categorized into three main phases as follows: **Phase** I (Initialization): The server **broadcasts** the initial model to all clients. Simultaneously, all clients receive common public parameters from a trusted third party. Then, each client generates a unique public-private key pair from public parameters and sends its public key to the server. Upon collecting all public keys, the server broadcasts a set of all public keys. **Phase** II (Shared mask generation at training round 1): Upon receiving public keys, each client calculates the index of two other clients in the participating list to create shared secrets using their public keys. Client i calculates the index of its pairs as $fp_i = [(i + distance) \mod |C|]$ and $sp_i = [(i - distance + |C|) \mod |C|]$. Here, $distance$ is a random integer within the range of $[1, \left\lfloor \frac{|C|-1}{2} \right\rfloor]$ (where $|C| \geq 6$). We limit the upper domain to $\frac{|C|-1}{2}$ to make sure that the pairs are different; more than $\frac{|C|-1}{2}$ makes the chosen pair equal to the previously found pairs. After finding the pairs, the client generates shared secrets, which are used in a pseudo-random number generator that creates two shared masks (denoted as m_{i,fp_i} and m_{i,sp_i}). **Phase** III (from training round 2): Each client i performs model training and then computes the masked model as $w_i^{masked} = w_i + m_{i,fp_i} + m_{i,sp_i}$, where w is the trained model. The computed w_i^{masked} is sent to the server. Then, the server generates the new global model by aggregating all masked models. The masks cancel out each other due to the pairwise generation of shared secrets since $m_{i,fp_i} = -m_{fp_i,sp_{fp_i}}$ the sum of masked models equals the sum of unmasked trained models. Finally, the server broadcasts the new global model to all clients. Considering all communications after n FL rounds, the total number of messages sent from all clients are $(n + 1) \times |C|$, and from the server are $n + 1$ messages. Considering all rounds, the communication order for each client and the server is $O(1)$. This analysis demonstrates the communication efficiency

of ACCESS-FL, as the number of messages exchanged remains constant regardless of the network size.

6.3.2 Explanation of Core Enhancements

In this section, we explain the core enhancement of ACCESS-FL.

Efficient key pair generation: In ACCESS-FL, one key pair is only required for creating shared secrets, whereas in SecAgg, two key pairs are necessary (one for shared secrets and one for encryption). Additionally, in ACCESS-FL, the key pair is generated once in the initial round. In contrast, in SecAgg, due to the reconstruction of self-masks for participants and the shared masks of dropout clients by the server, each client needs to generate a key pair in every round of FL. In SecAgg, if a client is delayed in sending its masked model, the server assumes it has dropped out. Consequently, every other client sends the portions of the delayed client's private key to the server, allowing the server to reconstruct the shared masks for this delayed client. If the delayed client's model is received in the following FL round, and the client continues to participate, the server receives portions of the client's random element and can calculate the self-mask. If the key pairs remain unchanged, the shared masks for this client also remain unchanged. After two rounds, the server can compute the client's trained model by subtracting the self-mask and shared masks from the masked model. Therefore, for security reasons, clients in SecAgg are required to generate new key pairs every round. However, in ACCESS-FL, clients do not share any information except for the masked model. This means the server cannot reconstruct the client's trained models due to the lack of private keys. Generating a key pair only once eliminates the need for clients to send their public keys to the server and for the server to broadcast these keys. This results in reduced communication and computation costs for both the server and clients. The efficient key pair generation in ACCESS-FL significantly reduces the computational burden on clients and the communication overhead between clients and the server.

Simplified masking techniques: ACCESS has refined the masking process to make it

more communication and computation efficient. These enhancements include using a more compact representation of masks and employing less mathematical computation that requires fewer data to achieve the same level of privacy in honest-but-curious scenarios. In contrast to SecAgg, which uses both shared and self-masks, our enhanced protocol utilizes only shared masks. In Google's protocol, a double masking strategy is necessary because clients share secrets with the server. However, in our proposed method, applying self-masks is not required since we do not share any secrets with the server. The shared masks are generated between pairs of two participant clients. Each pair collaborates to create a masking vector with its peer from the pair client. By eliminating self-masks, our protocol significantly reduces the computational burden on each client. The focus on shared masks simplifies the entire mask generation process. Since each client is only responsible for generating and managing masks with two nodes, the overall complexity of the masking process is reduced. This masking approach reduces the computational cost and the amount of data that needs to be transmitted for masking purposes. The use of shared masks means fewer data packets are required to achieve the same level of privacy, leading to lower communication costs. The simplified masking techniques in ACCESS-FL lead to more efficient computation and communication compared to SecAgg's double-masking approach.

6.4 Proof of Maintaining Aggregation Result Equal to Traditional FL

This section aims to demonstrate that in ACCESS-FL, the output of the aggregated model is maintained compared to traditional FL. The following settings for each client i are considered: 1) w_i represents the trained model of client i. 2) m_{i,fp_i} represents the shared mask between client i and its first paired client denoted by fp_i. 3) m_{i,sp_i} represents the shared mask between client i and its second paired client denoted by sp_i. 4) The equation $m_{i,j} = -m_{j,i}$ holds for any pair of clients i and j where $j < i$.

Creating Masked Models: The idea behind the proof is to use only two shared secrets to mask the individual models before aggregation. Each client i creates a

masked model w_i^{masked} by adding its trained model w_i with the shared masks m_{i,fp_i} and m_{i,sp_i}. The equation $m_{i,j} = -m_{j,i}$ ensures that the shared masks cancel out when summed across all clients. This feature keeps the output of the aggregation function in ACCESS-FL equivalent to the output of the aggregation function in traditional FL. Each client i creates a masked model w_i^{masked} as follows:

$$w_i^{masked} = w_i + m_{i,fp_i} + m_{i,sp_i}. \tag{6.2}$$

Aggregating Masked Models: The aggregation of all masked models across $|C|$ clients, denoted as W^{masked}, is computed by summing up the masked models w_i^{masked} for each client. This aggregate can be decomposed into three separate sums: the sum of the trained models w_i, the sum of the shared masks m_{i,fp_i}, and the sum of the shared masks m_{i,sp_i}, where $0 \leq i \leq |C| - 1$. Thus,

$$W_{aggregated}^{\text{masked}} = \sum_{i=0}^{|C|-1} w_i^{\text{masked}} \qquad = \sum_{i=0}^{|C|-1} \left(w_i + m_{i,fp_i} + m_{i,sp_i}\right). \tag{6.3}$$

We can decompose this into three separate sums:

$$W_{aggregated}^{\text{masked}} = \sum_{i=0}^{|C|-1} w_i + \sum_{i=0}^{|C|-1} m_{i,fp_i} + \sum_{i=0}^{|C|-1} m_{i,sp_i}. \tag{6.4}$$

Cancellation of Shared Masks: Equation $m_{i,j} = -m_{j,i}$ implies that the shared mask generated by client i with client j is equal in magnitude but opposite in sign to the shared mask generated by client j with client i. When these shared masks are summed across all clients, they cancel each other out. That is,

$$\sum_{i=0}^{|C|-1} m_{i,fp_i} = -\sum_{j=0}^{|C|-1} m_{j,sp_j}. \tag{6.5}$$

Thus, each shared mask m_{i,fp_i} pairs with $m_{fp_i,sp_{fp_i}}$ such that:

$$m_{i,fp_i} + m_{fp_i,sp_{fp_i}} = 0. \tag{6.6}$$

Here, sp_{fp_i} is defined as the second pair of the first pair of client i. Hence, summing across all clients:

$$\sum_{i=0}^{|C|-1} m_{i,fp_i} + \sum_{i=0}^{|C|-1} m_{fp_i,sp_{fp_i}} = 0. \tag{6.7}$$

and similarly, for the second paired client,

$$\sum_{i=0}^{|C|-1} m_{i,sp_i} + \sum_{i=0}^{|C|-1} m_{sp_i,fp_{sp_i}} = 0. \tag{6.8}$$

This means for every shared mask m_{i,sp_i}, there exists a corresponding $m_{sp_i,fp_{sp_i}}$ that cancels out. The notation fp_{sp_i} represents the first paired client of the second pair of client i. Thus, the sum of shared masks m_{i,fp_i} and m_{i,sp_i} across all clients becomes zero. This cancellation property is a key feature of ACCESS-FL, ensuring that the shared masks do not affect the final aggregated model.

Equivalence to Trained Models: Therefore, the aggregation of all masked models w_i^{masked} is equivalent to the sum of the individual trained models w_i across all clients:

$$W_{aggregated}^{\text{masked}} = \sum_{i=0}^{|C|-1} w_i. \tag{6.9}$$

The proof presented in this section establishes the correctness of ACCESS-FL, demonstrating that it achieves the same aggregation result as traditional FL while preserving the privacy of individual clients' models through the use of shared masks.

6.5 Evaluation Result

To verify the effectiveness of ACCESS-FL, we conduct experiments using the MNIST, FMNIST, and CIFAR10 datasets. MNIST is the main dataset in this chapter, which

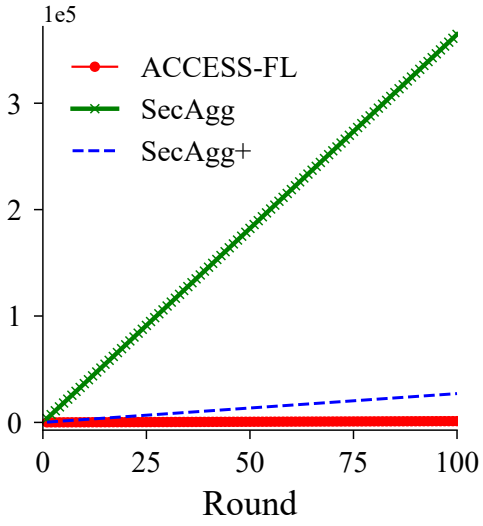

Figure 6.5: Clients to server.

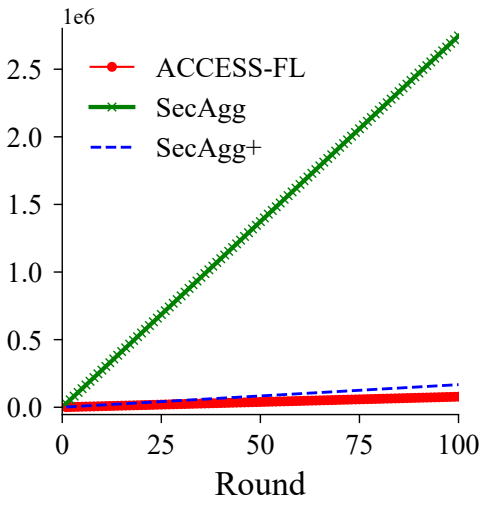

Figure 6.6: Server to clients.

Figure 6.7: Accumulative message size (kB) for MNIST experiments.

contains 60,000 handwritten digit images used for training and 10,000 images for testing. Each image is a 28x28 gray-scale digit, and the goal is to classify the images into one of ten digit classes (0-9). In our experimental setup, we utilized 100 clients and assigned each client only 1 label to simulate a practical Non-Independent and Identically Distributed (Non-IID) scenario; We implement a 2-layer Neural Network (2NN) model consisting of an input layer for the flattened 28X28 pixel images, followed by two dense layers with 200 units and ReLU activation, and a final output layer of 10 units with softmax activation. Each experiments were conducted with 100 communication rounds with SGD [152] optimizer in a learning rate of 0.1. To assess the communication and computation costs of each protocol, we present results based on the accumulated message size, the number of exchanged messages, and the running time for both clients and the server in ACCESS-FL, SecAgg, and SecAgg+.

6.5.1 Communication Cost of ACCESS-FL, SecAgg and SecAgg+

Figures 6.5, 6.6 illustrate the accumulated message size sent from clients to the server and sent from the server to clients, respectively, for protocols ACCESS-FL, SecAgg, and SecAgg+ over 100 rounds. We observe that the total size of the message for each

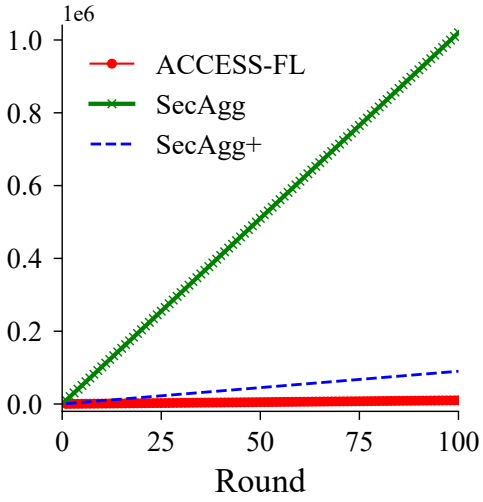

Figure 6.8: Clients to server.

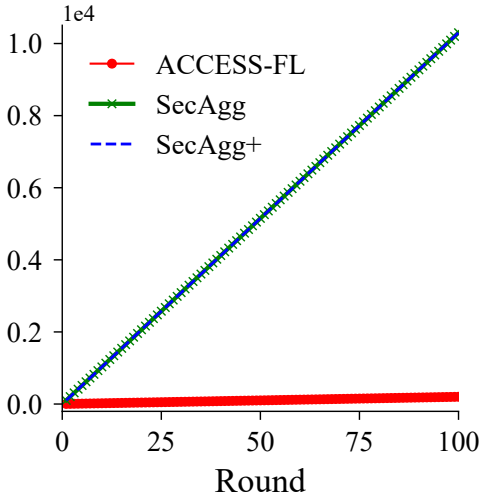

Figure 6.9: Server to clients.

Figure 6.10: Accumulated number of messages for MNIST experiments.

client in ACCESS-FL remains at approximately 0.01 MB through the 100 rounds. And **the communication cost for each client does not increase with the number of participating clients**, as each client only generates shared masks with two pairs, regardless of the number of clients. In contrast, the total message size for each client in SecAgg and SecAgg+ increases with the number of clients because, in both algorithms, the number of pairs that each client generates depends on the network size. In SecAgg, each client pairs with every other client, and in SecAgg+, each client pairs with its k neighbors in the predefined and randomly generated $k - regular$ graph by the server where $k = log(|C|)$ for the $|C|$ number of clients. In SecAgg, the total message size for each client is around 3.5MB by the 100th round. SecAgg+ reduces the message size for each client compared to SecAgg, but it still grows with the network size, reaching around 0.3MB after 100 training rounds. The server's accumulated message size in ACCESS-FL is almost 80MB by the 100th round. However, in SecAgg and SecAgg+, the volume of transmitted messages from the server through the network exceeds about 3000MB and 200MB, respectively, by the end of the 100 rounds.

The accumulated number of messages exchanged between the server and clients is demonstrated in Figure 6.10. The number of messages sent by each client in ACCESS-FL stays constant at around 100 throughout the 100 rounds and demonstrates the scalability of ACCESS-FL, as the communication overhead on each client remains fixed

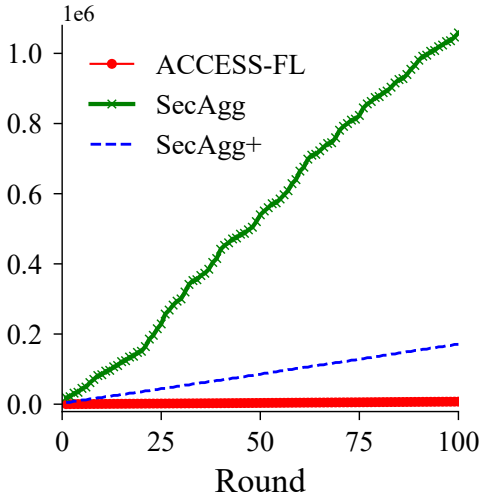

Figure 6.11: Clients to server.

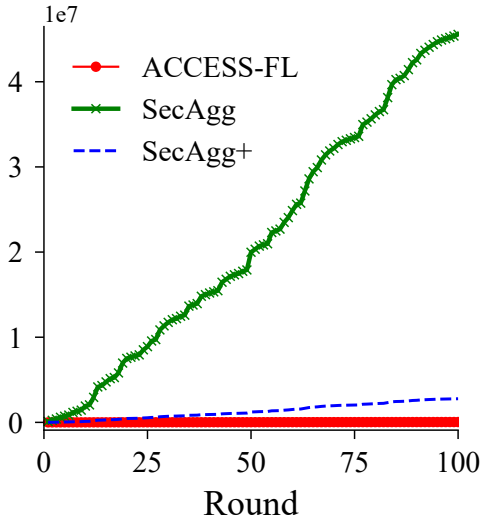

Figure 6.12: Server to clients.

Figure 6.13: Accumulated running time on server and client(ms) for MNIST experiments.

regardless of the number of participating clients. On the other hand, the number of messages sent by each client in SecAgg for the network size of 100 clients is around 10,000 messages by the 100th round. The number of messages sent by each client in SecAgg+ decreases compared to SecAgg, approximately 900 messages by the 100th round. The number of messages sent by the server remains constant at 2 messages per round, except for the initial round, where an additional message is sent to broadcast the public keys. However, in SecAgg and SecAgg+, at each round, the server sends around 100 messages per round that include broadcasting two public keys per client, sending encrypted values received by every client, and broadcasting the participants in sending masked model and the new global model. Although the number of messages sent by the server is equal in both SecAgg and SecAgg+, the sizes of messages are considerably different. That is, in SecAgg, the server sends cipher texts to every client, which includes the encrypted value generated from every other client. However, in SecAgg+, the size of each encrypted-value message equals $log(|C|)$ as each client only pairs with its $log(|C|)$ neighbors.

The constant communication cost for each client in ACCESS-FL is attributed to the protocol's design, which generates shared secrets between only two clients, regardless

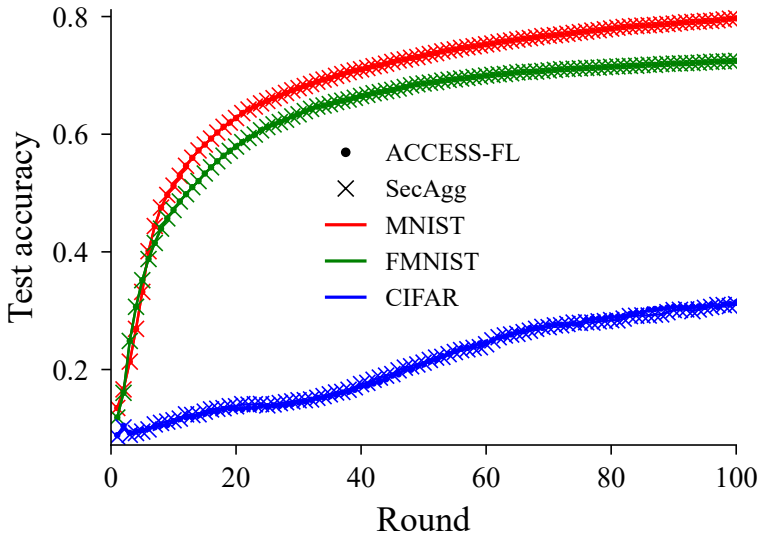

Figure 6.14: Learning curve comparison between ACCESS-FL and FedAvg in MNIST, FMNIST, and CIFAR.

of the total number of participants. By limiting the number of pairwise shared secrets, ACCESS-FL significantly reduces the communication overhead for each client compared to SecAgg and SecAgg+. In ACCESS-FL, clients themselves **find their pairs and change them in every round without the server's knowledge.** The only message that each client sends at each round to the server is a masked model update (except for the initial round, where each client needs to generate one key pair and send its public key to the server). Hence, the number of transmitted messages from clients and the overall load on the network in ACCESS-FL is comparable to the number of messages that each client sends in traditional FL. Thus, ACCESS-FL improves the privacy of a honest-but-curious stable FL system with approximately the same load on clients that participated in a traditional FL. Additionally, the size of the masked model update remains constant in different network sizes, as the model architecture and the masking scheme do not change based on the number of participants. Furthermore, the number of messages sent from the server at each round is only twice compared to the traditional FL (except for broadcasting the received public keys at the first round). This message includes the new global model and the list of participants where the latter does not need any cryptographic operation. Thus, ACCESS-FL, in large-scale FL stable networks, makes the overhead on the server approximately equal to the server's overhead in traditional FL while prevents the server from applying a model inversion attack by

concealing the trained model from a honest-but-curious server. In contrast, SecAgg and SecAgg+ require significantly higher communication costs due to the number of pairs that each client has and applying double masking along with the need for encryption to share secrets via server. In SecAgg, each client needs to generate shared secrets with every other client, which leads to an increase in the number of messages and message size for each client. SecAgg+, despite reducing the communication cost compared to SecAgg by having the clients pairing with $log(|C|)$ neighbors, still employs a double masking technique, and the clients are required to generate two key pairs and perform cryptographic operations to compute encrypted values for their neighbors. Despite this improvement, the server in SecAgg+ still knows who is paired with whom, and the communication cost for each client and the server grows with the number of clients. Furthermore, SecAgg+ requires message exchanges between the server and clients to handle the unmasking of the global model, even in stable FL environments with limited client dropouts and low delay variations. However, in ACCESS-FL, the server is not responsible for unmasking the aggregated masked models, and the protocol handles client dropout or incidents of delayed messages by making the clients find new pairs and resend their masked models. Which is efficient in large-scale stable FL systems such as healthcare systems where the privacy of data is crucial, the delay variation in the network is low, and the clients who participate in FL have reliable deployed devices, with a rare client dropout rate. Additionally, the server in such networks is honest-but-curious; However, it is required to prevent the server from accessing the critical clinical data of patients. Applying SecAgg and SecAgg+ for such networks makes unnecessary overload on the network, clients, and server only for privacy-preserving.

Figure 6.12 illustrates the accumulated running time for the server in ACCESS-FL, SecAgg, and SecAgg+ over the 100 training rounds. In ACCESS-FL, the server's running time at each FL round is constant (except for the initial round, where the server broadcasts the public keys to the network) and approximately reaches accumulatively 30 seconds by the end of the 100 rounds. This computation cost consists of the server aggregating the masked model updates without being responsible for handling client dropout or delayed messages, in contrast to SecAgg and SecAgg+, the server is required to unmask the aggregated model and manage the client dropout. Upon

receiving masked models from all participating clients, the server aggregates these updates, and generates the new global model in each round. As the number of clients remains constant (100 in our experiments), the server's running time per round also remains relatively constant. On the other hand, the server's accumulated running time in SecAgg increases quadratically, exceeding 445,000 seconds by the 100th round and approximately 2 minutes per round. SecAgg+ reduces the server's running time compared to SecAgg, summing up to around 3000 seconds by the 100th round. The high computation costs on the server side in SecAgg and SecAgg+ are the result of several factors, such as the need to perform cryptographic operations to reconstruct the shared masks for dropped-out clients and self-masks for participants (by using Shamir's secret sharing to generate the random elements of participants and private key of dropped-out clients, then running PRG function on the calculated elements to reconstruct the masks) that leads to an increased complexity of the aggregation process. Although SecAgg+ introduces an additional computation cost for the server to generate the random graph, the overall server cost is lower than SecAgg because the number of pairs per client is reduced from $|C| - 1$ in SecAgg (where each client is peered with every other client) to $log(|C|)$ in SecAgg+.

Figure 6.11 shows the accumulated running time for clients in ACCESS-FL, SecAgg, and SecAgg+ over the 100 training rounds. In ACCESS-FL, the running time for each client remains low and does not increase with the number of participating clients, staying at approximately 0.4 seconds throughout the 100 rounds. The constant computation cost for each client demonstrates that the protocol remains suitable for large-scale stable networks. In ACCESS-FL, each client runs a deterministic function to find its two pairs, performs local model training, and applies the masking process to the trained model (by using the shared secrets generated with only two pairs and running PRG function on the shared secrets to generate the shared masks), then it sends the masked model update to the server. In case of client dropout or delayed message, the server sends the participants within the same training round, then clients find new pairs and mask their trained models with the new shared masks. Thus, ACCESS-FL only applies dropout mitigation techniques or handles delayed updates in ACCESS-FL only when necessary. In contrast, the running time for each client in SecAgg is about 10 seconds by the 100th round. SecAgg+ reduces the running time for each client compared to SecAgg, but it still increases with the number of clients (approximately 2 seconds by

Table 6.2: Total number of messages sent from clients for scenarios with **node dropout (D) and without node dropout (ND)**.

Round	ACCESS-FL		SecAgg		SecAgg+		FedAvg
	ND	D	ND	D	ND	D	ND
10	1100	1099	102000	111702	9000	9594	1000
30	3100	3067	306000	4312520	27000	379764	3000
50	5100	4995	510000	12592570	45000	1109910	5000
70	7100	6883	714000	24951868	63000	2200032	7000
100	10100	9640	1020000	51139440	90000	4510170	10000

the 100th round. The computation cost for each client in SecAgg and SecAgg+ is a consequence of their more complex masking processes, which involve creating two key pairs, generating shared secrets (with every other client in SecAgg and the $log(|C|)$ neighbors in SecAgg+), performing double masking to its trained model, running PRG function on the random element to generate the self-mask, running $|C| - 1$ times of PRG function in SecAgg and $log|C|$ in SecAgg+ on shared secrets to creating shared masks, engaging in cryptographic operations that include splitting their private key and random element into $|C|$ parts in SecAgg and $|log(|C|)$ parts in SecAgg+, then it encrypts these values with its pair public key and sends these cipher texts to the server. After the server receives the masked models and sends the participants list, each client needs to decrypt the cipher texts of its pairs. Then, it sends the portion of the random element of its pairs if they are claimed as a participant by the server or the portion of the private key of its peers if they are recognized as the dropped-out clients by the server. Lastly, Figure 6.14 shows the comparison of learning curves between ACCESS-FL and FedAvg on the MNIST, FMNIST, and CIFAR datasets. When ACCESS-FL is applied, the results between ACCESS-FL and FedAvg on MNIST and FMNIST are exactly the same, and the accuracy difference on CIFAR is less than 1% in each training round.

6.5.2 Client Dropout for ACCESS-FL, SecAgg, and SecAgg+

In this section, we compare the communication costs of ACCESS-FL, SecAgg, and SecAgg+ under scenarios with and without client dropout. We evaluate the number of messages sent from clients and servers, as well as the size of these messages over 100

Table 6.3: Total number of messages sent from the server for scenarios with **node dropout (D) and without node dropout (ND)**.

Round	ACCESS-FL		SecAgg		SecAgg+		FedAvg
	ND	D	ND	D	ND	D	ND
10	12	13	1030	1228	1030	2218	11
30	32	35	3090	43848	3090	46788	31
50	52	57	5150	127660	5150	132510	51
70	72	79	7210	252664	7210	259384	71
100	102	112	10300	517405	10300	526855	101

Table 6.4: Total size of messages sent from the server (MB) for scenarios with **node dropout (D) and without node dropout (ND)**.

Round	ACCESS-FL		SecAgg		SecAgg+		FedAvg
	ND	D	ND	D	ND	D	ND
10	8.58	8.58	274.620	274.639	16.722	16.838	8.560
30	24.143	24.144	823.859	11534.09	50.165	702.659	24.123
50	39.707	39.708	1373.1	33778.32	83.609	2057.346	39.686
70	55.270	55.272	1922.338	67007.34	117.052	4080.9	55.25
100	78.615	78.619	2746.197	137447.4	167.218	8370.354	78.594

rounds on the MNIST dataset. Table 6.2 presents the comparison of the number of messages sent from clients. In the ACCESS-FL, the number of messages sent from clients in the dropout scenario slightly decreases compared to the stable scenario, with the reduction becoming more pronounced as the rounds progress. For instance, at the 100th round, the number of messages drops from 10,100 to 9,640 due to client dropouts. This is because, in each dropout scenario, the remaining clients compensate by sending additional portions of the dropout client's shares to the server. In contrast, SecAgg shows a significant increase in the number of messages in dropout scenarios due to the overhead of handling client dropouts and reconstructing shared secrets. SecAgg+ also shows an increase, though it is less severe compared to SecAgg, due to its more efficient handling of client pairs in a k-regular graph structure. Table 6.3 illustrates the number of messages sent from the server. ACCESS-FL demonstrates a constant and minimal increase in the number of server messages, as it maintains a steady communication pattern irrespective of client dropouts. In SecAgg and SecAgg+, the server's messaging overhead significantly increases in the presence of dropouts, reflecting the additional communication required to manage shared secret reconstructions and handle the redistribution of keys and masked values. Table 6.4 provides the size of messages

sent from the server. ACCESS-FL maintains a smaller message size, around 78.61 MB at the 100th round, even in the dropout scenario. This reveals the protocol's efficiency in managing communication overhead to handle client dropouts. In contrast, SecAgg's message size reaches 137.45 GB due to the intensive cryptographic operations required to manage dropouts and double masking. Although SecAgg+ reduces this overhead, it grows substantially in message sizes and reaches approximately 8.37 GB by the 100th round. The comparative analysis of ACCESS-FL with SecAgg and SecAgg+ demonstrates that ACCESS-FL significantly reduces both the number and size of messages exchanged between clients and the server. This reduction is achieved by eliminating unnecessary cryptographic operations and having shared secrets only between two other peers per client. ACCESS-FL is particularly effective in stable federated learning environments with limited client dropout rates and low network frequencies.

6.6 Discussion and Future Work

ACCESS-FL is optimized for large-scale, stable FL environments where node dropout is limited and network delays are low. Practical implementations of such environments include fraud detection for financial applications [145], privacy-preserving systems against money laundry by IBM[146], and AI applications in healthcare systems [147]. These applications could benefit from reduced communication and computation overhead, which makes ACCESS-FL a practical choice for privacy-sensitive domains. Future work could involve extending ACCESS-FL to handle active adversaries. Additionally, the integration of differential privacy techniques with ACCESS-FL could further enhance the privacy guarantees of the protocol. However, a limitation of ACCESS-FL is its performance when node dropout or delayed messages occur frequently, as this can lead to loop vulnerability where clients are stuck within a training round while finding new pairs.

6.7 Summary

In this chapter, we proposed ACCESS-FL, an efficient, secure aggregation protocol designed for honest-but-curious scenarios in a stable FL environment with limited client dropout and low network delay variations. ACCESS-FL addresses the high communication and computation costs associated with Google's SecAgg protocol and SecAgg+ while maintaining the same level of security against model inversion attacks. ACCESS-FL generates shared secrets between only two clients, regardless of the number of clients, which reduces the computational complexity to a constant level and makes the communication cost for each client $O(1)$. Our protocol eliminates the need for double-masking, cryptographic computations, and self-masks by having only shared masks which cancel out each other during the aggregation process without server intervention. This approach significantly reduces the computational and communication burden on both clients and servers. ACCESS-FL handles client dropouts or delayed updates by having participating clients generate new shared masks with new peers and resend their masked models, which ensures the server is not required to manage the removal of masks from dropped-out clients. We conducted experiments on the MNIST dataset to evaluate the performance of ACCESS-FL compared to SecAgg and SecAgg+. The evaluation results demonstrated that ACCESS-FL significantly reduces communication and computational costs. The accumulated message size and number of messages exchanged between the server and clients remained constant in ACCESS-FL, whereas they increased with the number of clients in SecAgg and SecAgg+. Furthermore, the running time for the server and clients in ACCESS-FL was substantially lower than in SecAgg and SecAgg+.

Chapter 7

Conclusions and Future Directions

The rapid development of technology and the increasing reliance on online services have transformed the ways in which society interacts, communicates, and accesses information. The Internet, now a critical infrastructure, supports essential services ranging from commerce and education to healthcare and government functions. However, as the Internet evolves, the growing centralization of control and resources presents increasing challenges for the security of online services. This thesis has investigated the impact of Internet centralization on the security of online services, with a particular focus on two key security aspects: availability and data privacy. Through analyses, this research has investigated how centralization can lead to risks and vulnerabilities. The thesis explored the digital divide in the context of Australian government services, particularly emphasizing the disparities faced by indigenous communities in terms of access to critical online services. Chapter 4 revealed how the centralization of DNS providers can exacerbate these disparities, leading to a digital divide compared to the general population. This chapter underscored the need for diversification of DNS providers to ensure more equitable and resilient access to online services. Building on this, Chapter 5 expanded the investigation to include indirect DNS dependencies. By examining the layered dependencies between DNS providers, the research revealed hidden vulnerabilities that arise when a small number of DNS providers control critical infrastructure. The study introduced three distinct attacker models based on the attacker's resources, knowledge, and intent, illustrating how the centralization

of DNS services makes Australian government domains—especially those serving indigenous populations—more susceptible to specific types of cyber-attacks. Moreover, the geographical distribution of DNS providers added another layer of vulnerability, as reliance on foreign-based DNS servers further implies the risk of disruption and reduces the resilience of these services. Through the analysis of DNS dependencies in both population groups, it became evident that the reliance on a small number of DNS providers not only contributes to the digital divide but also increases the vulnerability of these communities to service outages and cyber-attacks. The chapter concluded that mitigating these risks would require strategic diversification of DNS providers, particularly for services catering to vulnerable populations. In Chapter 6, the thesis investigates data privacy issues in federated learning systems, where the centralization of model aggregation can lead to privacy breaches. The chapter proposed ACCESS-FL, an Agile Communication and Computation for Efficient Secure Aggregation in Stable Federated Learning Networks, such as data centers or hospitals. The original protocol, Google's Secure Aggregation protocol, has the complexity order of $o(n^2)$ while ACCESS-FL reduced the cost to $o(n)$. ACCESS-FL ensures that clients only generate a shared mask with two other clients, unlike Google's Secure Aggregation, where each client needs to generate a shared mask with every other client.

Although both DNS resolution and FL are decentralized in principle— *DNS queries are resolved across multiple authoritative name servers, and FL trains models locally on each client's device without requiring raw data to be centralized—real-world deployments often end up depending on a small set of providers or aggregators.* Consequently, single points of failure and privacy risks may still arise. In this thesis, two such services are explored in depth, including DNS-based hosting for Australian government domains and FL-based model aggregation, revealing how partial centralization leads to security vulnerability and privacy risk. The findings of this thesis revealed the need for an understanding of how Internet centralization impacts the security, availability, and privacy of online services. The research has shown that centralization implies significant vulnerabilities, particularly for disadvantaged communities and in systems where privacy is paramount. By proposing solutions such as DNS provider diversification and enhanced secure aggregation protocols, this thesis provides insights for policymakers and technologists aiming to create a more secure and equitable Internet.

7.1 Implications:

The implications of this thesis are as follows:

7.1.1 Digital Divide:

The centralization of DNS services for Australian government websites has created disparities in service availability, particularly for indigenous communities. Chapter 4 exposed how centralized control over DNS providers can contribute to the digital divide, affecting access to critical services.

7.1.2 Indirect Dependencies:

Chapter 5 investigated indirect DNS dependencies and IP geolocation of DNS servers to imply the vulnerability of indigenous and general Australian domains against different cyber-attackers.

7.1.3 Data Privacy in Federated Learning:

ACCESS-FL protocol proposed in Chapter 6 demonstrated a significant reduction in communication and computation costs compared to Google's Secure Aggregation protocol and the SecAgg+ (the improved version of SecAgg proposed by Google) while preserving privacy against model inversion attack where honest-but-curious central server tries to infer sensitive client's information. ACCESS-FL is suitable for stable federated learning systems such as healthcare organizations.

7.2 Limitations and Future Research Directions

Although this thesis addressed several aspects of DNS centralization and secure federated learning, there are limitations that must be acknowledged and further investigation could explore a broader range of adversarial models, security enhancements, and practical use cases. In particular:

1. **Holistic Digital Divide Assessment:** Beyond the DNS layer alone, further exploration might consider factors such as poor connectivity, device affordability, or local hosting capacity. By incorporating these broader infrastructure variables, future research could investigate the correlation between DNS provider centralization and the socio-technical impact on indigenous communities.
2. **Quantitative Risk Modeling:** In this thesis, we investigated indirect DNS dependencies and performed IP geolocation of DNS servers. Building on these analysis, we introduced security risks associated with DNS centralization using qualitative attacker models to illustrate how these dependencies might be exploited under different threat scenarios. More refined quantitative frameworks (e.g., game-theoretic [153] approaches or cost-benefit analysis [154]) would systematically evaluate the impact of different diversification or secure aggregation strategies. For instance, adopting federated anomaly detection in DNS servers would clarify how policy decisions mitigate risk.
3. **Scaling to Multi-Operator DDoS Defense:** Future research might investigate a federated learning system for cooperative DDoS detection among diverse DNS providers and ISPs where each has distinct traffic patterns. Such a study could reveal best practices for inter-organization collaboration in preventing widespread DNS outages.
4. **Adversarial Server and Colluding Clients:** While ACCESS-FL assumes an honest-but-curious aggregator and non-colluding clients, real-world threats may involve both a malicious central server *and* a subset of clients acting in collusion. Future work could explore protocols resilient to such active adversaries, ensuring that neither the server nor a colluding coalition of participants can extract sensitive information from honest clients' updates.

5. **Advanced Cryptographic Techniques:** Existing secure aggregation primarily relies on masking techniques and/or secret sharing. Integrating more robust cryptographic primitives, (such as zero-knowledge proofs [155], fully homomorphic encryption [156], or verifiable shuffles [157]) could improve the correctness, particularly in high-sensitive scenarios such as healthcare or governmental infrastructure.
6. **Differential Privacy Integration:** Although secure aggregation hides individual updates from the server, a determined adversary might still obtain information. Combining secure aggregation with differential privacy methods could provide mathematically higher limits on the information that can be leaked from aggregated model parameters.
7. **Fault Tolerance and Dynamic Client Participation:** ACCESS-FL focuses on stable networks with low dropout rates. To extend the protocol's usability, a valuable direction is tolerating higher client drop-out rates or frequent join/leave events at scale without incurring excessive overhead.

The aforementioned directions reflect the ongoing need to strengthen both availability and data privacy in an Internet infrastructure that is increasingly centralized in practice, even if originally designed to be decentralized. By addressing malicious participants, malicious aggregators, scalability concerns, differential privacy, and broader socio-technical factors, future work can further improve the resilience and inclusivity of online services.

Bibliography

[1] **Niousha Nazemi**, O. Tavallaie, A. Y. Zomaya, and R. Holz, "DNS Dependencies as an Expression of the Digital Divide: the Example of Australia," in *European Symposium on Research in Computer Security*. Springer, 2023, pp. 496–509.

[2] R. Holz, **Niousha Nazemi**, O. Tavallaie, and A. Y. Zomaya, "Evidence for a digital divide? Measuring DNS dependencies in the context of the indigenous population of Australia," in *IETF/IAB Workshop on Barriers to Internet Access of Services (biasws)*, 2023, available at: https://datatracker.ietf.org/doc/slides-biasws-evidence-for-a-digital-divide-measuring-dns-dependencies-in-the-context-of-the-indigenous-population-of-australia/.

[3] **Niousha Nazemi**, O. Tavallaie, A. M. Mandalari, H. Haddadi, R. Holz, and A. Y. Zomaya, "Analysis of DNS Dependencies and their Security Implications in Australia: A Comparative Study of General and Indigenous Populations," 2024, Submitted to IEEE Journal of Transactions on Network and Service Management.

[4] **Niousha Nazemi**, O. Tavallaie, S. Chen, A. Y. Zomaya, and R. Holz, "Boosting Communication Efficiency of Federated Learning's Secure Aggregation," in *2024 54th Annual IEEE/IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S)*. IEEE, 2024, pp. 157–158.

[5] **Niousha Nazemi**, O. Tavallaie, S. Chen, A. M. Mandalari, K. Thilakarathna, R. Holz, H. Haddadi, and A. Y. Zomaya, "ACCESS-FL: Agile Communication and Computation for Efficient Secure Aggregation in Stable Federated Learning

Networks," Submitted to IEEE International Conference on Web Services, 2025. [Online]. Available: https://arxiv.org/abs/2409.01722

[6] S. Chen, O. Tavallaie, **Niousha Nazemi**, and A. Y. Zomaya, "RBLA: Rank-Based-LoRA-Aggregation for Fine-Tuning Heterogeneous Models in FLaaS," in *International Conference on Web Services*. Springer, 2024, pp. 47–62.

[7] S. Chen, O. Tavallaie, **Niousha Nazemi**, X. Chen, and A. Y. Zomaya, "AutoRank: MCDA Based Rank Personalization for LoRA-Enabled Distributed Learning," *arXiv preprint arXiv:2412.15553*, 2024.

[8] S. Harishkumar and R. Bhuvaneswaran, "Unveiling domain generation algorithms in dns log traffic: A next-generation intelligent framework for dynamic anomaly detection and mitigation through machine learning analysis," in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. IEEE, 2024, pp. 1–7.

[9] P. Mockapetris, "Domain names - Concepts and facilities," Internet Engineering Task Force, Request for Comments 1034, 1987. [Online]. Available: https://www.rfc-editor.org/rfc/rfc1034

[10] J. F. Kurose, *Computer Networking: A Top-Down Approach Featuring the Internet, 3/E*. Pearson Education India, 2005.

[11] P. Mockapetris, "Domain Names - Implementation and Specification," Internet Engineering Task Force, Request for Comments 1035, 1987. [Online]. Available: https://www.rfc-editor.org/rfc/rfc1035

[12] I. A. N. Authority, "Root Server Information," https://www.iana.org/domains/root/servers, 2023, accessed on Jun 1, 2023.

[13] D. Katabi and J. Wroclawski, "A Framework for Scalable Global IP-Anycast (GIA)," *ACM SIGCOMM Computer Communication Review*, vol. 30, no. 4, pp. 3–15, 2000.

[14] .au Domain Administration Ltd (auDA), "Welcome to auDA," https://www.auda.org.au/, accessed on September 01, 2024.

[15] J. Abbate, *Inventing the internet*. MIT press, 2000.

[16] J.-C. Plantin, C. Lagoze, P. N. Edwards, and C. Sandvig, "Infrastructure Studies Meet Platform Studies in the Age of Google and Facebook," *New Media & Society*, vol. 20, no. 1, pp. 293–310, 2018.

[17] C. Hesselman, P. Grosso, R. Holz, F. Kuipers, J. H. Xue, M. Jonker, J. de Ruiter, A. Sperotto, R. van Rijswijk-Deij, G. C. M. Moura, A. Pras, and C. de Laat, "A Responsible Internet to increase trust in the digital world," *Journal Network and Systems Management*, vol. 28, no. 4, oct 2020.

[18] J. Arkko, "Centralised Architectures in Internet Infrastructure," Internet Engineering Task Force, Internet-Draft draft-arkko-arch-infrastructure-centralisation-00, nov 2019. [Online]. Available: https://datatracker.ietf.org/doc/draft-arkko-arch-infrastructure-centralisation/00/

[19] J. Arkko, B. Trammell, M. Nottingham, C. Huitema, M. Thomson, J. Tantsura, and N. ten Oever, "Considerations on Internet Consolidation and the Internet Architecture," IETF, Internet Draft, 2019. [Online]. Available: https://tools.ietf.org/html/draft-arkko-iab-internet-consolidation-02

[20] "Global Internet Report 2019: Consolidation in the Internet Economy," https://www.internetsociety.org/resources/doc/2019/global-internet-report-2019, Internet Society, 2023, accessed on Jun 13, 2023.

[21] The Register, "AWS DNS DDoS attack overwhelmed its servers for hours," https://www.theregister.com/2019/10/22/aws_dns_ddos, October 2019, accessed on April 7, 2023.

[22] G. C. M. Moura, J. Heidemann, M. Müller, R. de O. Schmidt, and M. Davids, "When the Dike Breaks: Dissecting DNS Defenses During DDoS," in *Proceedings of the ACM Internet Measurement Conference*, 2020, pp. 8–21.

[23] Q. Scheitle, O. Hohlfeld, J. Gamba, J. Jelten, T. Zimmermann, S. D. Strowes, and N. Vallina-Rodriguez, "A Long Way to the Top: Significance, Structure, and Stability of Internet Top Lists," in *Proceedings of the Internet Measurement Conference 2018*, 2018, pp. 478–493.

[24] C. Lu, B. Liu, C. Lu, Z. Li, H. Duan, Y. Liu, Z. Zhang, S. Hao, M. Hao, and F. Li, "An End-to-End, Large-Scale Measurement of DNS-over-Encryption: How Far

Have We Come?" in *Proceedings of the Internet Measurement Conference*, 2019, pp. 22–35.

[25] U.S. Department of Commerce, "Falling through the Net: A Survey of the" Have Nots" in Rural and Urban America," July 1995.

[26] ——, "Falling Through the Net II: New Data on the Digital Divide," July 1998.

[27] ——, "Falling Through the Net: Defining the Digital Divide," July 1999.

[28] J. Van Dijk, *The Digital Divide*. John Wiley & Sons, 2020.

[29] E. M. Rogers, "The Digital Divide," *Convergence*, vol. 7, no. 4, pp. 96–111, 2001.

[30] Y. Eshet-Alkalai, "Digital Literacy: A Conceptual Framework for Survival Skills in the Digital Era," *Journal of Educational Multimedia and Hypermedia*, vol. 13, no. 1, pp. 93–106, 2004.

[31] P. Norris, *Digital Divide: Civic Engagement, Information Poverty, and the Internet Worldwide*. Cambridge University Press, 2001.

[32] C. Fuchs and E. Horak, "Africa and the Digital Divide," *Telematics and Informatics*, vol. 25, no. 2, pp. 99–116, 2008.

[33] K. Samaras, "Indigenous Australians and the 'Digital Divide'," *International Journal of Libraries and Information Studies*, vol. 55, no. 2-3, pp. 84–95, 2005.

[34] M. Warschauer, *Technology and Social Inclusion: Rethinking the Digital Divide*. MIT Press, 2004.

[35] D. Levin and S. Arafeh, "The Digital Disconnect: The Widening Gap Between Internet-Savvy Students and Their Schools," 2002.

[36] E. Beaunoyer, S. Dupéré, and M. J. Guitton, "COVID-19 and Digital Inequalities: Reciprocal Impacts and Mitigation Strategies," *Computers in Human Behavior*, vol. 111, p. 106424, 2020.

[37] A. Dongare, R. Kharde, A. D. Kachare *et al.*, "Introduction to Artificial Neural Network," *International Journal of Engineering and Innovative Technology (IJEIT)*, vol. 2, no. 1, pp. 189–194, 2012.

[38] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[39] S. J. Prince, *Computer Vision: Models, Learning, and Inference*. Cambridge University Press, 2012.

[40] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural Language Processing (Almost) from Scratch," *Journal of Machine Learning Research*, vol. 12, pp. 2493–2537, 2011.

[41] Y. Xiao, Z. Tian, J. Yu, Y. Zhang, S. Liu, S. Du, and X. Lan, "A Review of Object Detection Based on Deep Learning," *Multimedia Tools and Applications*, vol. 79, pp. 23 729–23 791, 2020.

[42] T. Iqbal and S. Qureshi, "The Survey: Text Generation Models in Deep Learning," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 6, pp. 2515–2528, 2022.

[43] S. Khare and M. Totaro, "Big Data in IoT," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, 2019, pp. 1–7.

[44] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated Learning in Mobile Edge Networks: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[45] R. Shokri and V. Shmatikov, "Privacy-Preserving Deep Learning," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 1310–1321.

[46] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Artificial Intelligence and Statistics*. PMLR, 2017, pp. 1273–1282.

[47] M. Fredrikson, S. Jha, and T. Ristenpart, "Model Inversion Attacks That Exploit Confidence Information and Basic Countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 1322–1333.

[48] L. Kissner and D. Song, "Privacy-preserving set operations," in *Annual International Cryptology Conference*. Springer, 2005, pp. 241–257.

[49] L. E. Olson, M. J. Rosulek, and M. Winslett, "Harvesting credentials in trust negotiation as an honest-but-curious adversary," in *Proceedings of the 2007 ACM workshop on Privacy in electronic society*, 2007, pp. 64–67.

[50] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical Secure Aggregation for Privacy-Preserving Machine Learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1175–1191.

[51] R. Cramer, I. B. Damgård *et al.*, *Secure multiparty computation*. Cambridge University Press, 2015.

[52] W. Diffie and M. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[53] E. Dawson and D. Donovan, "The Breadth of Shamir's Secret-Sharing Scheme," *Computers & Security*, vol. 13, no. 1, pp. 69–78, 1994.

[54] L.-J. Pang and Y.-M. Wang, "A new (t, n) multi-secret sharing scheme based on shamir's secret sharing," *Applied Mathematics and Computation*, vol. 167, no. 2, pp. 840–848, 2005.

[55] H. C. A. Van Tilborg, *Fundamentals of Cryptology: A Professional Reference and Interactive Tutorial*. Springer Science & Business Media, 2013, vol. 528.

[56] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1996.

[57] N. Koblitz, "A Course in Number Theory and Cryptography," *Graduate Texts in Mathematics*, vol. 114, 1987.

[58] ——, "Elliptic Curve Cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.

[59] V. S. Miller, "Use of Elliptic Curves in Cryptography," *Conference on the Theory and Application of Cryptographic Techniques*, pp. 417–426, 1985.

[60] I. F. Blake, G. Seroussi, N. P. Smart *et al.*, *Elliptic Curves in Cryptography*. Cambridge University Press, 1999, vol. 265.

[61] T. Dierks and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," *RFC 5246, August*, 2008.

[62] C. Kaufman, "Internet Key Exchange (IKEv2) Protocol," *RFC 4306, December*, 2005.

[63] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-Hellman Key Distribution Extended to Group Communication," in *Proceedings of the 3rd ACM Conference on Computer and Communications Security*, 1996, pp. 31–37.

[64] M. Burmester and Y. G. Desmedt, "A Secure and Efficient Conference Key Distribution System," in *Workshop on the Theory and Application of Cryptographic Techniques*. Springer, 1994, pp. 275–286.

[65] P. W. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," *SIAM Review*, vol. 41, no. 2, pp. 303–332, 1999.

[66] C. Peikert, "A Decade of Lattice Cryptography," *Foundations and Trends® in Theoretical Computer Science*, vol. 10, no. 4, pp. 283–424, 2016.

[67] D. Jao and L. De Feo, "Towards Quantum-Resistant Cryptosystems from Supersingular Elliptic Curve Isogenies," in *International Workshop on Post-Quantum Cryptography*. Springer, 2011, pp. 19–34.

[68] R. J. McEliece, "A Public-Key Cryptosystem Based on Algebraic Coding Theory," 1978.

[69] T. Fiebig, S. Gürses, C. H. Gañán, E. Kotkamp, F. Kuipers, M. Lindorfer, M. Prisse, and T. Sari, "Heads in the Clouds? Measuring Universities' Migration to Public Clouds: Implications for Privacy & Academic Freedom," in *Proceedings on Privacy Enhancing Technologies Symposium*, vol. 2023, no. 2, 2022.

[70] L. Zembruzki, R. Sommese, L. Z. Granville, A. S. Jacobs, M. Jonker, and G. C. Moura, "Hosting industry centralization and consolidation," in *NOMS 2022-2022 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2022, pp. 1–9.

[71] A. Kashaf, V. Sekar, and Y. Agarwal, "Analyzing third party service dependencies in modern web services: Have we learned from the mirai-dyn incident?" in *Proceedings of the ACM Internet Measurement Conference*, 2020, pp. 634–647.

[72] "Alexa - Top Sites," https://www.alexa.com, 2023, accessed on May 30, 2023.

[73] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and Other Botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[74] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage, "Inferring Internet Denial-of-Service Activity," *ACM Transactions on Computer Systems*, vol. 24, no. 2, pp. 115–139, 2006.

[75] M. Jonker, A. King, J. Krupp, C. Rossow, A. Sperotto, and A. Dainotti, "Millions of Targets under Attack: A Macroscopic Characterization of the DoS Ecosystem," in *Proceedings of the 2017 Internet Measurement Conference (IMC '17)*. ACM, 2017, pp. 100–113.

[76] M. Ikram, R. Masood, G. Tyson, M. A. Kaafar, N. Loizon, and R. Ensafi, "The chain of implicit trust: An analysis of the web third-party resources loading," in *The World Wide Web Conference*, 2019, pp. 2851–2857.

[77] ExpiredDomains.net, "Alexa Top Websites," 2024, Accessed: 2024-03-01. [Online]. Available: https://www.expireddomains.net/alexa-top-websites/

[78] T. Urban, M. Degeling, T. Holz, and N. Pohlmann, "Beyond the front page: Measuring third party dynamics in the field," in *Proceedings of The Web Conference 2020*, 2020, pp. 1275–1286.

[79] A. Kashaf, J. Dou, M. Belova, M. Apostolaki, Y. Agarwal, and V. Sekar, "A First Look at Third-Party Service Dependencies of Web Services in Africa," in *Passive and Active Measurement: 24th International Conference, PAM 2023, Virtual Event, March 21–23, 2023, Proceedings*. Springer, 2023, pp. 595–622.

[80] R. Houser, S. Hao, C. Cotton, and H. Wang, "A Comprehensive, Longitudinal Study of Government DNS Deployment at Global Scale," in *2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 2022, pp. 193–204.

[81] M. Candela, V. Luconi, and A. Vecchio, "Impact of the COVID-19 Pandemic on the Internet Latency: A Large-Scale Study," *Computer Networks*, vol. 182, p. 107495, 2020.

[82] T. Böttger, G. Ibrahim, and B. Vallis, "How the internet reacted to covid-19: A perspective from facebook's edge network," in *Proceedings of the ACM Internet Measurement Conference*, 2020, pp. 34–41.

[83] T. Favale, F. Soro, M. Trevisan, I. Drago, and M. Mellia, "Campus traffic and e-learning during covid-19 pandemic," *Computer networks*, vol. 176, p. 107290, 2020.

[84] S. Wang, A. Cao, G. Wang, and Y. Xiao, "The Impact of Energy Poverty on the Digital Divide: The Mediating Effect of Depression and Internet Perception," *Technology in Society*, vol. 68, p. 101884, 2022.

[85] G. Singleton, M. F. Rola-Rubzen, K. Muir, D. Muir, and M. McGregor, "Youth Empowerment and Information and Communication Technologies: A Case Study of a Remote Australian Aboriginal Community," *GeoJournal*, vol. 74, pp. 403–413, 2009.

[86] C. Intahchomphoo, "Indigenous Peoples, Social Media, and the Digital Divide: A Systematic Literature Review," *American Indian Culture and Research Journal*, vol. 42, no. 4, pp. 85–111, 2018.

[87] J. Wu and W. Zhang, "On the Security of Verifiable and Oblivious Secure Aggregation for Privacy-Preserving Federated Learning," *IEEE Transactions on Dependable and Secure Computing*, 2024.

[88] J. Bouamama, Y. Benkaouz, and M. Ouzzif, "EdgeSA: Secure Aggregation for Privacy-Preserving Federated Learning in Edge Computing," in *2023 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and*

Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech). IEEE, 2023, pp. 0375–0382.

[89] M. Georgieva Belorgey, S. Dandjee, N. Gama, D. Jetchev, and D. Mikushin, "Falkor: Federated Learning Secure Aggregation Powered by AESCTR GPU Implementation," in *Proceedings of the 11th Workshop on Encrypted Computing & Applied Homomorphic Cryptography*, 2023, pp. 11–22.

[90] S. Gupta, A. Kapoor, and D. Kumar, "A Resource Adaptive Secure Aggregation Protocol for Federated Learning Based Urban Sensing Systems," in *Proceedings of the 6th Joint International Conference on Data Science & Management of Data (10th ACM IKDD CODS and 28th COMAD)*, 2023, pp. 135–135.

[91] S. Jegadeesan, M. Navaneetha, P. Poovizhi, S. Pavithra, and P. Santhiya, "Blockchain-Based Lightweight and Secure Aggregation Scheme for Smart Farming," in *2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*. IEEE, 2023, pp. 1266–1271.

[92] B. Pejó and G. Biczók, "Quality Inference in Federated Learning with Secure Aggregation," *IEEE Transactions on Big Data*, 2023.

[93] Y. Ma, J. Woods, S. Angel, A. Polychroniadou, and T. Rabin, "Flamingo: Multi-Round Single-Server Secure Aggregation with Applications to Private Federated Learning," *Cryptology ePrint Archive*, 2023.

[94] C. Perry, "Secure Aggregation in Federated Learning: Finding a More Communication-Computational Efficient Protocol," 2023.

[95] K. Wan, Y. Yao, H. Sun, M. Ji, and G. Caire, "GroupSecAgg: Information Theoretic Secure Aggregation with Uncoded Groupwise Keys," in *ICC 2023-IEEE International Conference on Communications*. IEEE, 2023, pp. 3890–3895.

[96] M. Mansouri, M. Önen, W. B. Jaballah, and M. Conti, "SoK: Secure Aggregation Based on Cryptographic Schemes for Federated Learning," in *PETS 2023, 23rd Privacy Enhancing Technologies Symposium*, vol. 2023, no. 1, 2023, pp. 140–157.

[97] H. Fereidooni, S. Marchal, M. Miettinen, A. Mirhoseini, H. Möllering, T. D. Nguyen, P. Rieger, A.-R. Sadeghi, T. Schneider, H. Yalame *et al.*, "SAFELearn: Secure Aggregation for Private Federated Learning," in *2021 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2021, pp. 56–62.

[98] M. Rathee, C. Shen, S. Wagh, and R. A. Popa, "Elsa: Secure Aggregation for Federated Learning with Malicious Actors," in *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 1961–1979.

[99] T. Haibo, L. Maonan, and R. Shuangyin, "ESE: Efficient Security Enhancement Method for the Secure Aggregation Protocol in Federated Learning," *Chinese Journal of Electronics*, vol. 32, no. 3, pp. 1–14, 2023.

[100] Z. Liu, J. Guo, K.-Y. Lam, and J. Zhao, "Efficient Dropout-Resilient Aggregation for Privacy-Preserving Machine Learning," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1839–1854, 2022.

[101] X. Wang, S. Garg, H. Lin, G. Kaddoum, J. Hu, and M. S. Hossain, "A Secure Data Aggregation Strategy in Edge Computing and Blockchain-Empowered Internet of Things," *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 14 237–14 246, 2020.

[102] X. Li, S. Liu, F. Wu, S. Kumari, and J. J. Rodrigues, "Privacy Preserving Data Aggregation Scheme for Mobile Edge Computing Assisted IoT Applications," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4755–4763, 2018.

[103] J. Wu, X. Sheng, G. Li, K. Yu, and J. Liu, "An Efficient and Secure Aggregation Encryption Scheme in Edge Computing," *China Communications*, vol. 19, no. 3, pp. 245–257, 2022.

[104] J. Wang, Z. Wang, and A. B. Abdallah, "Robust Client Selection Based Secure Collaborative Learning Algorithm for Pneumonia Detection," in *2023 IEEE 6th International Conference on Knowledge Innovation and Invention (ICKII)*. IEEE, 2023, pp. 614–619.

[105] "Services Australia," https://www.servicesaustralia.gov.au, Australian Government, 2023, accessed on Jun 1, 2023.

[106] "Australian Institute of Health and Welfare (AIHW)," https://www.aihw.gov.au/, accessed: 2024-04-01.

[107] Department of Social Services, Australian Government, "Program services for people with disability," 2024, accessed: 2024-04-02. [Online]. Available: https://www.dss.gov.au/our-responsibilities/disability-and-carers/program-services/for-people-with-disability

[108] **Nazemi, Niousha**, "Australian Government Domains for General and Indigenous Populations: NS Records and DNS Provider IPGeolocation," 2024. [Online]. Available: https://dx.doi.org/10.21227/4p6z-j284

[109] "Telstra Group Limited," https://en.wikipedia.org/wiki/Telstra, 2023, accessed on March 26, 2023.

[110] "CITEC," https://services.citec.com.au/about/, 2023, accessed on March 26, 2023.

[111] "Common Crawl," https://commoncrawl.org/, Common Crawl Foundation, 2023, accessed on August 17, 2023.

[112] "OONI Data," https://ooni.org/data/, Open Observatory of Network Interference, 2023, accessed on August 17, 2023.

[113] "OpenINTEL," https://openintel.nl, 2023, accessed on May 29, 2023.

[114] (2023) Certificate Transparency. https://certificate.transparency.dev/. Certificate Transparency. Accessed on August 17, 2023.

[115] "Top 10 Managed DNS Service Providers in the World Today," https://www.emergenresearch.com/blog/top-10-managed-dns-service-providers-in-the-world-today, accessed: 2024-01-12.

[116] Amazon Web Services, Inc., "Amazon route 53," 2024, accessed: 2024-03-31. [Online]. Available: https://aws.amazon.com/route53/

[117] Microsoft Corporation, "Azure dns," 2024, accessed: 2024-03-31. [Online]. Available: https://azure.microsoft.com/en-au/products/dns/

[118] Cloudflare, Inc., "Cloudflare dns," 2024, accessed: 2024-03-31. [Online]. Available: https://www.cloudflare.com/en-au/application-services/products/dns/

[119] Google LLC, "Google cloud dns," 2024, accessed: 2024-03-31. [Online]. Available: https://cloud.google.com/dns

[120] Akamai Technologies, "Akamai edge dns," 2024, accessed: 2024-03-31. [Online]. Available: https://www.akamai.com/products/edge-dns

[121] "Centralization, Decentralization, and Internet Standards," RFC 9518, Dec. 2023. [Online]. Available: https://www.rfc-editor.org/info/rfc9518

[122] CrowdStrike, "Remediation and guidance hub: Falcon content update for Windows hosts," https://www.crowdstrike.com/falcon-content-update-remediation-and-guidance-hub/, 25 July 2024.

[123] Cloudflare, "Understanding how Facebook disappeared from the Internet," https://blog.cloudflare.com/october-2021-facebook-outage, 4 October 2021.

[124] "Root Servers," https://root-servers.org, 2023, accessed on Jun 1, 2023.

[125] Internic, "Named Root File," https://www.internic.net/domain/named.root, Internic, 2023, accessed on Feb 22, 2023.

[126] Telstra, "Broadband internet, nbn, 5g, tv & mobile phone services," 2024, accessed: 2024-06-12. [Online]. Available: https://www.telstra.com.au/

[127] Optus. (2024) Optus. Accessed: 2024-04-08. [Online]. Available: https://www.optus.com.au

[128] Webcentral, "Australian domain names, hosting, website & emails," 2024, accessed: 2024-06-12. [Online]. Available: https://webcentral.au/

[129] "Queensland State Archives," https://www.archivessearch.qld.gov.au/agencies/A482, 2023, accessed on March 26, 2023.

[130] Parliament of Australia, "Optus network outage," 2024, accessed: 2024-06-12. [Online]. Available: https://www.aph.gov.au/Parliamentary_Business/Committees/Senate/Environment_and_Communications/OptusNetworkOutage

[131] G. C. Kessler, "Lone Operator Cyberterrorism from the Perspective of a Hacker," *Lone Actors–An Emerging Security Threat*, vol. 123, p. 45, 2015.

[132] ——, "Lone-Operator Cyberterrorism," *Journal of Information Warfare*, vol. 15, no. 1, pp. 15–28, 2016.

[133] J. Jing, P. Liu, D. Feng, J. Xiang, N. Gao, and J. Lin, "ARECA: a highly attack resilient certification authority," in *Proceedings of the 2003 ACM workshop on Survivable and self-regenerative systems: in association with 10th ACM Conference on Computer and Communications Security*, 2003, pp. 53–63.

[134] R. Girtler, "The fringe group of hackers—rebels, spies and thieves," *e & i Elektrotechnik und Informationstechnik*, vol. 120, pp. 216–219, 2003.

[135] S. R. Davies, "Characterizing hacking: Mundane engagement in US hacker and makerspaces," *Science, Technology, & Human Values*, vol. 43, no. 2, pp. 171–197, 2018.

[136] P. A. Taylor, "From hackers to hacktivists: speed bumps on the global superhighway?" *New Media & Society*, vol. 7, no. 5, pp. 625–646, 2005.

[137] J. Sheldon, "State of the art: Attackers and targets in cyberspace," *Journal of Military and Strategic Studies*, vol. 14, no. 2, 2012.

[138] A. Pawlicka, M. Choraś, and M. Pawlicki, "Cyberspace threats: not only hackers and criminals. Raising the awareness of selected unusual cyberspace actors-cybersecurity researchers' perspective," in *Proceedings of the 15th International Conference on Availability, Reliability and Security*, 2020, pp. 1–11.

[139] C. Williams, "Bezos DDoS'd: Amazon Web Services' DNS systems knackered by hours-long cyber-attack," *The Register*, 2019.

[140] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[141] L. Blum, M. Blum, and M. Shub, "A simple unpredictable pseudo-random number generator," *SIAM Journal on Computing*, vol. 15, no. 2, pp. 364–383, 1986.

[142] R. Impagliazzo, L. A. Levin, and M. Luby, "Pseudo-random generation from one-way functions," in *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, 1989, pp. 12–24.

[143] B. Miller, A. Kantchelian, S. Afroz, R. Bachwani, E. Dauber, L. Huang, M. C. Tschantz, A. D. Joseph, and J. D. Tygar, "Adversarial active learning," in *Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop*, 2014, pp. 3–14.

[144] J. H. Bell, K. A. Bonawitz, A. Gascón, T. Lepoint, and M. Raykova, "Secure Single-Server Aggregation with (Poly)Logarithmic Overhead," in *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1253–1269.

[145] W. Yang, Y. Zhang, K. Ye, L. Li, and C.-Z. Xu, "Ffd: A federated learning based method for credit card fraud detection," in *Big Data–BigData 2019: 8th International Congress, Held as Part of the Services Conference Federation, SCF 2019, San Diego, CA, USA, June 25–30, 2019, Proceedings*. Springer, 2019, pp. 18–32.

[146] N. Baracaldo, H. Shaul, N. Drucker, S. Kadhe, and H. Ludwig. (2023) Building privacy-preserving federated learning to help fight financial crime. Accessed: 2024-06-01. [Online]. Available: https://research.ibm.com/blog/privacy-preserving-federated-learning-finance

[147] A. Rahman, M. S. Hossain, G. Muhammad, D. Kundu, T. Debnath, M. Rahman, M. S. I. Khan, P. Tiwari, and S. S. Band, "Federated learning-based ai approaches in smart healthcare: concepts, taxonomies, challenges and open issues," *Cluster computing*, vol. 26, no. 4, pp. 2271–2311, 2023.

[148] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[149] T. Datasets, "Fashion-mnist dataset," 2024, accessed: 2024-08-01. [Online]. Available: https://www.tensorflow.org/datasets/catalog/fashion_mnist

[150] ——, "Cifar-10 dataset," https://www.tensorflow.org/datasets/catalog/cifar10, 2024, accessed: 2024-08-01.

[151] SeeAccessFL, "Access-fl: Github repository," https://github.com/SeeAccessFL/ACCESS-FL.git, 2024, accessed: 2024-09-01.

[152] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010: 19th International Conference on Computational Statistics*. Springer, 2010, pp. 177–186.

[153] L. A. Cox, Jr, "Game theory and risk analysis," *Risk Analysis: An International Journal*, vol. 29, no. 8, pp. 1062–1068, 2009.

[154] S. A. Butler, "Security attribute evaluation method: a cost-benefit approach," in *Proceedings of the 24th international conference on Software engineering*, 2002, pp. 232–240.

[155] U. Fiege, A. Fiat, and A. Shamir, "Zero knowledge proofs of identity," in *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, 1987, pp. 210–217.

[156] P. Martins, L. Sousa, and A. Mariano, "A survey on fully homomorphic encryption: An engineering perspective," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, pp. 1–33, 2017.

[157] C. A. Neff, "A verifiable secret shuffle and its application to e-voting," in *Proceedings of the 8th ACM conference on Computer and Communications Security*, 2001, pp. 116–125.